Protecting Virtual Identities in Mobile IP-based Communication

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Christian Hauser

geb. in Hechingen

Hauptberichter:	Prof. Dr.-Ing. Dr. h. c. mult. Paul J. Kühn
Mitberichter:	Prof. Dr.-Ing. Erwin P. Rathgeb, Universität Duisburg-Essen

Tag der Einreichung:	08. Oktober 2007
Tag der mündlichen Prüfung:	20. Dezember 2007

Institut für Kommunikationsnetze und Rechnersysteme
der Universität Stuttgart
2007

Summary

There are several drivers that will increase the privacy risk, which future IT systems bear. Among the most prominent drivers increasing the value of the sensitive data are the wide spreading of IT in users' lives and the trend towards mobile and context-aware systems. An important driver increasing the probability of misuse is the consolidation of service platforms and networks as well as a presumed change in the network and service provider situation. Because of the growing risk the user's privacy has to be better protected.

In the real world users protect themselves by using different identities. The same concept is translated into the virtual world. For different applications—and thus different providers—users appear differently by using different virtual identities, VIDs. It is obvious that attackers follow the opposite goal. An attacker wants to extend the known VID in order to learn more about the user. This can basically happen in three ways: By observation, by inferring new facts, or by linking fact sets about the same user.

Often, this VID approach is only considered on the application layer, where most of the sensitive data is handled. Nevertheless, the communication system can also violate the VID approach. Thus, the communication system must be designed for supporting VIDs.

The design for a VID supporting communication system implies the evaluation and often the subsequent improvement of several candidate building blocks for their use in the architecture. Those are recurring tasks. Therefore, sound methodologies are needed first of all to evaluate the vulnerabilities and threats of a communication system or one of its building blocks and secondly to subsequently improve the system or the building block.

This thesis develops both methodologies—for evaluation of systems and for their improvement. The methodologies are introduced in a general way and applied to Mobile IPv6 as example in order to achieve a mobility management building block, which supports the VID approach. The methodologies provide for comparable and well-reasoned results and reduce the danger of overlooking design flaws. Moreover, they are strongly formalized in order to provide for a use by non-experts, too.

The application of the evaluation methodology to Mobile IPv6 shows, that all aimed protection goals are broken. The application of the system improvement methodology to Mobile IPv6 results in a new architecture for mobility management.

The new architecture is evaluated by two methodologies. First, the scenario-independent threats and vulnerabilities are evaluated by the introduced methodology. The result is that all protection goals are fulfilled for single attackers as well as for homogeneous attacker groups. Heterogeneous attacker groups may break some of the protection goals in specific scenarios and configurations. The second evaluation examines the remaining threats depending on usage scenarios by event-driven simulation. The main result is that there is a compromise between the values of the metric measuring the VID linking properties and the values of the metric measuring the amount of disclosed information. There are system configurations, in which both goals are met well. The simulation also shows the behavior of the new architecture with respect to changes in the scenario and gives hints for the dimensioning of the system.

Zusammenfassung

Mehrere Faktoren werden die Gefahr für die Privatsphäre von Nutzern zukünftiger IT Systeme vergrößern. Zu den prominentesten Faktoren, welche die Sensitivität der verarbeiteten und kommunizierten Daten vergrößern, gehören die weite Verbreitung von IT Systemen in dem Leben der Benutzer und der Trend hin zu mobilen und kontextbezogenen Systemen. Ein bedeutender Faktor, welcher die Wahrscheinlichkeit eines Missbrauchs dieser Daten erhöht, ist die Konsolidierung der Dienstplattformen und Netze sowie eine zu erwartende Änderung in der Landschaft der Anbieter. Wegen des wachsenden Risikos für die Privatsphäre der Benutzer muss diese besser geschützt werden.

In der realen Welt schützen sich Benutzer dadurch, dass sie mehrere Identitäten verwenden. Dieses Konzept wird in die virtuelle Welt übersetzt. Gegenüber verschiedenen Anwendungen und deren Anbietern erscheint der Benutzer unterschiedlich unter verschiedenen virtuellen Identitäten, VIDs. Es ist offensichtlich, dass Angreifer das gegenteilige Ziel verfolgen. Ein Angreifer will die VID, die er kennt, ausweiten, um mehr über den Benutzer zu erfahren. Eine derartige Ausweitung kann im Wesentlichen auf drei Arten geschehen: Durch Beobachtung, durch Folgerung neuen Wissens aus bekanntem Wissen oder durch die Verkettung von Faktenmengen über einen Benutzer.

Der VID-Ansatz wird häufig nur auf der Anwendungsschicht betrachtet, in welcher die meisten sensitiven Daten verarbeitet werden. Das Kommunikationssystem kann den VID-Ansatz aber auch verletzen. Daher muss das Kommunikationssystem so ausgelegt werden, dass es VIDs unterstützt.

Das Design eines Kommunikationssystems, das VIDs unterstützt, benötigt die Bewertung und häufig auch die Verbesserung von Bausteinen, die zum Einsatz in der Architektur kommen können. Die Bewertung und die Verbesserung sind wiederkehrende Aufgaben. Daher sind fundierte Methoden dafür notwendig.

Diese Arbeit entwickelt Methoden für die Bewertung und für die Verbesserung von Systemen. Die Methoden werden allgemeingültig eingeführt und im Folgenden auf Mobile IPv6 angewendet, um einen Baustein für Mobilitätsmanagement zu erhalten, der VIDs unterstützt. Die Methoden ermöglichen vergleichbare und gut begründete Ergebnisse und redu-

zieren die Gefahr, Design-Fehler zu übersehen. Die Methoden sind streng formalisiert, um auch von Nicht-Experten eingesetzt werden zu können.

Die Anwendung der Bewertungsmethode auf Mobile IPv6 zeigt, dass alle anvisierten Schutzziele verletzt werden. Die Anwendung der Methode zur Systemverbesserung auf Mobile IPv6 erzielt eine neue Architektur für Mobilitätsmanagement.

Diese neue Architektur wird durch zwei Methoden bewertet. Zuerst werden die Szenario-unabhängigen Gefahren und Schwachstellen durch die vorgeschlagene Methode bewertet. Das Ergebnis ist, dass für einzelne Angreifer und für homogene Angreifergruppen alle Schutzziele erfüllt werden. Heterogene Angreifergruppen können in bestimmten Situationen einige Schutzziele brechen. Die zweite Bewertung untersucht Szenario-abhängige Gefahren. Hierzu wird ereignisgesteuerte Simulation eingesetzt. Das Hauptergebnis ist, dass ein Kompromiss notwendig ist, zwischen den Werten der Metrik, welche die Verkettungseigenschaften quantifiziert, und den Werten der Metrik, welche die Menge der aufgedeckten Information quantifiziert. Es gibt hierbei Systemkonfigurationen, für die beide Ziele gut erreicht werden. Die Simulation zeigt auch das Verhalten der neuen Architektur bezüglich Änderungen im Szenario und gibt Dimensionierungshinweise für das System.

Contents

Figures

Tables

Abbreviations

4G	Fourth generation of mobile networks
AAA	Authentication, Authorization, and Accounting
ARP	Address Resolution Protocol, a protocol to resolve layer 2 addresses
CCS	Calculus of Communicating Systems, a process calculus
CN	Correspondent Node, the computer to which the Mobile Node is communicating
CoA	Care-of address, the variable address from the roaming network of a Mobile Node
CSP	Communicating Sequential Processes, a process calculus
DHCP	Dynamic Host Configuration Protocol
DIKW	Data Information Knowledge Wisdom
$\text{Eavesdropper}_{\text{CNHA}}$	An attacker listening on the link between the CN and the HA
$\text{Eavesdropper}_{\text{fMAMN}}$	An attacker listening on the link between the fMA and the MN
$\text{Eavesdropper}_{\text{fMAvMA}}$	An attacker listening on the link between the fMA and the vMA
$\text{Eavesdropper}_{\text{HAMN}}$	An attacker listening on the link between the HA and the MN
$\text{Eavesdropper}_{\text{HAShh}}$	An attacker listening on the link between the HA and the Shh
$\text{Eavesdropper}_{\text{LinkMN}}$	An attacker listening on the access network in which the MN is located
$\text{Eavesdropper}_{\text{MNvMA}}$	An attacker listening on the link between the MN and the vMA
$\text{Eavesdropper}_{\text{ShhMN}}$	An attacker listening on the link between the Shh and the MN
ENISA	European Network and Information Security Agency

EUI	Extended Unique Identifier
fMA	fixed Mobility Agent, one of agents of the new architecture
GSM	Global System for Mobile Communications
ICMP	Internet Control Message Protocol
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IFIP	International Federation for Information Processing
IOI	Item of Interest
IPSec	Internet Protocol Security, an extension of IP for security
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITSec	Information Technology Security Evaluation Criteria, a standard for security evaluations
JAP	Java Anon Proxy, a MIX network written in Java
MD5	Message Digest Algorithm 5, a cryptographic hash algorithm
MN	Mobile Node, the computer of the mobile user
MNCO	Mean Number of Care-of address Observers, a metric for quantifying the implications of the use of multiple servers
MTBF	Mean Time Between Failure, a reliability metric
MTFFL	Mean Time For the First Link, a metric measuring the threat to linking VIDs
nMTFFL	normalized MTFFL, the MTFFL normalized to $T_{A,\,com}$
nTC	normalized TC, the tracelet cardinality normalized to TC_{max}
OSI	Open System Interconnection
PA	Permanent Address, the IP address in the new architecture, by which a given VID will always be reachable
PID	Perceived Identity, the identity as an attacker knows it after interpreting all knowledge about a given user
QoS	Quality of Service
Shh	Shareholder, one of agents of the new architecture
SIP	Session Initiation Protocol

SPI	Security Parameter Index, a parameter of IPSec
SPI_{CNMN}	SPI of the security association between the CN and the MN
SPI_{CNPA}	SPI of the security association between the CN and the PA
TC	Tracelet Cardinality, a metric indicating how many care-of addresses a potential attacker can link to one user
TCmax	The mean value of the maximal length of the TC after which a change of the vMA will happen
TCSec	Trusted Computer System Evaluation Criteria, a security evaluation standard
TH	Temporary Handle, an arbitrary handle by which the agents of the new architecture can indicate a given VID without revealing information about this VID
UMTS	Universal Mobile Telecommunications System
UTRAN	UMTS Terrestrial Radio Access Network
VID	Virtual Identity
VIDindex	An index indicating the VID protection capability of a communication system
vMA	variable Mobility Agent, one of agents of the new architecture

Symbols

A	Offered traffic by all sources
p_{sc}	Probability of changing the vMA on a change of the care-of address
$T_{A, com}$	Interarrival time of communication actions
$T_{A, mov}$	Interarrival time of care-of address changes
$T_{A, sc}$	Interarrival time of changes of the vMA
$T_{H, com}$	Duration of communication actions

Chapter 1

Introduction

The introduction is separated into three sections. Section 1.1 motivates the relevance of privacy protection in future IT systems. Section 1.2 explains the privacy approach of multiple virtual identities. Finally, section 1.3 gives an overview of the thesis.

1.1 Privacy in Future IT Systems

The risk in security and privacy is usually calculated as the product of the value of the sensitive data and the probability of a misuse. There are several drivers that will increase the privacy risk, which future IT systems bring. Among the most prominent drivers increasing the value of the sensitive data are the wide spreading of IT in users' lives and the trend towards mobile and context-aware systems. An important driver increasing the probability of misuse is the consolidation of service platforms and networks as well as a presumed change in the network and service provider situation.

IT specialists are working continuously on new applications which support new aspects of users' lives. This is not restricted to business life, but also extends to private life. New applications often mean processing of additional personal information in IT systems. Every information in the system can potentially be subject to attacks. Thus, more personal information on more aspects of users' lives is going to be threatened.

For managing mobile users, the system has to store and process not only the user's location but also by location changes the user's movement behavior. This often allows for conclusions about other aspects of the user's behavior, e.g., about the current activity.

The goal of context-aware systems is to adapt their behavior according to the user's current situation. While this provides for a better user experience, it implies the need for storing and processing information about the user's situation. There is no clear definition and restriction of such context information to be stored. Basically, every information leading to detailed information on the user and thus to a better adaptation of the functionality of the system is a candidate herefore. Common examples for context information are the location, whether the user is in a business or in a private context, whether the user is alone or in a meeting, and so forth.

The attack probability is increased by a growing consolidation of application support infrastructure. More and more common functionality is grouped in service platforms and communicated over converged networks. Thus, attackers being in one part of the system—especially if it is one of the converged parts—can extend their influence to other parts more easily.

Another driver for an increased probability of data misuse is the growing number of providers—which increases the probability of malicious ones among them. First of all, mobile users will change the service area of their provider, which implies a change of the provider. Secondly, the vast amount of provided applications will imply that there is no possibility for a single provider to provide them all. Thus, users will most likely be in contact with several providers. Those providers can provide different services simultaneously, different services consecutively or one service jointly.

Users will not trust all providers to an equal extent. It can be assumed that many providers will be rather small and specialized. The probability that one of these small providers misuses data can be rated higher than in the case of today's large telecommunication providers, whose misuse would cause a huge scandal and negatively influence their stock price. Moreover, the system, the handled data and the provider structure will become intransparent to users, who do not know any more, which provider can see which personal data, i.e., which provider they have to trust.

Attackers in the sense of this thesis are principals having conflicting interests as compared to the user's interests. Thus, attackers are not necessarily criminals. They can also be legitimate persons or institutions, which follow legal and legitimate interests but those interests are not in line with the user's interests.

Such legitimate interests can be profiling or social sorting, i.e., a differentiated treatment of different users, differentiated conditions, or differentiated services. The probably best known example is personalized advertisement based on information the advertiser has about the user. While this might not be rated very intrusive by some users, there are also more intrusive social sorting possibilities like a rating of the credit worthiness, the restriction to certain payment methods, more expensive service prizes due to a risk penalty, the classification as low priority customer, or the restriction of the service accessibility.

From those different kinds of attackers, different kinds of attack possibilities result. While there can be illegitimate eavesdroppers or criminals undertaking phishing attacker, there can also be legitimate service providers, offering a wide range of services. The information disclosed voluntarily by the user when using those different services can be combined in the background.

Because the risk for abuse of personal data is going to increase, the user's privacy must be better protected. Due to the internationality of network and service infrastructures, it is hard to get a grip on privacy protection by pure legal regulation. Thus, user-controlled and technical privacy protection will be of growing importance. A fundamental approach how users can protect themselves is to use different identities.

1.2 Virtual Identities for Privacy Protection

In the real world, users protect themselves against these threats by using different identities. Users do not tell all individuals the same facts about themselves because the trustworthiness of these individuals varies. They only tell facts about those aspects, which are necessary for the specific interaction with a certain individual. Thus, users leave different impressions at different individuals, i.e., they use different identities. A famous example is that users often appear differently at home than they appear in the business world.

The same concept is translated to the virtual world and is currently gaining a large attention in research, e.g., [122], [201], [180], [77]. For different applications—and thus different providers—users appear differently. For shopping at a book store, they tell their book preferences and a bank account number, for searching help in a medical database, they only tell a pseudonymous name. Many technically interested users already have several different eMail addresses in the virtual world for those different transactions. With them, they actually appear as different virtual identities to different communication partners.

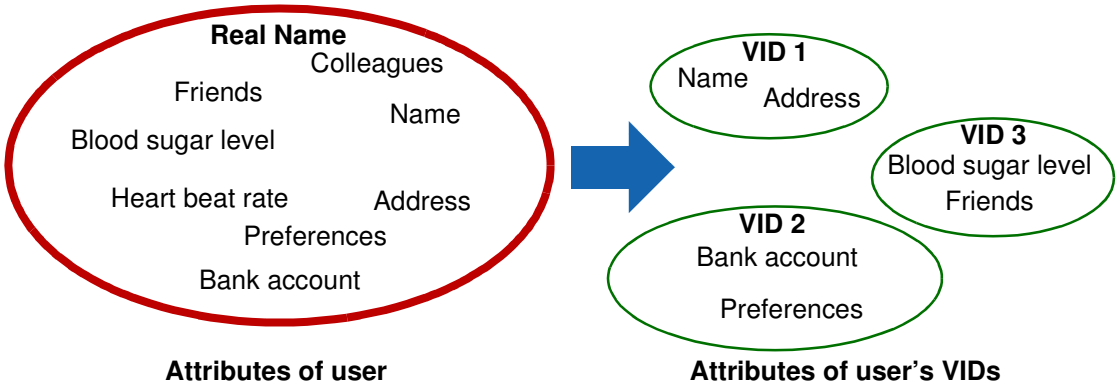


Figure 1.1: Principle of Virtual Identities

Figure 1.1 shows schematically the principle of *Virtual Identities, VIDs*. A virtual identity is a set of attributes of a user represented in a technical system. It contains an identifier, e.g., an eMail address for naming the virtual user. The contained attributes are configured by the user.

In the real world, a user has a real name and an countless number of attributes like a postal address or a set of friends' names. Because there is no need to tell everybody all the attributes, the user uses different VIDs towards different communication partners. Such a VID contains a VID-identifier and a subset of the attributes. Ideally, the subset is minimal in a way that the communication partner only gets the information necessary to fulfill the task.

In the example, the user uses VID 1 containing the name and address for receiving information material for a vacation trip. VID 2 is used towards a book store and VID 3 for a diabetes surveillance service, which alerts the user's friends in an emergency case.

It is obvious, that attackers follow the opposite goal. An attacker wants to extend the known VID in order to learn more about the user. This can basically happen in three ways:

1. By observation: The attacker can observe more facts about the user, e.g., the attacker can try to convince the user to reveal more facts or do more transactions with the user in order to get more facts.
2. By inference: The attacker can infer new facts from the facts, which the user revealed, e.g., infer the user's wealth from the postal address.
3. By linking: The attacker can link several VIDs of the same user in order to merge the known facts.

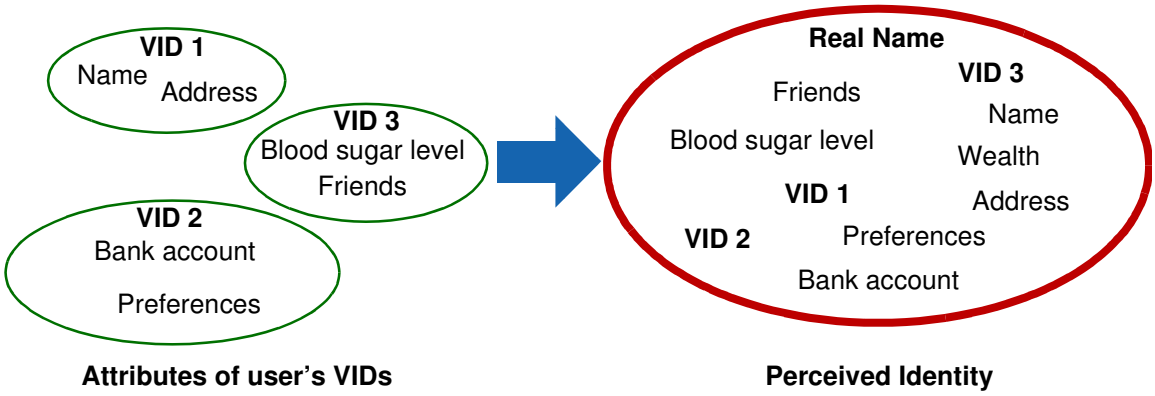


Figure 1.2: Attack on VIDs

It is possible that an attacker can see different VIDs of a user, e.g., if the attacker collaborates with another one or if the user uses different VIDs with the same attacker. Figure 1.2 shows the result if an attacker can link all three VIDs. Moreover, the user's wealth is inferred from the postal address. This results in a rich set of user attributes.

The resulting view the attacker has gained on the user is called *Perceived Identity, PID*, because it is how the attacker in fact perceives the user. This PID can consist of several VIDs and thus contain several VID-identifiers. The VID is the identity under which the user wants to be perceived by the respective communication partner. So, the goal of VID protection is to keep the PID as similar as possible like the intended VID.

Most personal data will be handled on the application layer. This is also the data, which usually is intuitive to the user. Thus, the VID approach is traditionally an application layer approach and the VIDs are configured regarding the contained application data. Nevertheless, the communication system reveals data about the users, too. If the user is using the same IP address for all VIDs, it will be a simple task for an attacker to see that the VIDs belong to the same user and thus to link all known VIDs. Moreover, the IP address allows for inference of the geographic location, which is a sensitive attribute of the user.

Thus, the communication system can directly undermine the VID approach and has to be specially designed for not doing so. This leads to the goal of the thesis, which is to evaluate the threats of the communication system to the VID approach and to improve the VID protection properties of the communication system.

1.3 Overview of the Thesis

Communication engineers have a complex task, when designing a new communication system. For fulfilling the given requirements, there are usually several proposed candidate architectures to choose from. Each communication system architecture consists of several building blocks, e.g., a QoS subsystem, a mobility management subsystem, an AAA subsystem, for which often again different realizations and even different implementations exist. Moreover, the configuration can influence the capabilities of a building block widely. The engineer has to evaluate, which architecture with which building blocks in which realization, implementation, and configuration protects VIDs best. Moreover, after introducing an extension, e.g., a functional extension or a protection mechanism, new problems could be introduced and the result has to be evaluated regarding the VID protection properties again.

Thus, evaluation of VID protection properties is a recurring task. Because it has to be applied to all systems, every communication engineer should be able to do the evaluation and subsequent improvement of the system. For this, a methodology is needed, which a VID expert can invent once, and which can be applied by communication engineers with only rough VID knowledge afterwards.

Another challenge is that the evaluation results of different candidate architectures must be comparable. Today, this is difficult to achieve due to a lack of metrics for VID protection and due to the lack of a standardized evaluation methodology. Thus, different designers of alternative solutions are evaluating their systems in different ways—if at all—making different statements about the systems.

Thus, this thesis has basically a threefold goal. The first goal is to evaluate the threats to the VID approach by Mobile IPv6 and to subsequently diminish them. For this, a methodology for evaluating communication systems will be developed. Then, a methodology for improving communication systems will be developed. Finally, the improved architecture yielded by the methodologies will be evaluated.

Several demands are made on the methodologies. The evaluation methodology has to answer the questions, which personal information attackers can observe, which additional personal information they can infer and which fact sets about one user they can link. The improvement methodology must allow to diminish the identified threats. Both methodologies have to reduce the danger of overlooking design flaws, have to provide a reasoning of the results, and have to provide for comparability and documentation of the results as well as of the assumptions. Finally, the methodologies must be semi-formal in order to well specify the necessary steps.

The structure of the thesis is as follows. Chapter 2 starts by explaining the fundamentals for understanding the thesis. Thereafter, chapter 3 explains the evaluation methodology and applies it to Mobile IPv6. Chapter 4 then develops the improvement methodology and improves Mobile IPv6. Chapters 5 and 6 give an evaluation of the new architecture. Thereby, chapter 5 applies the developed evaluation methodology in order to yield scenario-independent results, whereas chapter 6 evaluates the architecture by event-driven simulation and examines the scenario-dependent behavior. Finally, chapter 7 concludes and gives an outlook onto possible future work.

Chapter 2

Fundamentals

This chapter gives the fundamental knowledge necessary to understand the thesis. Therefore, section 2.1 starts by introducing security and privacy with a focus on the relevant terms. Because privacy is much about information and knowledge, section 2.2 gives the necessary fundamentals in knowledge engineering. Section 2.3 introduces methodological fundamentals from mathematics and database design, which are necessary to evaluate the protection of systems regarding the VID approach.

This thesis examines the protection of VIDs in mobile, IP-based communication. If not noted differently, IP means IPv6 [51] in this thesis. Therefore, section 2.4 gives the technological base with an introduction of mobility management in the Internet. This section contains an introduction of Mobile IPv6, which is the starting point of the thesis. Section 2.5 finally, structures the problem underlying this thesis and defines the focus of the thesis in this structure.

2.1 Security and Privacy

IT security is a rather new science. For basic mechanisms and principle there are a lot of textbooks, e.g., [174], [192], [208], [209]. While everybody has a notion of what security means in general, it still lacks a consolidated terminological base. Therefore, this section will set the terminological scene for the thesis. Here, the base from [78], [79] for overall IT security and [175] for relevant aspects of privacy is used. They are explained in section 2.1.1 and section 2.1.2 respectively.

2.1.1 Security

According to [78], security and safety are strongly related. Both prevent harm from the system. Safety thereby aims at preventing accidental harm, whereas security aims at preventing intentional, malicious harm.

[79] defines security as one quality factor of an IT system. This quality factor can be subdivided in a hierarchy of subfactors according to Figure 2.1, which uses UML notation. Integ-

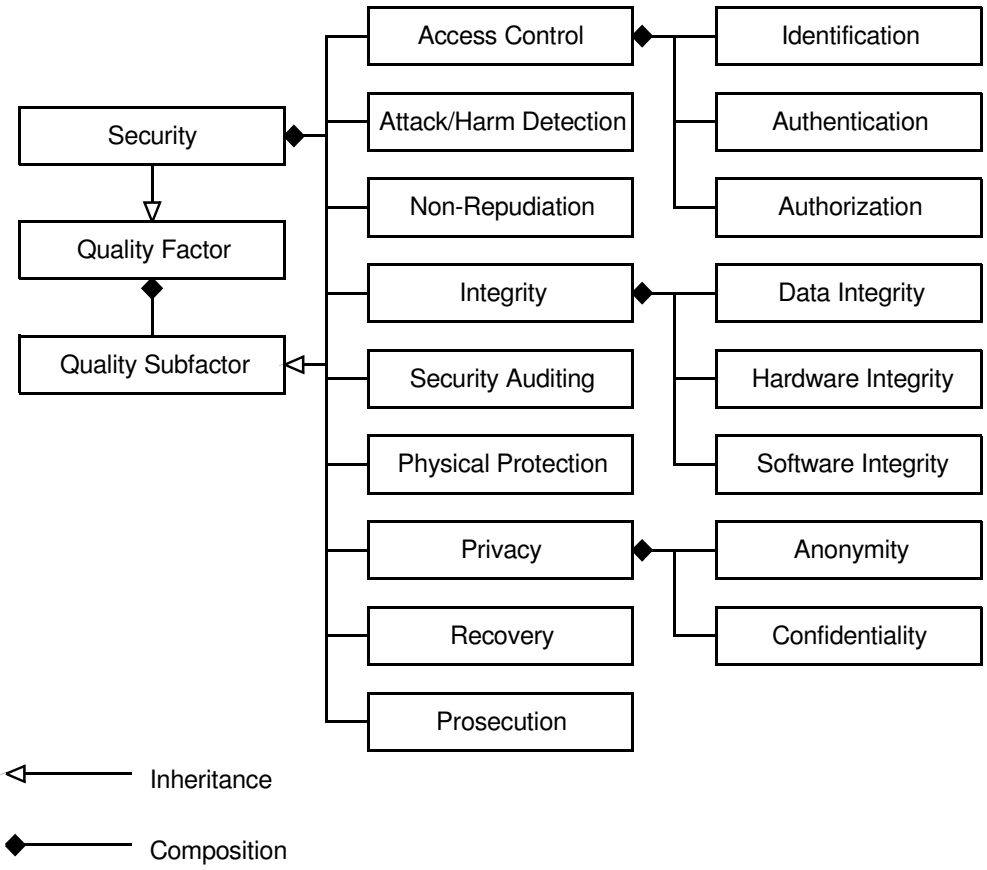


Figure 2.1: Quality subfactors of security, modified from [79]

rity protection subdivides into integrity protection of data, hardware, and software. Access control requires identification to identify the requestor's identity, authentication to verify the requestors identity, and authorization to check the requestor's access rights.

This view comes from the operation of an IT system and also contains procedures like detecting intrusions, prosecution of malicious actions, or physical protection. For this thesis, only the privacy subfactor is relevant, which is one part of security in this structure. Privacy consists of anonymity and confidentiality according to [79]. For this thesis, privacy is extended with additional subfactors concerning VID protection like shown in Figure 2.2.

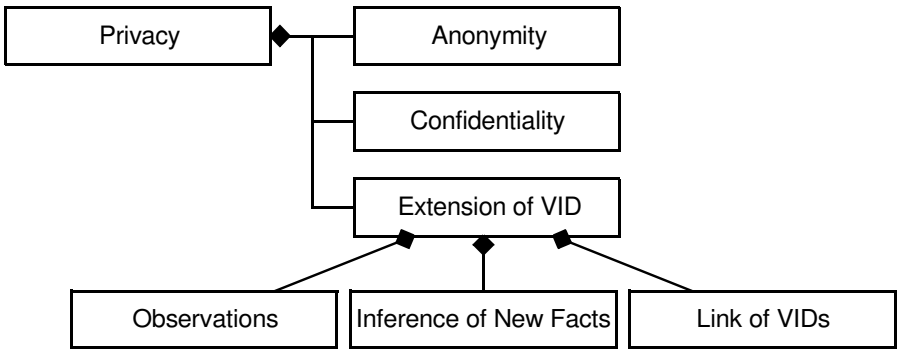


Figure 2.2: Extension of privacy subfactor

For system design and evaluation often only three protection classes are defined, e.g., in [174]. The three classes are confidentiality, integrity, and availability. The three classes allow for expression of all protection goals. A protection goal defines the protection class together with the asset, to which the protection class is to be applied. Anonymity, for instance, is confidentiality of the identity in this structure. The protection goal oriented structure is different to the structure of the quality factors in not emphasizing protection activities how to achieve the goals.

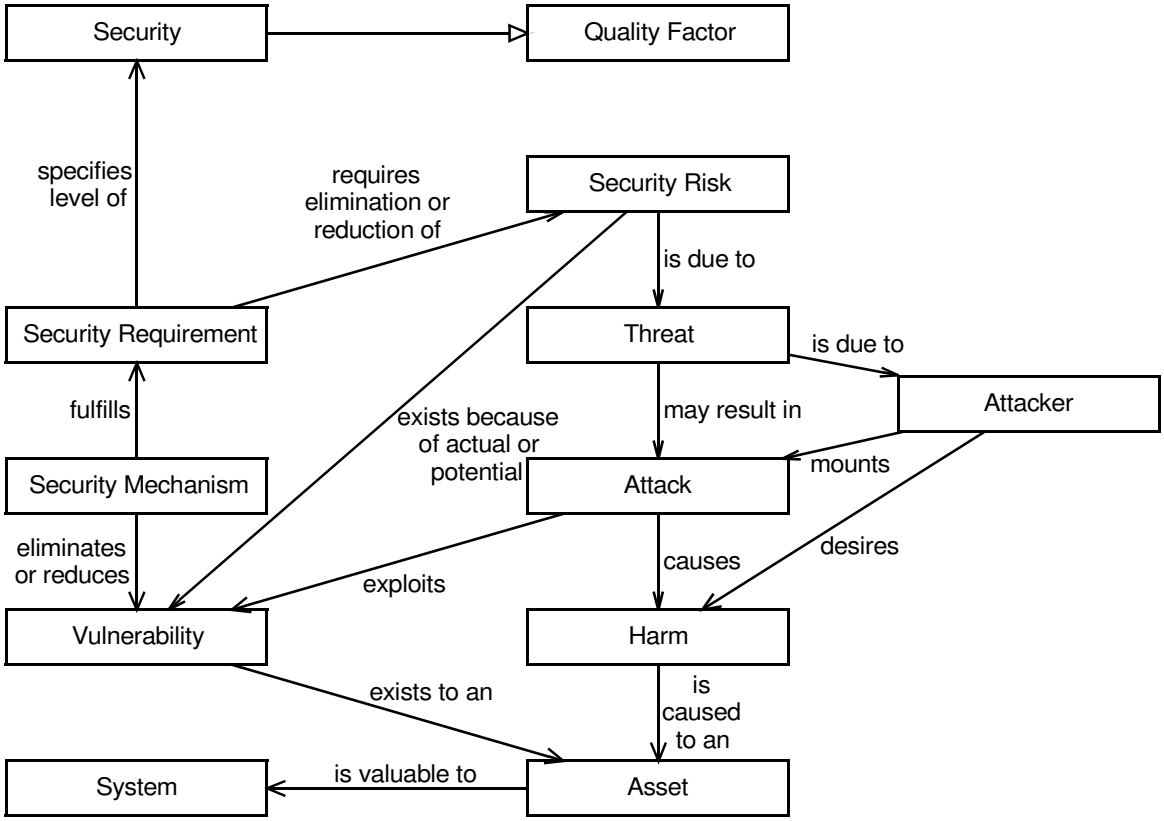


Figure 2.3: Concepts of security and their relations, modified from [79]

[79] additionally gives a comprehensive overview of the concepts of security and their relations. This is depicted in Figure 2.3 in a simplified version. For this thesis the lower part is the most important one. It starts by saying that assets are valuable to the system. Vulnerabilities may exist to those assets and harm can be caused to them by an attack, which exploits a vulnerability. A threat may result in an attack mounted by an attacker, who desires harm to an asset, and causes a security risk. Security requirements require elimination or reduction of security risks and specify the level of security. Security risks exist because of vulnerabilities. Security mechanisms—also called countermeasures—eliminate or reduce those vulnerabilities and fulfill security requirements.

The most relevant distinction here is that a vulnerability is not the same as a threat. A threat needs for existence an attacker mounting an attack to exploit an existing vulnerability. This is in line with [165] saying that a threat needs an actor to exploit a vulnerability.

Prior to analyzing or improving the security of a system, the attacker's power against which the protection must hold, needs to be defined. This results in an attacker model. The attacker model comprises aspects like the attacker's motivation and thus, the effort, which is likely

to be spent for an attack, or the information which an attacker has access to. Closely related is the question, whether the considered attacker is an insider or an outsider to the system. An insider is an entity being involved in the operation of the system. An outsider is not directly involved in the operation of the system. An example for an outsider is an attacker eavesdropping a system.

Attackers can also be classified according to the attacks they can launch. Attacks can be passive or active. In passive attacks, which are of relevance in this thesis, an attacker tries to gain information by eavesdropping the system. It is virtually impossible to detect eavesdropping by today's technologies. In contrast, an active attack alters the information in the system, e.g., by modifying message content. Because active attacks change the information, they can principally be detected, e.g., by evaluating a digital signature being applied over the sensitive information.

There is a multitude of other aspects to distinguish attackers, e.g., according to the lifecycle of a system—operation vs. design stage vs. pre-design stage—according to the role—user vs. network provider vs. maintenance staff.

For most applications, it can be assumed that an attacker does not forget anything. Thus, the attacker's knowledge only increases. Yet, in some cases it might be that the value of the knowledge decreases, because it dates out. Therefore, it is usually necessary to record not only the observed information, but also the time of the observation [175].

The focus of this thesis is in identification and extinction of vulnerabilities regarding the VID approach. To this aim, the potential attackers are considered and the threats resulting from the vulnerabilities are evaluated. Thereby, insiders as well as outsiders during operation of the system are considered, which are undertaking passive attacks, and which are users—or application service providers, which is equivalent here—or network providers. Table 2.1 illustrates the considered attackers.

Protection of the VID approach is a subfactor of privacy. The relevant fundamentals in the privacy area are described in the next section.

2.1.2 Privacy

Privacy is an important aspect in real life and therefore also in digital life. Its meaning is very broad and comprises many aspects, even when only focused on the IT world. Basically, it means protection of access to an individual's personal information by other individuals. Here, only the aspects around protection of a user's identity are important. They are described in the following based on [175], which gives the probably soundest definitions in this area.

The description is based on the setting of a communication network with senders and recipients exchanging messages. Generally, an Item of Interest, IOI denotes what the attacker wants to know, e.g., who is the sender of a message. While the definitions in the area of anonymity, unobservability etc. are claimed to be rather stable, the area of identity management is still rapidly evolving and thus, the definitions do not have the same maturity. While section 2.1.2.1 describes the aspects around anonymity, section 2.1.2.2 describes terms in the area of pseudonymity and identity.

Aspect	Possibilities
Location	Insider
	Outsider
Attacks	Passive
	Active
Role	User, application service provider
	Network service provider
	Maintenance staff
	...
Lifecycle	Pre-design time
	Design time
	Operation time
	...
...	

Table 2.1: Selection of aspects of an attacker model

2.1.2.1 Anonymity, Unlinkability, Undetectability, and Unobservability

It is first necessary to define the term of a subject. A subject is a possibly acting entity. Usually, this is a private individual or a computer acting on behalf of that individual. In these definitions, a subject is a single entity and there are no groups considered to be a subject.

Anonymity is the state of being not identifiable within a set of subjects. This set of subjects is called *anonymity set*. In case of acting entities, the anonymity set is the set of subjects who could have caused an action. In case of addresses, the anonymity set is the set of subjects, who might be addressed.

If all other circumstances are kept equal, there are two drivers increasing the strength of anonymity. The first driver is a growing anonymity set. The second driver is a more uniform distribution of the probability, that a certain contained subject is the real subject. There are approaches, which are a more exact metric of the strength of anonymity by considering the probability distribution. They result in a metric considering the entropy of the distribution [58], [200].

Besides this quantification of anonymity, there is also *robustness of anonymity* describing how stable the anonymity is regarding changes in the setting. *Anonymity quality* comprises both, the quantity—also called strength—and the robustness.

Regarding a communication network, anonymity can be subdivided into *sender anonymity*, *recipient anonymity* or *relationship anonymity*. The latter means that it is untraceable who communicates with whom.

Unlinkability is defined relating two or more items of interests, e.g., subjects or messages. It is defined that from an attacker's perspective, in the system the IOIs are no more and no less

related after an observation than they have been before. With the unlinkability, anonymity can be further explained. Anonymity is unlinkability of an IOI and the related subject, i.e., the owner or holder of the IOI is unknown.

Undetectability means the state that it is indistinguishable, whether an IOI exists or not. For instance, messages that are indistinguishable from noise are undetectable.

Slightly different is the term of *unobservability.* This is the state that a) whether the IOIs exist or not is indistinguishable by all subjects unrelated to the IOI and b) the subjects related to the IOI are anonymous even against each other. The following example of sending a message as IOI illustrates this definition. Condition a) means that nobody but the sender and the recipient can detect the sent message. Condition b) means that even the sender is anonymous against the recipient and vice versa.

The next section explains the terms around pseudonyms and identities.

2.1.2.2 Pseudonymity and Identity

A *pseudonym* is an identifier of a subject, e.g., a sender of a message, other than the subject's real name. Consequently, if a subject uses a different identifier than the real name, the subject is *pseudonymous*. The use of pseudonyms as identifiers is called *pseudonymity.* The term pseudonym comes from Greek "pseudonumon", which means "falsely named". It is concatenated from "pseudo" meaning "false" and "onuma" meaning "name".

Often, pseudonyms are used when the subjects want to conceal their real names, but still need to link their actions. This can be necessary for building up a reputation or for accountability. It is possible to transfer credentials from one pseudonym to another by mechanisms described in, e.g., [31], [33]. This might be used for access control or payment, for instance. There may exist *group pseudonyms* for a set of subjects and *transferable pseudonyms*, which can change the subject.

Identity is an identifier with a set of attributes of an individual. It is defined in [175] in a way that the set of attributes must make the individual identifiable in a certain set of subjects, an *identifiability set*. Consequently, there is the term *identifiability* defined as the state of being identifiable in the identifiability set. Identifiability is the better the larger the identifiability set is, i.e., the more exactly the individual can be identified. A *complete identity* contains the complete set of attributes of an individual. In this thesis, the identity is always meant to denote a complete identity as defined in [175].

A *partial identity* represents an identity in a certain context and is a subset of the complete identity's attributes. A *digital identity* denotes a representation of a partial identity in a technical system. It only considers attributes, which are accessible by this system. A *partial digital identity* or *virtual identity* is an identifier with a subset of the technically accessible attributes.

This thesis evaluates unlinkability with respect to protecting virtual identities from being linked to other virtual identities of the same user. While the definition of unlinkability in 2.1.2.1 makes well sense in a scenario with uncertain information, here a different definition is used. As it will be pointed out later in 2.5.2, only certain information is of relevance in this thesis. Regarding unlinkability this means, that an attacker only assumes a link between two VIDs if being absolutely sure. Therefore, unlinkability of VIDs here is defined as uncertainty about the link from the attacker's perspective.

When evaluating a system regarding its VID protection properties, it is necessary to represent the flowing information and the attacker's knowledge. Subsequently, the knowledge, which an attacker can gain from the system should be reduced. This requires a basis from knowledge engineering, which will be laid in the next section.

2.2 Knowledge Engineering

This section is structured as follows. At first terms about information and knowledge are described in 2.2.1, followed by explanations of facts and rules in 2.2.2. Finally, 2.2.3 describes the terms of the inference area, which is a research area in database technology. Finally, section 2.2.4 gives the necessary background in knowledge representation techniques.

2.2.1 Data, Information, Knowledge, Wisdom

Knowledge engineering and related areas like artificial intelligence are still an emerging area and differ in approach depending on the specific goal. Therefore, there is not any agreed terminology, which suits all purposes. This thesis sticks with the definitions of data, information, and knowledge in, e.g., [18], [107], [202]. For completeness, also the terms of understanding and wisdom are explained. [202] provides an essay about the history of this hierarchy, which goes back to the poet T. S. Eliot [65].

- *Data:* Data are raw symbols with only a syntax. The symbols do not contain any information per se besides their existence.
- *Information:* Information is based on syntax and on semantics. It gives a meaning to data by putting it into relation to other information. An example is to define an integer value as age of a user. This implies meanings such as the fact that an age older than 150 is not possible for a human being of the year 2007 or that an age younger than 18 implies that the individual does not have certain rights by law in Germany. For all those statements, the age is put into relation to other information like the mortality age or the legal grown-up age. According to [18], information provides answers to the questions of "who", "when", "what", and "where".
- *Knowledge:* Knowledge is the collection of information. This information has a useful meaning to its owner. Knowledge alone does not provide for the possibility of inferring new knowledge. It provides answers to "how" questions. An example of knowledge is the memorized "times table", which is learned in elementary school. The pupils are memorizing the results of "1x6", "2x6", "3x6" etc. But the coherences behind the table are not known and thus, the pupils cannot answer the result of "124x376", for instance. There are more detailed views on different kinds of knowledge like tacit and explicit or declarative and procedural knowledge, e.g., in [168].
- *Understanding:* Understanding is cognitive and analytical. It provides for the ability to infer new knowledge from existing knowledge. It provides the answer to "why" questions. The difference between understanding and knowledge is like the difference between learned and memorized facts.
- *Wisdom:* Wisdom is the highest level. It provides for understanding where there has not been any understanding, so far.

Often, understanding is not seen as a level on its own, but rather as the transition between the other levels. Data, information, knowledge, and wisdom, DIKW are explained in [41] by Figure 2.4. The transition from one level to the other happens by a process of deeper understanding. With rising levels, data, information, and knowledge are set into a larger context.

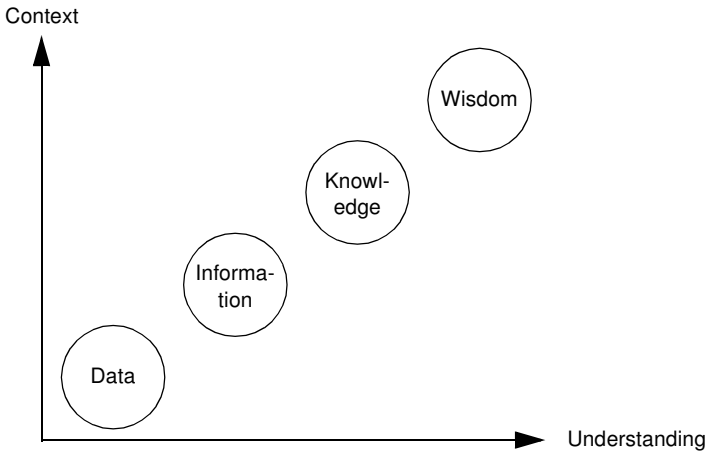


Figure 2.4: The DIKW hierarchy, modified from [41]

Only in simple examples, privacy engineering can be based on the pure data level. This is the case, if the sensitivity of the revealed information directly corresponds to revealed data. Then, it is not necessary to evaluate it further, e.g., to put it in relation to other revealed data in order to rate the sensitivity of the aggregate. In this case, there is no reasoning on the information contained in the disclosed data necessary. When analyzing threats to VIDs, the data level is not sufficient but a consideration on information level with knowledge aspects is necessary. For this, the terms of facts and rules are relevant. The next section explains them.

2.2.2 Facts and Rules

Basically, facts contain basic pieces of knowledge. Rules are used to increase knowledge by concluding new facts, i.e., by adding new information to the knowledge. The exact explanation used in this thesis is taken from the area of deductive databases [131], [67], [183], [141]. Deductive databases have their origin in the problem of interpretation and understanding of large amounts of data being stored in databases. Handling the large amount of data is supported by deductive databases, which allow for a certain degree of automatic reasoning about the stored data.

The framework of deductive databases introduces an extensional database and an intensional database. In the extensional database, the explicit *facts* are stored. This resembles a common relational database. The intensional database contains inference *rules*, which can be considered as virtual relations in the sense of a relational database. By those rules, an inference component can automatically conclude new facts from the facts being stored in the extensional database. Thus, the new facts are virtually stored in the database and are implicitly contained in the database. A more detailed discussion of the inference process itself is contained in 2.2.3.

The deductive database framework is beneficial for large amounts of data, because it is sufficient to store a basic set of facts and the rules in secondary storage. The rest of the knowl-

edge can be computed from those two bases. Thus, the rules are containing the knowledge in a condensed form of higher abstraction. In this sense, a fact is a concrete piece of information and a rule is a prescription how to interpret facts in order to conclude new facts. Both, facts and rules are forms of knowledge and contain information.

The attacks on the VID approach introduced in chapter 1 can be considered as a special application of a deductive database. The observations of an attacker build the extensional database, i.e., the basic set of known facts. The attacker can compute new facts by applying rules on the known facts and thus, extending his knowledge about a user's VID.

For the proposed methodology, some own definitions are necessary in addition:

- *Fact type:* A fact type is a structure consisting of a content field and a timestamp field. The content field has a certain application specific type, i.e., it may be for instance an IP address or a MAC address.
- *Fact*: A fact is an instantiation of a fact type, with the content field containing a value of the content field's type and the timestamp field containing the time when this value was valid. The value of the content field and of the timestamp field never change, once a fact is instantiated.
  There are so-called *constant* fact types, where only one instance, i.e., exactly one fact of the given type, may exist and *variable* fact types, which may have an arbitrary number of instantiations. For constant fact types, the timestamp is not important, because the value is always valid. Two facts of a constant fact type are identical, if the value of the content field is identical. Two facts of a variable fact type are only identical, if the value of the content field and the timestamp are identical. Then, the two facts are indeed the same fact.
  The value of the content field contains a piece of information about a state in the system or about the transition of a state.
- *Fact set:* A fact set is a set of facts about the same user. Note, that this can comprise facts about several VIDs, if these VIDs belong to the same user. Types can be specified for fact sets equivalently to elementary fact types.
- *Trace and Tracelet:* Traces and tracelets are fact sets containing facts of the same fact type with different timestamps. This fact set is called a trace, if the contained facts are more sensitive than a single fact of the given fact type. This is aligned with the definition of aggregation problems in databases [54], [144], [109]. The sensitivity here is meant regarding vulnerabilities of the VID approach and is further explained in 3.2.2. The fact set is called a tracelet, if the contained facts are not more sensitive regarding VIDs than a single fact of the given type.
- *Observation:* An observation is a special fact or fact set. It is special in the sense that it is directly observed by inspecting the real world instead of being inferred from other observations by application of rules.

After having laid the foundation of knowledge engineering and the basic terms on which the proposed methodology is built, it is possible to address knowledge interpretation. The next section introduces the terms around the inference process.

2.2.3 Inference

The term *inference* is common in logical reasoning as well as in philosophy. It means the process of gaining new facts by cognitive reasoning about already known facts. There are different ways of inferring new facts. The best known ones are deduction, induction, and abduction. The foundations of inference go back up to early philosophers like Aristotle or Plato. A well understandable summary including a discussion is given in [106].

In the following, the three major ways of inference are shortly introduced:

- *Deduction:* Deduction is the inference from generality to the single specific fact. The general rule is known and by application to a specific fact, a new fact is concluded. An example from [106] for a general rule is that all bananas are yellow. The known fact is that a given thing is a banana. Therefore, the new fact can be deduced that the thing under consideration is yellow. In [233], it is claimed that it is not possible to gain new knowledge by deduction as all the underlying knowledge is already available. The new facts are just made explicit, but implicitly, they have already been known. This is in line with the view of extensional and intensional databases of a deductive database.
- *Induction:* Induction follows the direction opposite to deduction. It observes single but common pieces of facts and concludes a general rule. In the example, it is observed that from six given yellow fruits, five ones are a banana. By induction, the general rule that bananas are yellow can be inferred. This rule is true by a high probability as several facts satisfying the rule are observed. Induction indeed increases knowledge as the inferred rule covers more facts than the sources of the induction.
- *Abduction:* Abduction resembles induction, but is only concluded from a single observed fact. Therefore, the inferred rule is less probably true than a rule being inferred by induction. In the example, one yellow fruit is observed. It is known that bananas are yellow, so the hypothesis is abduced that the given fruit may also be a banana. Abduction also increases knowledge.

From the explained inference possibilities, only the first one is of relevance to this thesis. In the remainder, the umbrella term inference will be used, because it is commonly used in related privacy work in the database sector.

The next section gives the background in knowledge representation. This is usually a prerequisite to work with knowledge.

2.2.4 Knowledge Representation

Databases are representing a certain part of the real world in an IT system by storing the relevant data to describe it. The database contains this respective part of the world completely in the sense that it comprises all relevant aspects to describe the status and if necessary the behavior. A database model determines the necessary data items and their organization, i.e., data types and relations between them. The data description must be complete enough for the application on top of the database to yield correct results, which can be used in the real world again. Thus, the database model is often described as *miniworld* or *universe of discourse*, because it is the part of the world or of the universe in which the IT system is acting. It abstracts the real world.

Usually, database design follows four steps [67], [131]. The first one is a requirements analysis, which is followed by a conceptual design. Only when the conceptual design is stable, it is translated to a logical design on implementation level, which is finally fine tuned on physical level. The following list describes each step.

- *Requirements analysis:* The requirements analysis captures all relevant requirements. Thereby, not only functional requirements about the part of the real world to be represented are important. The analysis also comprises nonfunctional parameters like necessary response times for a query, scalability requirements, or security constraints.
- *Conceptual design:* The conceptual design aims at structuring the miniworld. One important goal of this step is to identify conflicting requirements already in an early stage of development. The database experts first of all classify the data in types and relationships between those data types. Then, they identify constraints, which define legal instantiations of the data types and relationships. The conceptual design serves well for discussing the results for verification with both, database experts realizing the database as well as non-experts later using the database, because it is rather abstract and usually defined in an expressive representation.
- *Logical design:* The logical design translates the conceptual model to the actual implementation level. For this, the experts model the database by means of the realization technology. In terms of a relational database, this is an expression in relations, attributes, and tuples.
- *Physical design:* The physical design has the primary goal of increasing the database performance. It tunes physical concepts like data blocks, intelligent pointers or index structures. The experts must have a good know-how of the underlying operating system or even of the hardware.

The questions of this thesis are a typical application at the conceptual level. The goal is not to build a real database but to evaluate sensitivity of information. The answers are independent of the realization how the attacker would actually store the data, i.e, of the realization technologies.

A wide spread modelling technique on the conceptual design level are entity relationship diagrams. The following paragraphs summarize the most important aspects for this thesis from [67], [131].

According to their name, entities and relationships are the building blocks, which can both occur in the form of types as well as of instantiations. Thereby, similar entities and similar relationships are abstracted to types like it is done with the facts in this thesis or with variables in programming languages.

An entity usually is a thing in the real world with an independent existence. The entity can be physical like an individual or conceptual like a job. It is further described by attributes. It is possible to distinguish actually stored attributes, e.g., a birth date, and derived attributes, which are not stored in the database but which can be computed from stored data, e.g., the age.

A relationship describes the association between entities. It can be binary as well as n-ary. Attributes can further describe a relationship. A relationship type can be considered as a mathematical relation between the participating entity types, i.e., a subset of the cartesian product of the participating entity types [67]. A relationship instantiation is an association

with participating entities from each entity type that participates in the relationship type. Relationship types can be subject to constraints, which model certain conditions of the miniworld. Such a constraint can be a cardinality ratio defining the number of entities of each type, which have to participate in a relationship instantiation.

Entity relationship diagrams usually operate on entity and relationship types instead of individual instances. This is sensible, because a database schema changes rarely, whereas its extension—the instances—changes rather frequently. Moreover, the type-level can fulfill the tasks of a conceptual representation. The instance-level is not needed for this thesis, focusing on type-level interpretations as will be shown in 2.5.2.

In this thesis, entities are fact types and the mathematical relations between the fact types are represented by relationships between the entities. The entity relationship diagram can also describe the properties of the relations between fact types by cardinality ratios of the relationships.

In the literature, there are different notations of an entity relationship diagram. Most commonly, a form is used, where entities are represented as boxes and relationships as diamonds. In Figure 2.5, entities of type employee work for an entity of type company. Employees as well as companies have a name and an address as attributes. Employees additionally have a salary. The WORKS_FOR relationship has a start date of the employment as attribute.

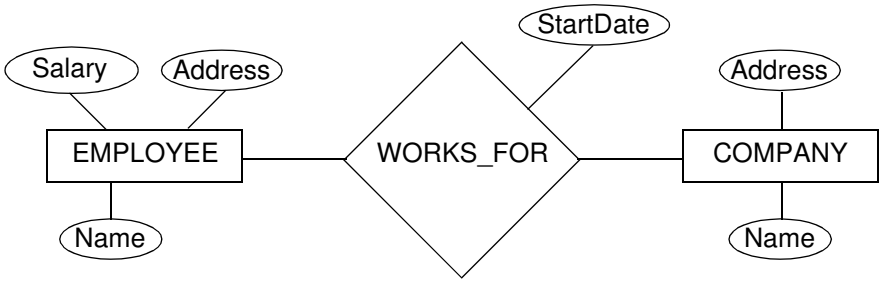


Figure 2.5: Simple entity relationship diagram

For this thesis, this notation is too expressive and the conciseness would suffer unnecessarily. First of all, only binary relationships are used in order to achieve a simpler modelling methodology. Secondly, attributes of entities or relationships are not necessary. The only important property of relationships is their cardinality as will be motivated in section 2.3. The relationships can be considered as being all of the same type "is-related-to".

Entity relationship diagrams can be graphically represented in different ways. This thesis uses a mixture of the popular *chicken-feet notation* and the notation used in *Bachman Diagrams* [67].

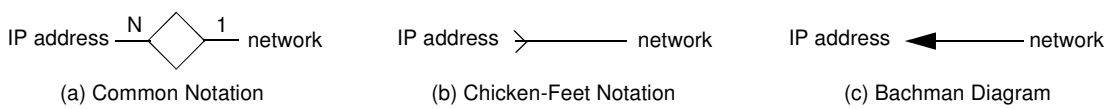


Figure 2.6: Different notations of entity relationship diagrams

Figure 2.6 shows an example of a network consisting of several IP addresses. Three graphical representations are shown—(a) the most popular one, (b) the chicken-feet notation, and

(c) the notation used in Bachman Diagrams. (a) is used for relationships with more than two participants. This is not necessary here. The form of arrows used in (c) is more intuitive than the chicken-feet notation in (b), because functional dependencies, which correspond to the relationships here, are represented by arrows, cf. section 2.3.2. (c) is misleading for our work, because it shows the reverse direction of the arrow like in the representation of functional dependencies, cf. 2.3.2.

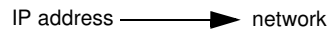


Figure 2.7: Representation used in this thesis

Figure 2.7 shows the example in the chosen representation. It means that a network has several IP addresses, but each IP address belongs to exactly one network. This notation resembles the ones used for evaluating database inference problems in, e.g., [109] and [159], which improves coherence to related work and helps understanding. Often, the names of types in an entity relationship diagram start with a capital letter or even consist only of capital letters, whereas instantiations are depicted with small letters. For readability reasons, the representation here uses only small letters except for names like IP, MAC, or VID, although types are modelled. Because instantiations are not modelled at all, there is no danger for misunderstandings.

A relation with an arrowhead on at least one side reflects a function, cf. 2.3.1. The representation must be extended in order to reflect whether the function can indeed be computed or whether only its existence is known. This is done by using a filled arrowhead in case the function can be computed, i.e., if the considered attacker is assumed to know or to be able to easily determine the mapping of the function. As section 2.3 will show, the attacker can compute the fact type at the end of the arrow from a fact type at the beginning of the arrow. An empty arrowhead instead, means that the mapping is not known but only the cardinality of the relationship is known, i.e., whether both participating fact types are related by a function or by a different relation.

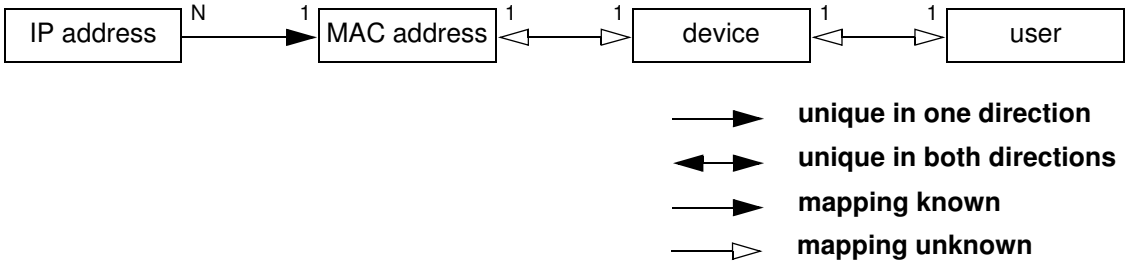


Figure 2.8: Simple knowledge model

Figure 2.8 shows a simple example using all elements of the representation. It shows four fact types: IP(v6) addresses, MAC addresses, devices and users. IP addresses are assumed to be built using stateless autoconfiguration by including the MAC address [218]. A device is assumed to have only one MAC address and to belong to one user. A user only has one device.

The relation between the IP address fact type and the MAC address fact type is of cardinality N:1. To each IP address, there is one corresponding MAC address, from which it was built. Each MAC address can be built into many IP addresses, i.e., refers to several IP

addresses. Thus, IP addresses functionally determine the corresponding MAC addresses. The mapping of this function, i.e., how to compute the MAC address from a given IP address, is known by the attacker. Therefore the arrowhead is solid.

The relation between the MAC address type and the device type is a 1:1 relation, because each device is assumed to have one MAC address. The same holds true for the relation between the device type and the user type. From all those functions, attackers do not know the mapping, i.e., no attacker can compute the device from a given MAC address or vice versa. Neither can an attacker compute the user from the device or vice versa. For explanatory reasons, the cardinality of the relations is included in the example. A "1" corresponds to an arrowhead and an "N" corresponds to no arrowhead.

Protection of VIDs is based on evaluating, when an attacker can link two VIDs, i.e., know that both VIDs uniquely relate to the same user and on evaluating, which additional facts are determined by the facts known to an attacker. For understanding those issues, the next section defines the necessary base.

2.3 Functions and Functional Dependencies

The methodological fundamentals underlying this thesis stem from the related sciences mathematics and database technology. They are introduced deep enough to understand the principles required for this thesis. More details for the interested reader can be found in the literature of those domains, e.g., in [27], [67], [131].

2.3.1 Mathematical Relations

As defined in 2.2.1, information puts data in relation to other information. Therefore, it is essential to understand the relations between fact types, which are the classification of information used here. The relations between the types of the facts known by an attacker determine whether two VIDs can be linked or a fact set can be increased by deducing new facts. Therefore, a closer look on relations is provided. Only binary relations are of relevance here.

- *Binary Relation:* Generally speaking, a binary relation is an association, which exists or does not exist between two elements of a set [27]. More formally, a binary relation R on two sets A and B is a subset of $A \times B$. Let $a \in A$ and $b \in B$. If $(a,b) \in R$, the relation applies. If $(a,b) \notin R$, the relation does not apply.
- *Function:* A function is a special kind of a relation [204]. A function f from set A to set B is a relation, where 1) for each $a \in A$ there is a $b \in B$ such that $(a,b) \in R$ and where 2) if $(a,b_1) \in R$ and $(a,b_2) \in R$, then $b_1 = b_2$. The set A is called the domain and set B is called the codomain of f. Thus, a function is a relation that completely covers the domain and maps it in a single-valued way onto the codomain.

There are different kinds of functions. The most important ones are explained in the following.

- *Injective function (one-to-one function):* A function is injective, iff whenever $f(a_1) = f(a_2)$, then $a_1 = a_2$, i.e., there are no two different elements of the domain, which are mapped onto the same element of the codomain.

- *Surjective function (onto function):* A function is said to be surjective, iff for every $b \in B$ there is an $a \in A$ with $f(a) = b$, i.e., every element of B can be reached by the function from at least one element in A.
- *Bijective function:* A bijective function is injective and bijective.
- *Composite function:* Let f be a function from set A to set B and let g be a function from set B to set C. Then, the composition of functions g and f is $g(f(a))$ with $a \in A$ and a result $c \in C$. A relation, which is the composition of two functions, still is a function with the properties described above.

Figure 2.9 illustrates the definitions of the different functions.

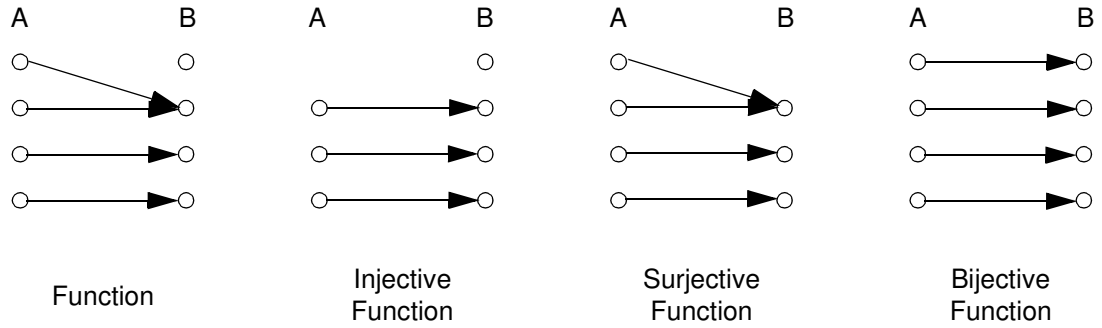


Figure 2.9: Classification of functions

Functions are the fundament for understanding functional dependencies. Functional dependencies are the essential concept underlying the vulnerabilities regarding the VID approach. The following section explains them and shows their importance for VIDs.

2.3.2 Functional Dependencies

For understanding under which conditions it is possible for an attacker to extend his knowledge about a certain perceived identity, it is necessary to have a closer look at database technologies. There are two possibilities of knowledge interpretation being relevant here. This is first inference of new facts from already known facts and second linking fact sets, that are observed in the context of different VIDs, i.e., merging fact sets from different VIDs of the same user. Both interpretations exploit the concept of *functional dependencies*, which is known from database design. In [67] and [131], the authors explain this concept with the example of a relational database.

A relation schema of a relational database defines the tables and the contained attributes. A tuple is an instantiation of the schema with one specific value for each attribute. A functional dependency is a constraint on two attributes. It constrains the possible tuples that can be formed. For this thesis, the attributes correspond to fact types and a functional dependency relates two fact types. The dependency defines all possible legal instantiations of tuples formed from the cartesian product of those two fact types, i.e., a specific relation between the fact types.

A functional dependency is denoted as $X \rightarrow Y$ with X and Y being the considered two attributes in the schema or the two considered fact types. In a legal tuple, the value of the Y attribute (fact of type Y) is uniquely—or functionally—determined by the X value in the tuple (fact of type X). This is analogous to mathematical functions introduced in 2.3.1 with

X and Y being sets whose elements are related to each other. There, a given element from the domain $x \in X$ uniquely determines the corresponding element in the codomain $y \in Y$. There can be no other element in the codomain, on which the given element in the domain can be mapped to by the considered function.

Typically, functional dependencies are an aspect of the semantics underlying the database schema. Knowledge about the semantics tell the database expert, which functional dependencies hold on the extensions, i.e., on the stored tuples, of the database schema. Thus, functional dependencies are a definition of legal extensions and specify constraints that must hold at all times. Functional dependencies are a property of the relation schema itself and not a property of a particular legal relation state. Here, this means especially that a functional dependency between fact types holds true for all possible instantiations of the fact types, i.e., it is a type-based rule to interpret facts.

In [231], the authors show, that functional dependencies are not only existent on schema-level but also sometimes on the level of the actually stored data. In those cases it is possible to infer additional information from some specific stored data instances, although this is not in general true for all instances of the given data type. Therefore the database expert cannot model the dependency on schema-level defining the attributes—or fact types—only. This issue is especially important for privacy considerations, when the focus is not on specification of legal system states, but rather on possible inference vulnerabilities of the stored data. Such relations are called value-based relations or instance-based relations in opposite to type-based relations on schema-level [159].

Two properties of functional dependencies are important for this thesis—for inference and for linking of fact sets of one user:

1. Inference: A functional dependency between two attributes X and Y in a relational database, or between two fact types X and Y in this thesis, defines the corresponding value of attribute Y or of the content field of fact type Y if the value of attribute X or of the content field of fact type X is known. Thereby the timestamp of the new fact is identical to the timestamp of the originating fact.

   Compared with deduction introduced in 2.2.3, a functional dependency between fact types X and Y defines the generic rule for deducing a new fact of type Y from a given fact of type X. The rule functionally determines the concluded fact type from the originating one.

   Such a function can be a complex set of operations including actions like consulting a telephone book. The term of function comes from the properties of the relation between the types. This should not mislead the reader to only think of simple equations here.

2. Linking: If the values of an attribute X in two given tuples are identical, then also the values of a functional dependent attribute Y must be identical. Or in the terms of this thesis: If two facts of type X are identical, then also the two facts of a functionally dependent type Y must be identical, because there is no other fact of type Y, on which the given fact of type X can be mapped to.

For VIDs, the latter means that if an attacker observes in the context of two VIDs an identical attribute, which functionally determines the user, then the attacker also realizes that the

user underlying both VIDs is identical. In the example of section 1.2, such an attribute is the IP address, because each IP address is uniquely associated with one user.

The thesis aims at providing protection of the VID approach in IP-based mobility management. While the methodological base is complete with this section, the next section gives the necessary background in mobility management.

2.4 Mobility Management in IP

Mobility can refer to different aspects in networking. The most common one being also known from mobile telephony is host mobility, where the Internet hosts change their network attachment point. This thesis considers host mobility. Other aspects like session mobility, user mobility, or service mobility are out of scope.

The change of the network attachment point implies two important tasks to be fulfilled by a mobility supporting network. The first task is to locate a mobile host, when another host wants to establish a communication to this host. The second task is to maintain the communication during a move of a mobile host.

Section 2.4.1 gives an overview of mobility management solutions. Section 2.4.2 introduces Mobile IPv6 as the standard solution on network layer. Mobile IPv6 serves as the base of this thesis.

2.4.1 Overview

The mobility challenges can be solved on different layers of the OSI stack, e.g., layer 2, layer 3, layer 4. Layer 2 solutions are technology specific and usually are used in local access networks with only one technology. Introduction of a new technology requires reinventing mobility management in this new technology. Examples for layer 2 mobility management can be found in IEEE 802.11b, GSM, or UMTS. There are also solutions on the transport layer, which are surveyed in [13] or on higher layers, e.g., based on the Session Initiation Protocol, SIP [193] or based on the Stream Control Transmission Protocol, SCTP [61]. Recent approaches often rely on the technological base of peer-to-peer networks, e.g., [235], [71].

As future 4G mobile networks aim at integration of different access network technologies in order to combine the best of all worlds, it is widely acknowledged that IP will become the convergence layer and will be pushed towards the access networks [36], [5], [189], [64], [123]. Consequently, much effort was spent on mobility support on the network layer, which is also the focus of this thesis, other layers are not considered here. Moreover, the thesis restricts itself to IP technology.

There are many proposed IP based mobility management solutions, which are compared and put into relation to each other by several surveys, e.g., [5], [36], [189], [105], [32]. The solutions can be classified into two [5] or three [189] classes. This thesis sticks with the terms of [5], which is based on the notions of macro-mobility and micro-mobility. Macro-mobility refers to interdomain movements, whereas micro-mobility considers movements in one administrative domain. [189] uses the same notions plus the notion of global mobility. Global mobility refers to interdomain mobility, which was named macro-mobility in [5].

Macro-mobility in [189] refers to micro-mobility in [5]. Finally, micro-mobility in [189] refers to movements in one subnet, which is not considered in [5].

Mobile IPv6 is the standard solution for macro-mobility spanning over several domains when using IPv6. It is standardized by the IETF [126]. It has some deficiencies in scalability when considering fast moving users. Thus, Mobile IPv6 is often extended by combining it with a micro-mobility scheme. [84] compares the micro-mobility approach with the approach of Mobile IPv4, which was the predecessor of Mobile IPv6.

According to, e.g., [36], [5], or [105], micro-mobility can be classified in tunnel-based and in routing-based also called host-based schemes. Tunnel-based schemes work conceptually like a hierarchically extended Mobile IPv6. For each mobile node, there is a fixed addressing point, which tunnels arriving packets to the current network attachment point of the mobile node. During movement in one domain, the mobile node only has to update the last hop of the hierarchy, which saves signalling. In routing-based schemes, the mobile node retains a constant addressing point per roaming domain, irrespectively of the current network attachment point. In the domain, the forwarding tables of the forwarding nodes are modified on the movement of the mobile node to ensure reachability.

The base of this thesis is a pure-IP scenario, with Mobile IPv6 as mobility management like used in [64], [123], or [122]. Different access network technologies are integrated by using the network layer as the convergence layer and by pushing IP into the access networks, e.g., into the UTRAN. This results in a network infrastructure, where access routers connect the access network with the core network. Each access router controls one radio cell or Ethernet domain, which directly corresponds to one IP subnet.

Mobile IPv6 was chosen due to its minimalistic change of the plain IPv6 architecture. Thus, the benefits with respect to VID protection can directly be shown without side effects of a complicated mobility management architecture. Moreover, Mobile IPv6 is the default mobility management architecture for IPv6 [51]. The next section introduces Mobile IPv6 on the level, which is required for understanding the thesis.

2.4.2 Mobile IPv6

This section gives an overview on Mobile IPv6. The focus is on the conceptual functionality rather than on details of packet structures or on definition of error cases. Moreover, aspects of neighbor discovery, ICMP message operation and security are neglected, because the thesis concentrates on the core operation of mobility management and leaves those aspects for future work.

2.4.2.1 Mobile IPv6 Functionality

Mobile IPv6 provides for management of macro-mobility. By managing the movement and the addresses on the IP layer, Mobile IPv6 is independent of the underlying layer 2 technology. Thus, it is well suited for managing mobility across heterogeneous access networks with different technologies.

Mobile IPv6 introduces the following terminology: The moving computer is called Mobile Node. The communication partner of the Mobile Node is called Correspondent Node.

The basic problem of a Mobile Node moving through different access networks is the addressing. If the Mobile Node retained its old IP address on a move into a new network, it would be unreachable in the new network, because the address still points to the old network. If the Mobile Node used a new IP address from the new network, it would be reachable, but higher layer protocol connections would be disrupted and Correspondent Nodes could no longer reach the Mobile Node.

Mobile IPv6 solves the problem by assigning the Mobile Node two separate addresses. A constant one from the home network—called home address—and a variable address from each visited network—called care-of address. Higher layer protocols only see the constant home address. Correspondent Nodes can always address the Mobile Node by using the home address.

When the Mobile node is at home, it has only the home address and communicates like in a common IP scenario. Figure 2.10 shows this scenario. The figure shows four networks, the home network, two roaming networks, which the Mobile Node will visit, and the Internet, which connects all the networks. The Correspondent Node is located anywhere in the Internet. The Mobile Node and the Correspondent Node communicate by standard IPv6 means.

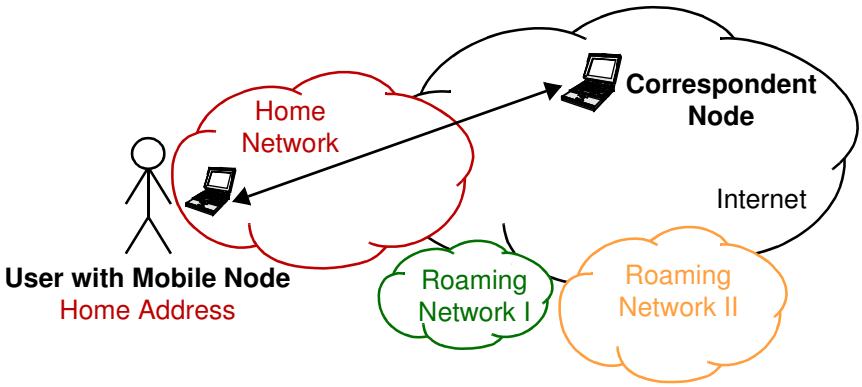


Figure 2.10: Mobile IPv6: The user is in the home network

As soon as the Mobile Node moves away into a new network, it gets a care-of address from the address space of the roaming network in addition to the home address. Figure 2.11 shows this scenario.

The Correspondent Node still sends the packets to the home address. In the home network, a new functionality enters the scene—the Home Agent. The Home Agent registers its layer 2 address for the home address of the Mobile Node when the Mobile Node is in other networks. By this, the Home Agent is intercepting packets from the Correspondent Node destined to the Mobile Node. Those packets are then tunneled to the current care-of address of the Mobile Node using generic packet tunneling in IPv6 [44]. Therefore, the Mobile Node must keep the Home Agent updated regarding its current care-of address.

For communication originating from the Mobile Node, there are two possibilities. The first one is that the Mobile Node uses the care-of address and sends the packets on the direct path from the roaming network to the Correspondent Node. This is the possibility shown in Figure 2.11. In the subsequent operation, it is possible for the Mobile Node to tell the current care-of address to the Correspondent Node, so that the direct path can also be used in the opposite direction. This latter procedure is called Route Optimization. The second possibility for communication originating from the Mobile Node is to send the packets to the Correspondent Node via the Home Agent, i.e., to use this tunnel bidirectionally.

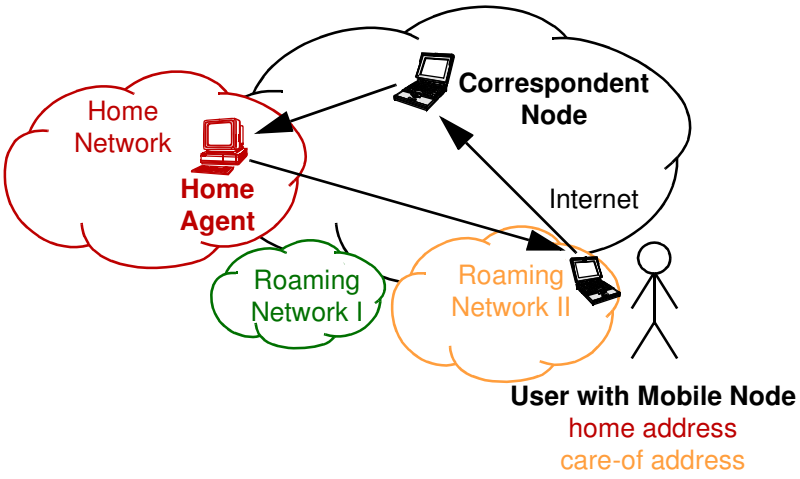


Figure 2.11: Mobile IPv6: The user visits a roaming network

When a Mobile Node moves to a new network, it gets a new address from the address space of this network. There are different possibilities to configure this address, which must be globally routable and unique, i.e., of global-scope. The next section explains the two possibilities of relevance for this thesis.

2.4.2.2 Autoconfiguration of IPv6 Addresses

There are two principle ways of assigning an IP address to a node without manual intervention of an administrator. The first principle way is by a stateful server and is called stateful autoconfiguration. The best known approach is DHCPv6 [62]. This is beneficial when the administrator needs to keep track of the exact IP addresses in the network.

The second possibility is a stateless autoconfiguration. This is beneficial, if it is only necessary that the assigned address is unique. The node here uses locally available information for making the address unique and information about the network for making it routable. In case of a link local address, the network information is a well-known prefix. For global-scope addresses, this information stems from router advertisements.

In stateless autoconfiguration according to [218], at first a link-local address is to be configured. This is done by using the interface identifier of the node, e.g., a MAC layer address, i.e., Ethernet or ISDN, of general length N. The other (128-N) bits are the well-known link-local prefix FE80::0, which is prepended to the interface identifier. This prefix is indicating that the corresponding address is valid on a local link only and is not globally routable. The numbers are hexadecimal and the two colons are an abbreviation for omitted zeros. The interface identifier replaces the rightmost N zero bits. Often, 64 bits are required for the interface identifier. [108] and [115] define a mechanism to create an EUI-64 identifier from a 48 bit MAC identifier: Two octets with FF_{hex} and FE_{hex} are inserted in the middle of the MAC address. Figure 2.12 illustrates such an EUI-64 identifier.

Figure 2.12: EUI-64 identifier created from a 48 bit MAC identifier

By using stateless autoconfiguration according to [218], each care-of address of the Mobile Node will contain the interface identifier. By this, it is easy for attackers to track the Mobile Node even across several networks, i.e., across several care-of addresses. Thus, [164] introduces privacy extensions for the autoconfiguration process.

The extensions base on two principles. First of all, the interface identifier is replaced by a random number. Second, the resulting address is changed over time. The random number is generated by using an MD5 hash. The random number is not only based on a random part, but also on the real interface identifier, which makes it less probable that two different nodes are computing the same random numbers.

Those privacy extensions prevent linking of several care-of addresses to the same interface and thus to the same user by attackers. The privacy extensions do not prevent that the location of the Mobile Node can be determined from the network part of the care-of addresses. Neither do the extensions help to keep VIDs of a user unlinked, because all VIDs still have the identical care-of address.

After each autoconfiguration a duplicate address detection [218] has to be run. For this, the node sends a Neighbor Solicitation message containing the tentative address. If another node already has this address, it answers with a Neighbor Advertisement message. Then, the autoconfiguration is ended and manual intervention is needed. In the stateless autoconfiguration, the duplicate address detection is done with the link-local address. The result is also valid for the global-scope address, because the uniqueness of both addresses is based on the rightmost N bits, which are the interface identifier.

With the fundamentals for understanding the thesis being complete, the next section structures the whole problem area. Moreover, it defines the focus of the thesis this structure.

2.5 Problem Structure and Focus

This section is structured as follows. Section 2.5.1 introduces an overall model of the scenario. After that, section 2.5.2 structures the problem scope and defines the focus of the thesis.

2.5.1 Overall Model of the Scenario

Figure 2.13 shows an overall model of the scenario. A user in the real world on the left hand side has a real identity with a number of attributes, e.g., a hair color or a movement behavior. When using the communication system—or any IT system in more generality—several of the user's attributes are represented in the system, e.g., the movement behavior.

On the right part of the figure, a potential attacker is depicted with its view. The attacker looks at the communication system at several points in time and gains a set of observations. In a next step, the attacker interprets the observations. Thereby, external knowledge can be used, e.g., about the semantics of the system. Finally, the attacker gains a perception of the user in the real world. This is the view, which the attacker finally has on the user and thus is called perceived identity.

The interpretation process can be subdivided into two separate functions as indicated in Figure 2.14. In the first step, the attacker links all observed facts, which are about the same

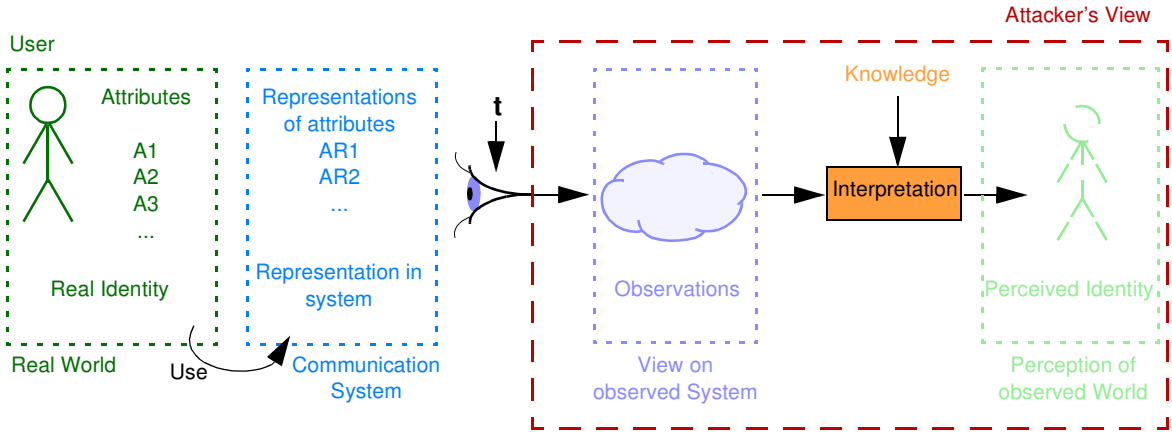


Figure 2.13: Overall Model

user. In the next steps, the attacker tries to conclude new facts from the known facts about the user.

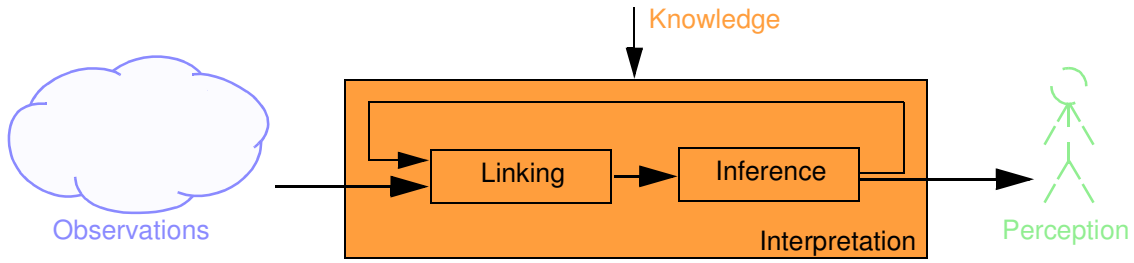


Figure 2.14: Interpretation functions

The attacker can apply the two interpretation functions several times. Figure 2.15 shows an example. In a first linking step, the attacker groups all observed facts about the same user resulting in a set of fact sets. In the following inference step, the attacker infers new facts and thus, increases the sets. Thus, two sets are overlapping and can be merged in a second linking step, resulting in the two final sets on the right hand side. The larger sets might allow for inference of new facts, which might allow for merging other sets and so on. The attacker stops applying the interpretation steps when the knowledge is not increasing any more.

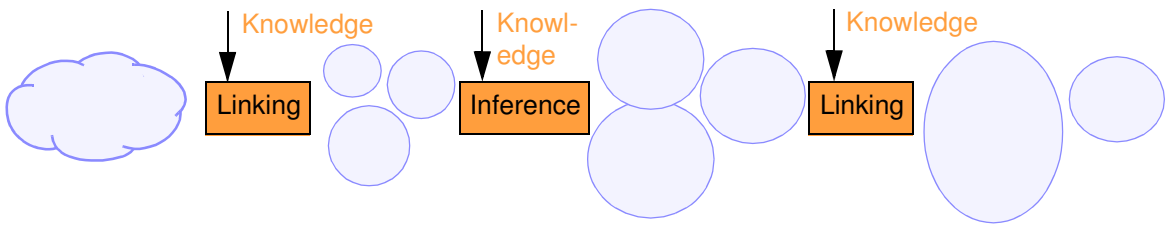


Figure 2.15: Extending knowledge by interpretation

Considering the attacker's goals from chapter 1, it becomes obvious that the increase of fact sets above relates to the extension of VIDs. To increase the PID, an attacker can extend the knowledge about a VID by inferring new facts from the known ones or by merging the known facts about this VID with the known facts about another VID.

Thus, the attacks on the VID approach, which are to be evaluated and against which a system should protect, are a special case of the overall model, depicted here. When the PID embraces several VIDs, it contains their VID-identifiers and values of attributes being contained in the merged VIDs.

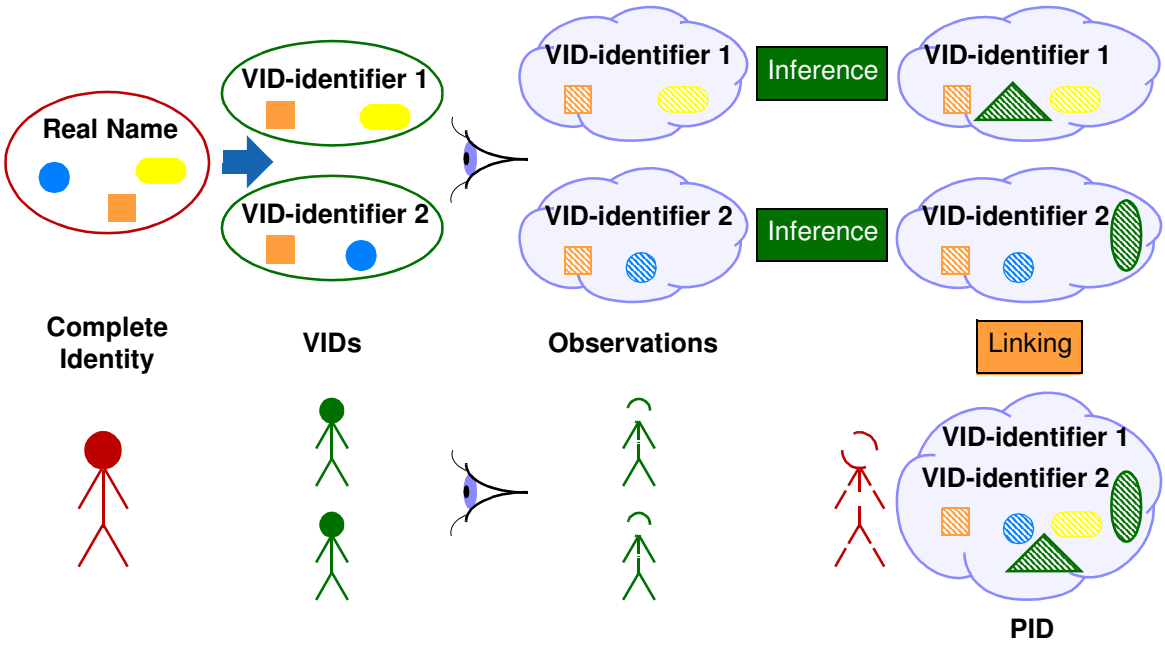


Figure 2.16: VIDs and Fact Sets

Figure 2.16 links the overall scenario from this section with the VID approach explained in chapter 1. On the left side, there is the user's domain, i.e., the part of the scenario, which is under the user's control. On the right side, the attacker's domain is shown.

The figures below the ellipses and clouds represent the principle: The user splits the real identity into several virtual identities, each containing only a part of the real identity's information. The attacker observes the virtual identities resulting in an incomplete picture of the virtual identities. Then, the attacker tries to build up a representation of the user as close as possible to the real identity by interpreting the observations. In the following, this process is described in more detail.

By using the system under consideration, the user's attributes become represented in form of attributes of the system. The user defines, which attributes are contained in which VID. Each VID gets a VID-identifier for addressing on application layer. Some attributes are contained in only one VID, e.g., the blue circle, whereas other attributes are contained in several VIDs, i.e., the orange box. The user's intention is to show potential attackers only one of those VIDs.

The attacker observes the system with its attributes and gains observations, e.g., if the user is doing business with the attacker's application or if a VID's communication is handled by a machine of the attacker. The observations are facts about attributes of the system, e.g., about IP addresses, which the attacker now knows. In the example of Figure 2.16, the attacker can observe both sketched VIDs and can observe all attributes of the VIDs. Observations in general contain less aspects than the real attribute, because they are abstracting for the intended use of the observations. Therefore, the observed facts are hatched. The observations build fact sets, from which the attacker knows, that the contained facts are about one user.

Then, the attacker interprets the knowledge. At first, the observed fact sets are increased by inferring new facts. These are the green facts in the figure. After that, the attacker links the increased fact sets based on the identical fact represented by the orange box. This results in the final view the attacker has on the user, i.e., the perceived identity. The PID in this example contains two VID-identifiers, because facts from two VIDs are merged.

The next section defines the scope of this thesis.

2.5.2 Problem Scope

The goal of this thesis is about preventing attackers from increasing their knowledge by exploiting vulnerabilities of the communication system. An important step for increasing the knowledge is the interpretation of known facts. Thus, the interpretation possibilities are structured in this section. By defining the considered dimensions, the assumed attacker strength in this thesis is defined, because the dimensions relate to different capabilities of the attacker. There are basically four dimensions:

- Type-based vs. value-based interpretation

  For the protection approaches of this thesis, it is important, whether an attacker is able to interpret the existing knowledge by applying rules. The existence of an interpretation rule can be defined on a type-basis or on a value-basis. Some interpretation rules exist for every possible instantiation of a fact type. They are called type-based rules. Other interpretation rules only exist for certain instantiations or values of the content field. They are called value-based rules.

  This thesis is based on type-based evaluation. Thus, the statements are valid for every possible instantiation of the system and it is possible to reason about the protection already during design stage of a system, when the concrete values are not known yet [182], [161], [162]. For the inferences, functional dependencies, cf. 2.3.2, are considered. More complicated inference channels like, e.g., multi-valued dependencies [131], [67] are out of scope.

- Certain interpretation vs. uncertain interpretation

  The attacker can either decide to only consider interpretations, that result in a 100% correct result. Another possibility is also to reason on uncertain information and possibly even with uncertain interpretation rules.

  In this thesis, only certain interpretation of certain information is considered, because uncertain interpretation would introduce a level of speculation making the results unsure and discussable.

- Known vs. unknown data structure

  The attacker can either know the structure of the observed and interpreted facts or not. If the data structure is known, it is assumed that the attacker knows the semantics of the facts.

  In this thesis, a known data structure is assumed. Attackers, who observe a technical system in order to spy out a user will most likely know how the system works and thus know the meaning of the observed facts and how they are to be interpreted.

- Snapshot vs. history

  The attacker might just record a snapshot of the state of the system and reason about the contained knowledge. Another possibility is to record a series of snapshots, i.e., a history of observations.

  In this thesis, a history is assumed, because there is no reason besides storage capacity, why an attacker could be assumed to be capable of observing one single snapshot, but not of observing a second one.

Also the section of the world, which will be considered in this thesis has to be defined. First of all, the focus is on the mobility management building block of the communication system, where Mobile IPv6 is chosen as technology. Thereby, the layers above and below IP are not considered, except in well reasoned exceptions.

Second, the scope is restricted to privacy questions, i.e., unwanted revelation of information about the user's personal affairs. This can be direct revelation of personal information or revelation of information leading to violation of personal information.

In [50], personal data is defined implying that the user can be identified from it, i.e., that the user's real identity is known. This requirement is not followed in this thesis, because it is considered as looking too short. In future IT systems, the user is using so many applications, that the virtual identity in the IT world is already sensitive. By knowledge of sensitive information, an attacker can do considerable harm to a user in the IT world. For this, the user's real identity does not necessarily have to be known to the attacker. Nevertheless, the harm can be extended to the real world, if the attacker knows the user's real identity.

The personal information, which is considered in this thesis, is contained in variables, constants, or in the semantics of the communication system. Information outside the system, i.e., information which the user discloses via other communication channels than the considered system, is out of scope.

Chapter 3

Threat Analysis Methodology and its Application to Mobile IPv6

This chapter describes the developed evaluation methodology [99] as well as the actual evaluation of Mobile IPv6. The methodology is defined in the form of a procedure model. Usually, procedure models are on a conceptual level in order to be applicable to many different systems with some common properties. During the single steps, the evaluator has to provide for intelligence and decisions. This is a difference to fine granular formal methods, which can be automated by pure machine processing, but which are only applicable to restricted problem spaces due to the low level of abstraction. The low abstraction level causes a big complexity with many dependencies and states to be considered [198].

After section 3.1 has defined the goals, section 3.2 explains how to build a conceptual information model of the system to be evaluated. Then, section 3.3 shows how to use this model for an evaluation regarding threats and vulnerabilities to the VID approach. During its explanation, the theory is accompanied by a concrete application to Mobile IPv6. Section 3.4 sums up the evaluation result. Finally, section 3.5 discusses related work to the presented methodology.

3.1 Goals

When designing a communication system for protection of a user's private sphere, the relevant question is, which knowledge a potential attacker can gain about the user's private information. The answer depends on the semantics of the revealed data. Thus, the answer requires a view on the communication system from a knowledge perspective. A privacy engineer must analyze the semantics of the data, which is stored at a potential attacker, is communicated to a potential attacker, or can be observed by a potential attacker in any other way. Thus, the information contained in the observed data has to be analyzed [18]. This holds also true for the specific privacy question about threats to the VID approach, which is underlying this thesis.

Today's communication systems consist of many subsystems, protocols, databases etc. In order to evaluate the threat to a user's VIDs by an existing system, an evaluator has to analyze all those parts. When designing a new system or extending an existing one, there are usually several design alternatives, which lead to different systems. There, the evaluator has to analyze those solutions regarding their intrusiveness to the VID approach. In both activities, this evaluation of many architecture parts results is a recurring task for the privacy engineer.

In an evaluation, it is important that the results of the evaluations are comparable, traceable, reproducible, and as objective as possible. Moreover, the privacy engineer has to be sure not to forget any evaluation aspect. Finally, protection often implies new problems, so that in the end it is difficult to state whether the overall protection has improved or not. Those goals are difficult to achieve without a clear and to a certain degree formalized methodology—for evaluation as well as for designing improvements of the system.

The following sections describe the evaluation methodology. The methodology for improving systems is in chapter 4.

3.2 Knowledge Model

For an analysis of a communication system regarding VID vulnerabilities, the evaluator must analyze, which knowledge potential attackers can gain by observation and how they can interpret this knowledge. Therefore, the evaluator creates a model, which puts the relevant pieces of information of the attacker's knowledge in relation to each other. This section describes the modelling process. First 3.2.1 defines the miniworld of interest. Then, 3.2.2 describes how to build the model and actually constructs the model of Mobile IPv6.

3.2.1 Miniworld of Interest

In order to define the relevant miniworld, it is necessary to recall the goals of the thesis. Pieces of information that are observable by potentially malicious parties shall be evaluated with respect to vulnerabilities of the VID approach. This evaluation shall be done during the design phase of a new communication system. The results shall be comparable, traceable, reproducible, and as objective as possible.

The goal is not a physical storage of the model in a real database. Therefore, the conceptual layer of data modelling is suitable. It is necessary to express the way in which pieces of information relate to each other. To reach this aim, static integrity constraints, to which the real system always adheres, are used.

In general, models represent artifacts of the real world, e.g., in an information system. Here, the real world artifacts are already pieces of information in an information system. The modelled entities are fact types and the relationships between the entities describe the relations between the fact types. The goal is not to analyze a given database at runtime, but to analyze a system in specification state that does not handle any concrete data, yet. Thus, it is not possible to estimate all possible instantiations of facts. Therefore, the evaluation will consider the level of the fact types. This is a common rationale also followed, e.g., in [161] and [162].

The evaluator has to consider those fact types that are observable by potential attackers. From these fact types, only the ones containing personal information or violating indirectly protection of personal information have to be considered. A fact type can violate protection indirectly by either allowing to infer a fact type with personal information or by allowing to link several fact sets.

In order to achieve objective and comparable results, only relation properties that are 100% certain are considered. In research literature, especially in the area of artificial intelligence, science is aiming at operation on uncertain data. However, there is not yet a general methodology established that suits all needs of different application domains and that always yields intuitive results. If the model considered uncertain relation properties, different vulnerabilities would be resulting according to the chosen methodology of operating on uncertain data. Thus, the results would be neither objective nor comparable.

It is important to note, that the proposed methodology considers an intelligent privacy expert creating the model. This expert can decide to include specific uncertain relations, if the expert rates them as too dangerous to ignore. This is usually the case, if either the probability of the existence of the relation is high or if it could lead to a high damage in case that the relation is ignored.

An important aspect of the necessary miniworld to evaluate vulnerabilities to the VID approach are the properties of relations between fact types. If there is an N:1 or a 1:1 relation between a fact type A and a fact type B, there exists a functional dependency $A \rightarrow B$. According to 2.3.1, a potential attacker then can infer a new fact of type B from a given one of type A. For computing the resulting fact, the attacker needs to know the mapping of the function besides the knowledge that there exists a function. Moreover, if attackers know two identical facts of type A, they can conclude that also the functionally dependent facts of type B must be identical. The latter conclusion is possible even without knowing the mapping of the function. If A and B participated at an M:N relation, those conclusions would not be possible, because the relation between a fact of type A could refer to different facts of type B.

Therefore, the model must reflect two aspects. First, the existence and cardinality of the relations between fact types must be visible. Second, the model must show whether the definition of the relation, i.e., the mapping in case of a function, is known by an attacker.

After knowing the relevant miniworld, the next section describes how to map the real world into the model.

3.2.2 Modelling

This section introduces the general procedure of the modelling task and applies the methodology to model Mobile IPv6 as an example. The description starts by explaining which fact types are to be included in the model and how the relations between them are designed. The focus here is on the so-called elementary fact type view on the model. Then, the protection goals and the assumptions for the actual evaluation of Mobile IPv6 are defined. Afterwards, the elementary fact type view is built for Mobile IPv6. Finally, the handling of variable fact types is described and the resulting so-called dynamic view on the model is built for Mobile IPv6.

3.2.2.1 Methodology for Creating the Elementary Fact Type View on the Model

First of all, evaluators have to model the fact types, whose facts can be observed by potential attackers. Attackers can observe facts which are either stored in their databases or forwarded by their machines. Second, evaluators have to extend the model with fact types that can be inferred by functional dependencies. Third, they must include a fact type for the user as core entity of privacy evaluations. The user here is represented by the real world name, i.e., a fact type "real name".

Fact types build the entities of the entity relationship model. Besides the entities, the evaluator needs to model those relationships that represent functions, i.e., relations between the determined fact types with either a 1:1 or an N:1 cardinality ratio. N:M relations are not representing functional dependencies and are thus not used for any evaluation step. Therefore, they can be omitted.

In order to determine the cardinality ratio of the relations, the evaluator does not have to gather specific database knowledge. There are two simple rules, which can be used interchangeably. The rules to determine the cardinality ratio between fact type A and fact type B are a direct result of the properties of functions and functional dependencies.

1. If each fact of type A relates uniquely to one fact of type B—and not ambiguously to several possible facts of type B—A functionally determines B. Then, the cardinality on the side of B is a "1" and the relation receives an arrowhead at the side of B in the graphical representation.
2. If identity of two facts of type A implies identity of the corresponding facts of type B, then A functionally determines B. Therefore, the cardinality on the side of B is a "1" and the relation receives an arrowhead at the side of B in the graphical representation.

Here, a snapshot of the system is considered, i.e., one single point in time. If not doing so, there would never be a functional dependency between a constant type A and a variable type B, because there would be different facts of type B on which the single fact of type A will be mapped at different points in time.

If A does not functionally determine B, there will be an "N" on B's side of the relation, i.e., no arrowhead in the graphical representation. Each pair of fact type A and fact type B has to be evaluated in both directions to determine the relation in both directions. The evaluator is free to choose from both rules. Depending on the actual fact types one or the other may be more intuitive.

Finally, the evaluator must determine for each function, whether a potential attacker knows the mapping. This is defined by the attacker model, which is determined by the evaluator. If the knowledge can be gained easily, e.g., by a simple lookup mechanism or by technical knowledge about the system, the evaluator must assume that most attackers will be able to gain the knowledge. In general, it is not wise to assume that attackers are not able to learn how the system will technically work. If the attacker is assumed to know the mapping of the relation, the arrowhead in the graphical representation is solid.

It might occur that attackers can aggregate heterogeneous fact sets, which consist of facts of different fact types. Such heterogeneous fact sets might impose additional vulnerabilities as compared to consideration of only fact types, whose instantiation is a single fact. In those cases, the evaluator models the fact sets as a new fact type and includes this fact type in the

model. This handling allows for sticking with the simple modelling technique and binary relationships.

After having built the model, evaluators can decide to omit certain fact types. Fact types, which are neither usable to infer sensitive fact types nor uniquely mapped to the real name type, can be omitted.

The fact type model is dependent on the attacker model chosen by the evaluator. The reason is that different attackers might observe different fact types, might know different relations, or might know the mappings of different relations.

The next sections create this first view consisting of elementary fact types, i.e., not containing updates, tracelets, and traces, which are handled in sections 3.2.2.5 and 3.2.2.6, on Mobile IPv6. For this, at first protection goals and assumptions must be defined in the following sections.

3.2.2.2 Protection Goals for Evaluation of Mobile IPv6

A sound specification of protection goals is the base of each privacy evaluation. This thesis focuses on the protection class confidentiality. The definition of the sensitive assets generally requires a human evaluator, who can rate the potential risk of disclosure of certain facts or fact sets. The protection goals are motivated by the high-level protection goals of protecting the VID approach. The VID approach is violated, if either the attacker can disclose more sensitive information than the user is intending to show, or if the attacker can link several VIDs as belonging to the same user.

Here, the high-level protection goals translate to protection of confidentiality of the following facts and fact sets:

- Fact F1 Home provider

  The fact of the home provider, i.e., of the primary provider of the user, might allow for estimating the user's preferences, because in environments with several suitable providers, the choice of the provider is based on user preferences, e.g., the cheapest price or the best quality. Whereas providers of roaming networks are often determined by contractual relationships between those providers and the user's home provider, the user usually is free to choose the home provider.

- Fact F2 Home location

  The home location, i.e., the location where the user is at home, might contain sensitive information in certain cases, e.g., it allows for discrimination of users according to their nationality. Although the home communication provider is not always the physical home of the user, this might be assumed by attackers, because it will be true for the majority of users.

- Fact F3 Real name

  The user's real name must not be revealed. This is the original intention of pseudonymous communication.

- Fact F4 Personal attributes

  The communication system must not allow for observation or inference of personal attributes. F4 does not include the real name, the home provider, the home location, and the location. The evaluation treats all of them explicitly and in more detail.

- Set S1 Large location trace

  The term location here denotes the roaming location. The communication system does not disclose the real name and personal attributes other than locations and providers. Nevertheless, a long trace of roaming locations, i.e., locations of networks in which the user is roaming, often allows for inferring the real name or other personal attributes, because the roaming location often allows for inference of a user's activity or identity—which is also the reason, why a large location trace is sensitive even without any VID-identifier being contained in the fact set.
  This conclusion of F3 and F4 is the only way how F3 and F4 can be revealed here, because they are not contained in any other information of the system under consideration. If it is avoided that any potential attacker can see a large location trace, F3 and F4 are inherently protected. It is thus not necessary to consider F3 and F4 further in this thesis, because S1 is considered.

- Set S2 Location and VID-identifier

  The term location here denotes the roaming location. Protection of set S2 prevents potential attackers from knowing the location of a virtual identity. Although not the user's real name is revealed, this fact set is sensitive. Knowing compromising locations of a virtual identity allows for discrimination in the virtual world. This can be very cumbersome to users.

- Set S3 n VID-identifiers

  A fact set containing several VID-identifiers would directly undermine the privacy approach of using several VIDs.

Single location facts are not included in this list on purpose. A single location fact without anything else, e.g., a VID-identifier, is not sensitive, because no attacker will know what this location is about.

3.2.2.3 Assumptions for Evaluation of Mobile IPv6

Like in each modelling technique, assumptions allow for simplification and abstraction of the problem. Changing assumptions would change the result, in general. Therefore, assumptions must be explicitly stated in order to show potential readers, whether the result is applicable to their concrete questions or whether they have to adapt the evaluation.

This section defines the assumptions of the evaluation of Mobile IPv6. The evaluation focuses on the core part of the protocol. It does not consider related parts like the neighbor discovery procedures or the ICMP messages, which control things like solicitation of the prefix of a link or agent advertisements.

The following assumptions originate from the protection goals. Sensitive information is contained in the following fact types:

- The real name
- The geographical roaming location as well as traces and tracelets of such locations

- The provider of the home network
- The location of the home network

The following assumptions originate from the considered system under evaluation as well as its configuration and usage:

- A user is using exactly one device.
- Each device has exactly one interface.
- Each interface has exactly one MAC address
- Each interface has two IP addresses—one home address and one care-of address.
- A user is addressed outside the communication system by VID-identifiers.
- Each VID—and thus each VID-identifier—is used for exactly one set of application data.
- Each application is used with exactly one VID—and thus one VID-identifier.
- The IP addresses are built without including the MAC address, e.g., like described in [164].
- No IPSec [132] is used.
- Mobile IPv6 is used with route optimization.

The following assumptions originate from the considered attackers and define the attacker model in a general way:

- Providers of the agents of the communication system are potential attackers.
- The communication partner is a potential attacker.
- Eavesdroppers are potential attackers.
- Every attacker knows the system, the configuration and the usage like described above.
- Every attacker knows the mapping to get the user's home address from the VID-identifiers. This is necessary to reach the VIDs via the network.
- No attacker knows the mapping to get one or several VID-identifiers from the home address.
- Every attacker knows the mapping to get the network corresponding to an IP address.
- Every attacker knows the mapping to get the geographical location corresponding to a given network.

  This is in many cases easy to determine, e.g., by [112], [118], [119]. If subnetworks will become smaller, this information is supposed to become even more accurate.
- Every attacker knows the mapping to get the provider corresponding to a given network.

3.2.2.4 Elementary Fact Type View on Model of Mobile IPv6

This section describes the creation of the first view on the model of Mobile IPv6. Throughout this thesis the only difference between the considered attackers is that they can observe different fact types. They all know the same relations and the same mappings. Therefore, it

is possible to start by a full model containing the sensitive fact types, whose facts can be observed by the superset of all considered attackers. Later during the evaluation step, the evaluator cuts out the part of the model, which corresponds to the fact types that the actual evaluated attacker can observe.

Data revealed by applications is not part of the communication system and therefore not considered in detail. It is included in the model in form of a fact type "application data" in order to show the link to the applications. This fact type corresponds to the data being revealed by all applications, which the user uses with one VID. A potential model of applications could be linked here to the model of Mobile IPv6 for an overall evaluation. In the evaluation, the application data is not considered.

The same approach is chosen for lower layers. They are represented by a fact type "MAC address". It provides the hook for evaluations, e.g., of the ARP protocol [177], the Neighbor Discovery Protocol [163], or of any wireless channel identifiers.

Figure 3.1 shows the resulting model. It contains only the model for one VID—represented by its VID-identifier. This is sufficient to analyze the threats, because the model looks the same for all other VIDs. Another VID is only indicated in a dashed way. It refers to different application data, but to the same real name and home address like the evaluated VID. So do all other VIDs of this user.

Each VID-identifier is uniquely associated to one user. Each user can have several VID-identifiers. Therefore, there is an N:1 relation between the VID-identifier and the real name. This functional dependency cannot be computed by anyone. Therefore, the arrowhead is empty.

To each VID, a set of application data is uniquely associated. Each set of applications is used with only one VID. Thus, a 1:1 relation connects those fact types. No attacker is able to derive the application data from the VID-identifier or vice versa. Therefore the arrowheads are empty.

According to the configuration assumptions there are 1:1 relations between the user, the device, the interface, the MAC address, and each IP address. There is no plausible way, how attackers could know the mappings of those relations. Therefore, all arrowheads are empty. The model shows that no other fact types are related to the device or the interface. Thus, the evaluator can simplify this chain of functions by omitting the device type and the interface type. This is possible, because the properties of composed functions remain the same, cf. 2.3.1. Then, the MAC address is directly related to the real name fact type. The remainder of the thesis will use this simplification.

Users only have one home address. It must be used for all VIDs. If the VID-identifier is the same in two fact sets, also the corresponding home addresses are the same. There is a functional dependency in this direction. An identical home address instead does not allow for the conclusion of identical VID-identifiers. Thus, there is an N:1 relation between the VID-identifier and the home address. The assumptions state, that attackers can infer the home address from the VID-identifier. Therefore, the arrowhead is solid.

Each IP address corresponds to exactly one network, but each network corresponds to many IP addresses. This results in an N:1 relation. Each network is at exactly one geographical location, but at one location there can be several networks. This also results in an N:1 relation. According to the assumptions, attackers can identify the networks from the home

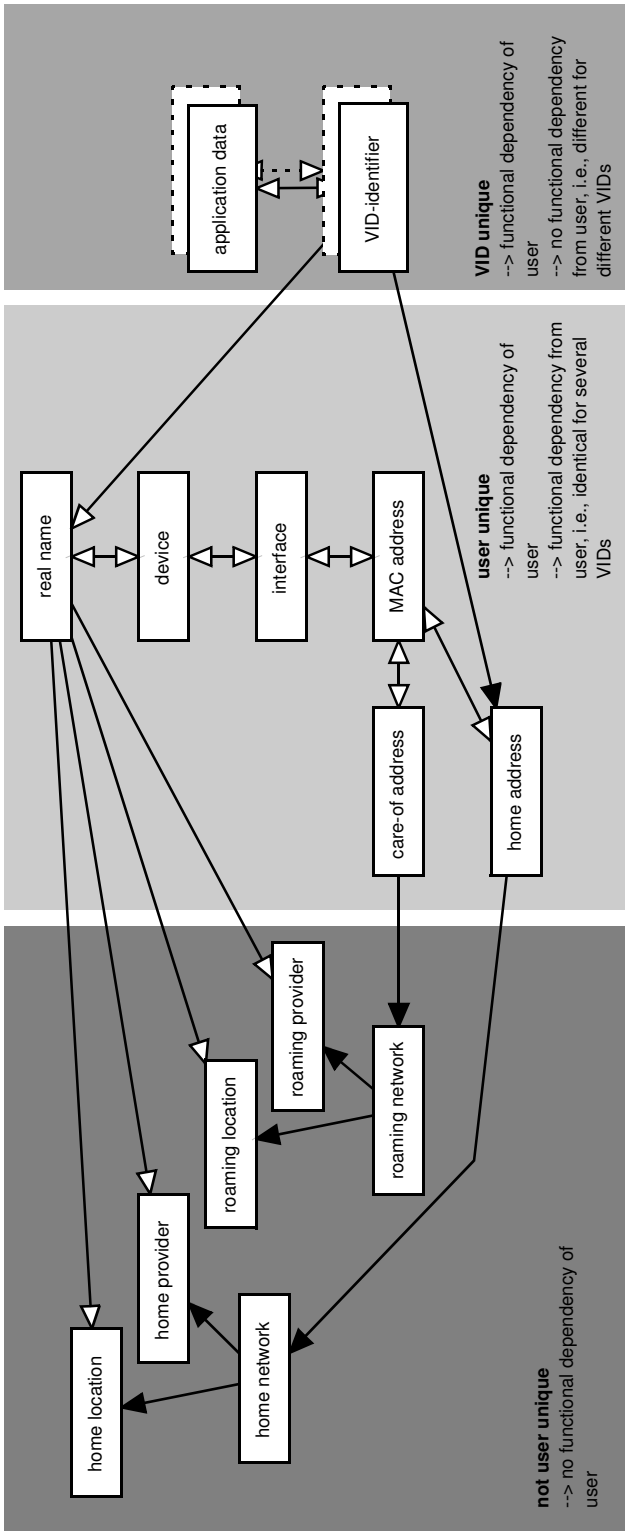


Figure 3.1: Elementary fact type view on Mobile IPv6

address and from the care-of address. Also, attackers can gain the geographical locations as well as the providers of both networks. Therefore, both arrowheads are solid.

The grey boxes subsume equivalent fact types from the perspective of their relation to the user and to VIDs. Fact types in the middle box, which is light grey colored, are unique for one user. Thus, they functionally determine the user and are identical in fact sets of all

VIDs. Fact types in the right box, which is colored in middle grey, are unique for one VID. Thus, they neither determine the user nor are they identical in fact sets of different VIDs. Finally, fact types in the left box, which is colored in dark grey, are neither unique for one user, nor unique for one VID. They also do not functionally determine the user.

Each grey box subsumes fact types, which have equivalent properties from a VID linking perspective. The types in the middle grey box are varying with respect to their properties for linking tracelets.

All relations from fact types of one grey box to fact types of another grey box are of equivalent cardinality. The relations into the dark grey box are all of nature 1:N. Those relations reflect the anonymity set of users or VIDs respectively which are possibly the originator of facts of the types in the dark grey box. The relations from the middle grey box to the light grey box are of N:1 cardinality, because the user unique fact types in the light grey box are visible in the context of several VIDs, whose types are in the middle grey box.

It is possible to have an abstracted view on the model based only on the grey areas. This view is indicating that a user as abstraction of the types in the light grey area shares a 1:N relation to VIDs as abstraction of the types in the middle grey area. Moreover, users as well as VIDs are sharing an N:1 relation to networks as abstraction of the types in the dark grey area. This reflects the configuration in which one user has several VIDs. Both are connected to one network at a time. Each network serves several users and thus several VIDs at a time.

Variable fact types need a further, additional treatment. This is described in the next sections.

3.2.2.5 Methodology for Creating the Dynamic View on the Model

Variable fact types need a special treatment for two reasons. First, there exist different facts with different values of the content field and different timestamps of variable fact types. The changes of the system attributes, which are modelled by a variable fact type are triggered by update events. Such update events can reveal additional information. Therefore, for each variable fact type a corresponding update fact type, whose facts contain this revealed information, must be introduced in general.

Update events, often correspond to events in the real world. Moreover, they are often dependent on each other, i.e., one real world event triggers updates of several system attributes and thus corresponds to several update fact types. Instances of an update fact type are a fact set containing two instances of the corresponding elementary fact type: The old fact and the new fact after the update. The update fact type has an additional timestamp stating when the update took place.

The second peculiarity of variable fact types is, that often an attacker who can observe one fact of a variable fact type can also observe several facts of this fact type. This is captured in the model by introducing tracelet fact types and trace fact types. They represent fact sets containing facts of the same fact type with different timestamps. The following discussion holds true for traces and tracelets. For readability reasons, only traces are named in the text.

The elementary variable fact type is surrounded by other fact types in the elementary fact type view of the model. The elementary fact type view shows the relations of the elementary variable fact type to those surrounding fact types. The update and the trace fact types of an elementary fact type both relate to their surrounding fact types like the elementary fact

type itself. This is so, because only type-based functional dependencies are of relevance here. Those functional dependencies hold true for all possible facts of the given fact types and thus also for sets of such facts.

An instantiation of a trace fact type is an aggregation of facts of the same fact type. A trace functionally determines all facts, which are functionally determined by the contained elementary facts. Thus, the trace fact type functionally determines the same fact types like the corresponding elementary type. Because the instantiation of a trace may contain more information than a contained single fact, the trace can allow for inference of additional facts by a functional dependency. These additional facts contain the information, that can be inferred from the aggregate of facts in the trace.

Like traces, updates are also an aggregation of facts of the same type. Thus, the same argumentation is valid for update fact types and the relations to fact types of the surroundings in the model.

Both peculiarities—updates and traces—are patterns. They are common for all variable fact types and thus can be treated in a standardized way for modelling and subsequent evaluation. The next two sections describe the patterns in detail.

Updates

Figure 3.2 shows the pattern for the update fact type (grey) of an elementary variable fact type (orange). An instantiation of the update fact type uniquely determines the contained facts of the elementary fact type. Thus, the update fact type functionally determines the elementary fact type. Each fact of the elementary fact type can be contained in several instantiations of the update fact type, because from one originating fact of the elementary fact type, different updates to different new facts or at different points in time are possible. Thus, the elementary fact type does not functionally determine the update fact type. Therefore, there is an N:1 relation between the update fact type and the corresponding variable fact type. Every attacker is assumed to be able to extract the single elementary facts from the update fact. Thus, the arrowhead is solid.

There are three categories of fact types surrounding the elementary fact type and the corresponding update fact type in the model:

- Constant fact types
- Variable fact types modelling system attributes, which are not triggered by the same real world event as the system attribute being modelled by the considered elementary fact type
- Variable fact types modelling system attributes, which are triggered by the same real world event as the system attribute modelled by the elementary fact type

Fact types of the third category are said to be dependent on each other, because one update functionally determines the other updates. Fact types of the second category are called independent, because they are triggered by independent real world triggers.

Fact types of the first two categories are depicted on the left side of Figure 3.2. They are aggregated to one representative fact type "constant or independent variable surrounding fact types".

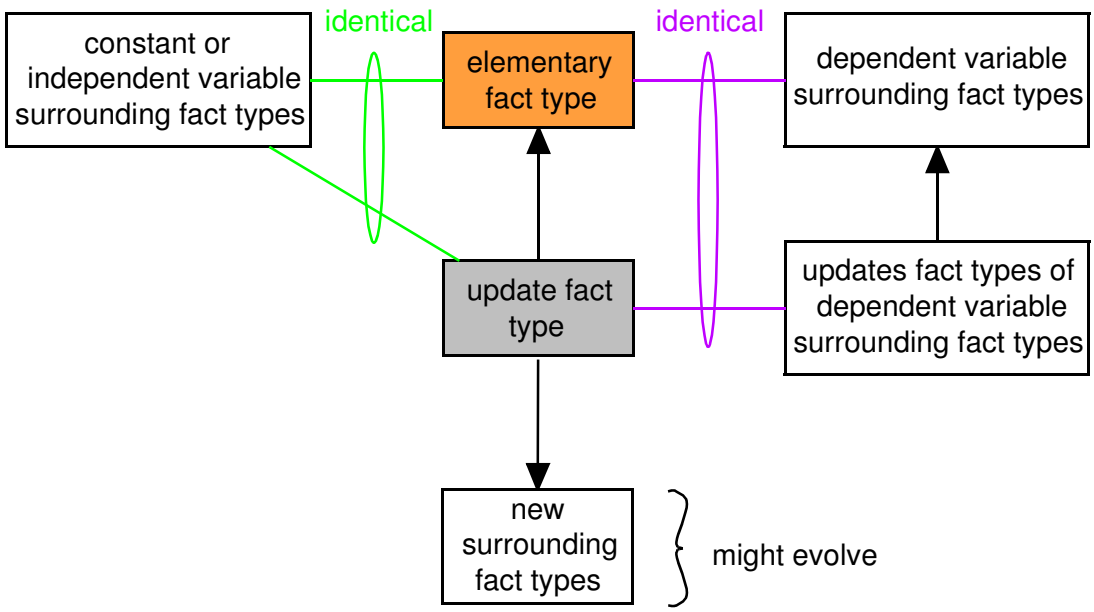


Figure 3.2: Pattern of variable fact types

The fact type "dependent variable surrounding fact types" on the right side aggregates all dependent variable fact types with the same trigger. The (grey) update fact type of the considered (orange) elementary fact type relates in the same way to the update fact types of the dependent variable surrounding fact types like the considered (orange) elementary fact type relates to the dependent variable elementary fact types. For those dependent update fact types, the same pattern applies. Therefore, those update fact types are related in an N:1 relation to their respective elementary fact types.

The relations of the update fact type and of the elementary fact type to surrounding fact types are identical, which is indicated by the same colors in green or purple respectively.

If instantiations of the update fact type allow for inference of new facts, the types of those facts have to be connected to the update fact type by arrows with solid arrowheads. This is depicted by the representative fact type "new surrounding fact types", which might evolve with the modelling of an update fact type.

Summarizing, the modelling steps are as follows:

1. Identify variable fact types.
2. Group together fact types, which surround the variable fact type in the elementary fact type view, according to identical real-world triggers.
3. Introduce the corresponding update fact types, which are pointing onto the fact types under consideration with an arrow with solid arrowhead.
4. Connect tracelet fact types and trace fact types to surrounding fact types like the corresponding elementary variable fact type.
5. Evaluate, whether the update fact type functionally determines new fact types. If so, include the new fact types and the corresponding functional dependencies.

Tracelets and Traces

Variable fact types imply the possibility that an attacker can collect and aggregate several facts of the same type in a homogeneous fact set. In reality, there exist fact sets of arbitrary cardinalities. However, for this thesis a classification into two classes is sufficient, cf. section 2.2.2. Fact sets of the first class—tracelets—consist of a rather small number of facts. They are as a whole aggregate not more sensitive with respect to the evaluations in this thesis than a single fact. Fact sets of the second class—traces—are larger and are more sensitive than a single fact of the elementary type.

The additional sensitivity can have several reasons, which are not mutually exclusive:

- The trace aggregates more information than one attacker is allowed to know.
- The trace allows for inference of one or several new fact types.
- The trace becomes unique for a user or for a VID. Then it allows for linking of fact sets containing equal instantiations of the considered trace type. This is the case, if the long traces are unique for a user or a VID respectively, whereas shorter tracelets may still map ambiguously to users or VIDs respectively.

  Traces are unique for a user, if the real world events triggering the change of the corresponding system attribute originate from user behavior. The behavior is unique for each user. Then, the trace type functionally determines the user. Those traces can serve for linking different fact sets about the same user irrespectively of the VID under which the traces are revealed.

  Traces are unique for a VID, if the real world events triggering the change of the corresponding system attribute originate from behavior that can be different for each VID. Then, the trace fact type functionally determines the VID, but not the user. Traces of those fact types cannot be used for linking fact sets of different VIDs.

It is not possible to define the exact boundary, when a homogeneous fact set is that large, that it is no longer a tracelet but a trace, for every possible elementary fact type. If such a definition is needed, this has to be evaluated for each variable fact type separately. Often, the critical cardinality can only be defined by an experienced privacy engineer. For this thesis it is sufficient to distinguish, whether the cardinality of a tracelet can be restricted by user configuration or whether the user does not have any influence on its cardinality. In the first case, the user of the system is assumed to configure the system in a way that only tracelets are evolving for the respective elementary fact type. In the latter case, traces also evolve.

Figure 3.3 shows an orange elementary variable fact type and the resulting trace pattern. Again, one fact type at the left hand side represents the constant fact types and independent variable fact types from the surroundings in the elementary fact type view of the model. A fact type on the bottom right represents the dependent variable fact types surrounding the elementary fact type in the elementary fact type view on the model.

Generally, a tracelet fact type and a trace fact type exist for each elementary variable fact type. However, it is possible that already a minimal collection of two facts is more sensitive than a single fact of a given variable fact type. Then, there is not any tracelet fact type for this variable fact type, but only a trace fact type. On the other hand, it is possible that even an infinitely large collection of facts does not bear additional vulnerabilities as compared to

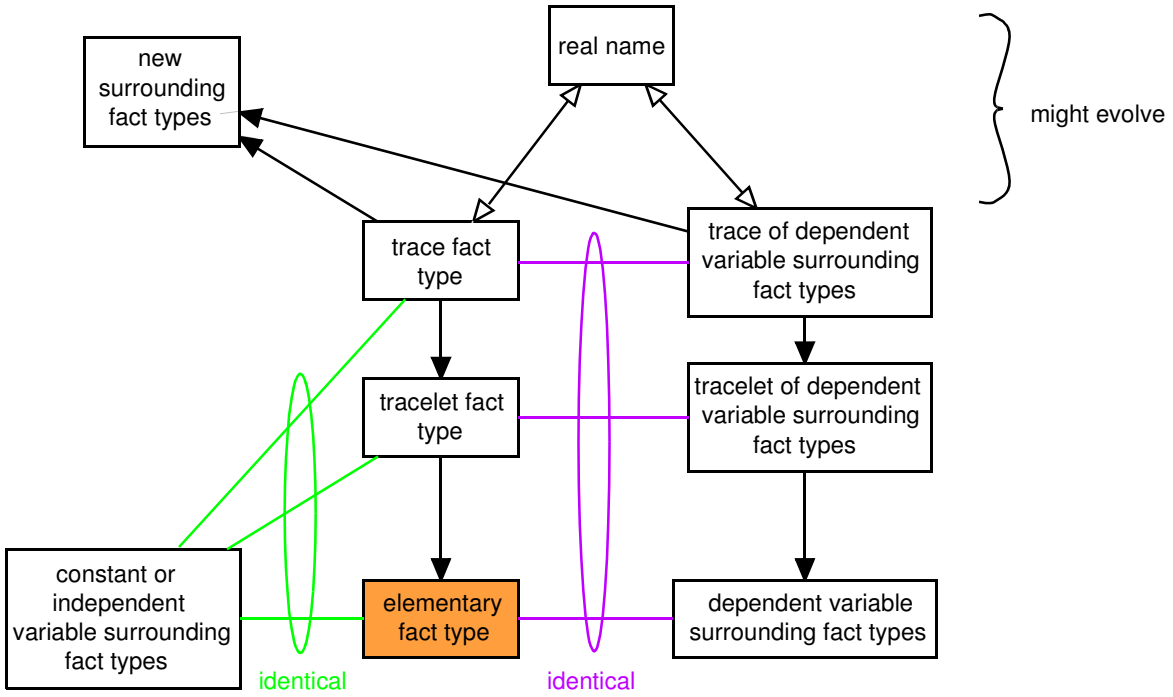


Figure 3.3: Pattern of tracelets and traces

a single fact of the given variable fact type. Then, no trace fact type exists for that elementary fact type.

A tracelet contains single facts and a trace contains tracelets. Therefore, the tracelet fact type functionally determines the elementary fact type and the trace fact type functionally determines the tracelet fact type, respectively. In the other direction, there are no functional dependencies, because a single fact can be contained in different tracelets and a given tracelet can be contained in different traces. Therefore, there are N:1 relations from the trace fact type to the tracelet fact type and from the tracelet fact type to the elementary fact type, respectively.

At the top of the figure, there the two fact types—"real name" and "new surrounding fact types". Those two fact types correspond to the latter two possibilities, why a trace can be more sensitive than a single contained fact. Either one, none, or both of those possibilities might be true for the considered elementary fact type. Therefore, the two fact types at the top of the figure are marked with "might evolve"

A trace of facts of a variable fact type functionally determines the real name fact type, if the variable fact type represents a system attribute, which is changed due to user behavior, e.g., due to a user's location change. Because user behavior is unique, there is no other user—represented by the real name fact type—to which this trace can map. Usually, this function's mapping will not be known. Therefore, the arrowhead is empty.

The fact type "new surrounding fact types" represents the fact types, whose facts can be inferred from the additional information, which the aggregate of facts in the trace contain. An example is that a large trace of locations will allow for inference of some of a user's activities. Such fact types will only be part of the model, if the attackers can infer the corresponding facts, i.e., the mapping of the function is known. Thus, the arrowhead is solid.

This view on the model containing update fact types, tracelet fact types, and trace fact types is called dynamic view—because the fact types have their origin in the dynamics of system attribute values—as opposed to the former view called elementary fact type view. There is no actual need for this separation of views and no clear guideline about which fact types to include in which view. Logically, it still is one single model. However, the separation of the two views proved feasible for readability purposes. If clarity allows for it, the evaluator can integrate both views into one figure.

The steps to achieve the dynamic view are as follows:

1. Identify variable fact types.
2. Group together fact types, which surround the variable fact type in the elementary fact type view, according to identical real-world triggers.
3. If existent, introduce tracelet fact types and trace fact types with corresponding N:1 relations from the respective fact types of the larger fact sets to the fact types of the smaller fact sets. This is to be done for the elementary fact type as well as for the dependent surrounding variable fact types if they are not yet represented in the model.
4. Connect tracelet fact types and trace fact types to surrounding fact types like the corresponding elementary variable fact type.
5. Evaluate, whether the trace fact type functionally determines new fact types. If so, include the new fact types and the corresponding functional dependencies.
6. Introduce the relation to the real name fact type if the trace fact type is uniquely mapped onto the user.

The dynamic view contains only fact types and relations, which model additional information as compared to the elementary fact type view. The dynamic view does not have to repeat parts of the elementary fact type view, unnecessarily. This means, that the evaluator can choose to omit relations to the surroundings, because update fact types, tracelet fact types, and trace fact types are connected to surrounding fact types like the elementary variable fact type itself. If attackers know a trace, a tracelet, or an update, they definitely know a single fact of the respective elementary fact type, too. Thus, the elementary fact type view already reveals vulnerabilities originating by those relations to the fact types of the surroundings.

In practice, this often means, that update fact types can be neglected completely, because they often do not contain more information relevant for the evaluation of the VID protection capabilities than the respective elementary fact type does.

The next section describes the dynamic view on the model of Mobile IPv6.

3.2.2.6 Dynamic View of Mobile IPv6

Figure 3.4 shows the dynamic view on Mobile IPv6. The grey areas are equivalent in their meaning to the grey areas of Figure 3.1. Constant surrounding fact types are of yellow color. They will be evaluated based on the elementary fact type view and are not relevant in the dynamic view. The same holds true for the elementary variable fact types in dark green. Tracelet fact types are light green. Trace fact types are of red color. So are the new fact types being inferred from traces.

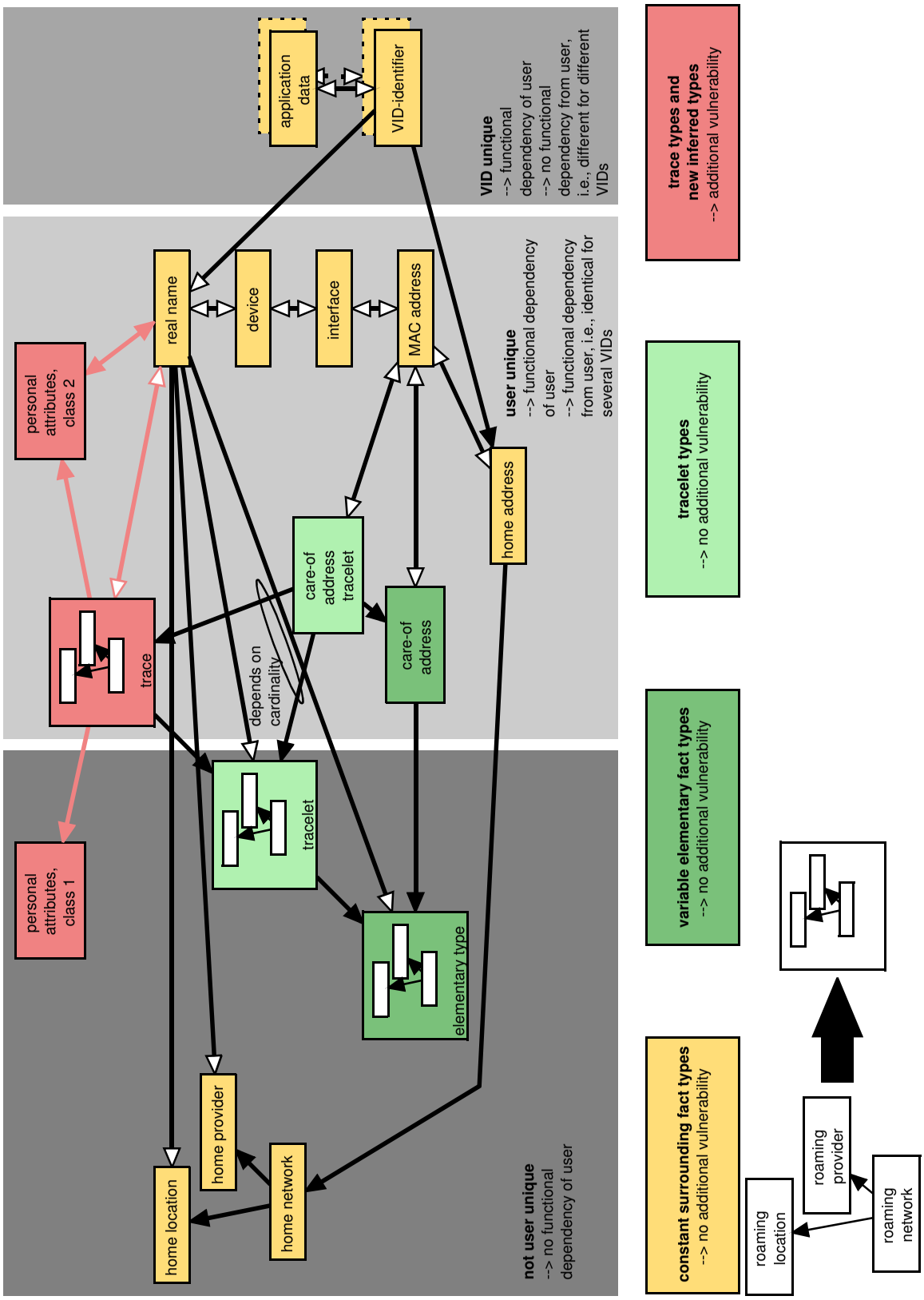


Figure 3.4: Dynamic view on model of Mobile IPv6

From the fact types of the elementary fact type view, the care-of address, the roaming network as well as its location and its provider are variable fact types. The roaming network,

the roaming location, and the roaming provider are together abstracted into one tracelet fact type. This is possible, because the system attributes being modelled by those fact types are changing at the same time and the corresponding facts are building similar tracelets. The corresponding updates are all triggered by the same real world event, i.e., a position change of the user. Movement is an aspect of user behavior and thus is identical for all VIDs.

To each variable fact type there are corresponding tracelet fact types and potentially trace fact types. The relations between the tracelet fact types, and between the trace fact types respectively, are inherited from the elementary variable fact types. All traces originate from user behavior and thus, they all map uniquely onto the real name fact type. No attacker can compute these 1:1 relations in any direction. Therefore the arrowheads are empty.

There are no trace fact types of care-of addresses, because traces of care-of addresses would not contain more vulnerabilities than the elementary fact type does. Thus, it depends on the cardinality of the care-of address tracelet, whether it maps to a trace or to a tracelet of the aggregated fact type around the roaming location fact type. The care-of address tracelet fact type will map to the aggregated trace fact type if additional vulnerabilities are introduced as compared to a single fact of the aggregated fact type. Otherwise, the care-of address tracelet fact type will map to the aggregated tracelet fact type. The trace fact type of the aggregated fact type is unique for a user and thus located in the light grey area, whereas the tracelet fact type is only in the dark grey area, because several users could be the source of tracelets of this type.

It is assumed that from a large location trace additional inferences are possible. Those inferred fact types can be classified in two classes, each represented by one fact type. Facts of the first class are sensitive but do not allow for inferring the real name. Facts of the second class do allow for inference of the real name like, e.g., the postal address does. Although these inference possibilities are not 100% certain to exist, their probability is rated high. Therefore, the model contains them. It is assumed that attackers can indeed infer the user's name from the second class of sensitive attributes and the other way round. Therefore, both arrowheads are solid.

Update fact types are neglected. They do not contain more relevant information than the corresponding elementary fact types do.

With the dynamic view, the model is complete. The next section describes how to exploit the model for evaluation of threats to the VID approach.

3.3 Evaluation

At first some preparations are necessary, in order to lay the base for the evaluation. The methodology for doing these preparations as well as the actual preparations for the evaluation of Mobile IPv6 is described in 3.3.1. After that, 3.3.2 shows the actual evaluation. Therein, the methodology is described and simultaneously applied to Mobile IPv6. Finally, 3.4 summarizes the evaluation results.

3.3.1 Preparations

At first, the potential attackers, which are to be considered in the evaluation have to be defined. For the evaluation of Mobile IPv6, this is done in 3.3.1.1. Then, the evaluator has to collect the possible observations, the inference possibilities, and the possibilities for linking fact sets. This allows for a more fluent evaluation afterwards. The methodology for getting those collections, as well as the actual collections for Mobile IPv6 are described in 3.3.1.2 for the observations, in 3.3.1.3 for inferences, and in 3.3.1.4 for linking fact sets.

3.3.1.1 Potential Attackers

This section instantiates the general attacker model from section 3.2.2.3 for the concrete evaluation of Mobile IPv6. Here, two classes of attackers are considered. The first ones are the communication partners of the user and the providers of the servers of the communication system. They are necessary for the communication system to work. For readability purposes, it is not always written that the providers of servers are potential attackers, but it is only written about servers as potential attackers in the following. The second class consists of eavesdroppers, who can listen on certain communication links. They are not needed for the intended communication. This classification helps the evaluator in finding all relevant attackers. Moreover, it is likely that the evaluator will rate the probability of the entities being indeed malicious by a different probability.

In Mobile IPv6 potential attackers of class 1 are:

- Correspondent Node
- Home Agent

Potential attackers of class 2 are:

- $\text{Eavesdropper}_{\text{CNHA}}$: An eavesdropper on any link between the Correspondent Node and the Home Agent.
- $\text{Eavesdropper}_{\text{HAMN}}$: An eavesdropper on any link between the Home Agent and the Mobile Node.
- $\text{Eavesdropper}_{\text{LinkMN}}$: An eavesdropper being able to see all information on the link of the Mobile Node. This eavesdropper requires deeper evaluation. It may see user-related information of the lower layers as well.

The evaluation starts by organizing the relevant information. Therefore, the evaluator extracts relevant knowledge from the model into several tables. The first table contains observations, which potential attackers can make.

3.3.1.2 Observations

Observations of information by an attacker are the source of all vulnerabilities. Potential attackers can observe all facts that are visible in their domain, e.g., stored on their machine or communicated in their network.

Table 3.1 shows the observation possibilities of the potential attackers in Mobile IPv6. The Correspondent Node can observe the care-of address, because route optimization is assumed to be used.

Attacker	Observation
Correspondent Node	VID-identifier home address care-of address
Home Agent	home address care-of address
$\text{Eavesdropper}_{\text{CNHA}}$	home address
$\text{Eavesdropper}_{\text{HAMN}}$	home address care-of address
$\text{Eavesdropper}_{\text{LinkMN}}$	home address care-of address MAC address

Table 3.1: Observation possibilities

The next section summarizes, which fact types are functionally determined by the observations. Such functional dependencies allow the attacker to infer more information, than is directly revealed for the operation of the system. This is a vulnerability. The section starts by describing, how to extract the possible inference vulnerabilities from the model.

3.3.1.3 Inference of New Facts

A certain condition must hold that an attacker can infer a new fact type. This condition consists of three subconditions, which must all hold true for the inference vulnerability to exist.

From a known fact of a given type A, a new sensitive fact of type B can be inferred, if

1) type A functionally determines type B

AND 2) facts of type B are sensitive

AND 3) the attacker knows the mapping of this functional dependency.

The first subcondition defines, that each fact of type A only has one single corresponding fact of type B. The second subcondition should always be true, because only fact types of sensitive facts are modelled. Nevertheless, this subcondition is relevant for the improvement methodology of chapter 4. The third subcondition is necessary for the attacker to actually be able to compute this new fact.

Table 3.2 shows the observations and the possible inferences according to the model. The solid arrows in the model are referring to functional dependencies. Thus, the table contains all fact types connected by a solid arrow.

Inferences are only one possible attack on VIDs. The other relevant attack is linking of fact sets about a user. This is described in the next section.

3.3.1.4 Linking of Fact Sets

Linking fact sets of the same VID allows for building a trace or a tracelet of facts or of heterogeneous fact sets of one VID. It allows for extending knowledge about a single VID.

Known Fact Type	Inferred Fact Types
VID-identifier	home address
home address	home network
home network	home location home provider
care-of address[a]	roaming network
roaming network[a]	roaming location roaming provider
care-of address trace[b]	care-of address tracelet
care-of address tracelet[b]	care-of address
roaming location trace	personal attributes, both classes
personal attributes, class 2	real name
real name	personal attributes, class 2

Table 3.2: Possibilities for inference of new facts

[a] similarly for tracelets and traces
[b] similarly for roaming network and roaming provider

Fact sets of one VID are naturally also of one user. Thus, the vulnerabilities about linking fact sets of one VID are a superset of vulnerabilities about linking fact sets of different VIDs of one user. Vulnerabilities about linking fact sets of different VIDs are more dangerous, because they allow for merging facts, which the user deliberately wanted to keep separated. Generally, attackers are interested in both vulnerabilities.

A condition can be defined as a means for finding linking vulnerabilities in the model. The condition consists of three subconditions, which must all hold true for the linking vulnerability to exist. There is a second condition, which must only hold true for linking of fact sets from different VIDs.

1. Two fact sets may be linked to the same user and may be merged, if

   1.1) there may exist two instantiations of an identical fact type, one in each fact set

   AND 1.2) those two instantiations are indeed the same fact set, i.e., their values and their timestamps in case of a variable fact type are identical

   AND 1.3) the fact type functionally determines the real name fact type.

2. Attackers can use the two fact sets to merge fact sets across several VIDs, if

   2.1 condition 1 is true

   AND 2.2) the real name fact type functionally determines the fact type of the identical fact sets, i.e., the fact sets are identical in all VIDs.

The conditions are formulated in the general form of fact sets, but can also be applied to single facts, which are fact sets with cardinality one. The mechanism underlying these conditions is, that identity of two facts of a given fact type implies identity of facts of a functional dependent fact type.

The evaluator can easily derive fact types from the model, which are usable for linking fact sets. From those fact types, the real name fact type must be reachable by arrows following their directions. From a process perspective, it is easiest to start by the real name fact type as root and to follow arrows in the model reversely. Attackers can use all fact types that are reachable in the model through this method.

Table 3.3 is summarizing the linking candidates for Mobile IPv6. The fact types are called candidates, because they fulfill subcondition 1.3 and partly subcondition 2.2, which is necessary but not sufficient for being usable for linking fact sets to the same user. There is no statement about whether the facts of the candidates also fulfill subconditions 1.1 and 1.2 here. The link candidate types are marked as being unique for the user, i.e., the user functionally determines them, or only unique for a VID. In the latter case, the second condition does not hold.

Link Candidate Types	User Unique	VID Unique
VID-identifier		X
home address	X	
care-of address[a]	X	
device	X	
interface	X	
roaming network trace	X	
roaming provider trace	X	
location trace	X	
personal attributes, class 2	X	

Table 3.3: Link candidate types

[a] similarly for tracelets and traces

The tables constructed in this section organize the knowledge contained in the model in a way that helps the evaluator in the actual evaluation of the threats. The next section explains this step.

3.3.2 Evaluation

The evaluator has to analyze the system regarding each potential attacker as well as groups of collaborating attackers that are to be considered. Such attacker groups can be of different constitution. There can be groups containing only similar attackers, e.g., only Correspondent Nodes. There can also be heterogeneous groups containing attackers of different nature, e.g., a Correspondent Node and a Home Agent. The following section discusses homogenous attacker groups together with single attackers. The methodology for both is strongly related.

Section 3.3.2.1 explains the methodology for the evaluation regarding single attackers as well as regarding homogeneous attacker groups and evaluates Mobile IPv6 regarding those attackers. Section 3.3.2.2 then does the same for heterogeneous attacker groups.

3.3.2.1 Single Attackers and Homogeneous Attacker Groups

The evaluator has to consider every potential attacker by doing the following steps. At first, the evaluator creates a table showing the observations, which each attacker can make and the possible inferences from those observations. The evaluator marks, whether the attacker can build a tracelet or a trace of a certain elementary fact type. This table then contains all fact types the attacker can get. The table is complemented by a row indicating whether the fact types allow for linking fact sets of one user or of one VID. All this information is already present in the tables of 3.3.1 and can simply be constructed from there. The model itself is not needed in this step.

After building the table, the evaluator examines the sensitive facts and fact sets that have to be protected. If the attacker can gain one of them, the evaluator has found a threat. Then, the risk resulting from this threat must be discussed. It might be that it is an acceptable risk or that the probability for disclosure or the implied danger is high and therefore it is a non-negligible risk. For this discussion, the expertise of the evaluator is required.

The fact types an attacker can see might be partitioned. If this is the case, it must be mentioned in the discussion. Partitions occur, if an attacker has several observation possibilities, e.g., sees a fact set in outgoing packets, and another fact set in incoming packets but never both fact sets together. Then, the attacker knows in fact several fact sets, but cannot merge those sets if no linking vulnerability exists.

Finally, the evaluator discusses what happens when several of such attackers are cooperating. This is a discussion of the fact sets, which can be linked in order to collect more information about the user.

In Mobile IPv6, the Correspondent Node, the Home Agent, the $\text{Eavesdropper}_{\text{HAMN}}$ as well as the $\text{Eavesdropper}_{\text{LinkMN}}$ all know virtually the same fact types. To avoid repetitions, the next section handles them all together. This is possible, because the attacker model defines that all attackers know the same relations and the same mappings of relations. The only differences are that the Correspondent Node can observe the VID-identifier and that the $\text{Eavesdropper}_{\text{LinkMN}}$ additionally observes the MAC address.

The MAC address is linked to the home address and to the care-of address by 1:1 relations and does not introduce any further relation—neither a relation to a new fact type nor a new known mapping. Therefore, it does not imply any additional threat. The additional VID-identifier at the Correspondent Node instead makes a difference that will be considered.

Correspondent Node, Home Agent, $\text{Eavesdropper}_{\text{HAMN}}$, $\text{Eavesdropper}_{\text{LinkMN}}$

Table 3.4 shows the fact types being known by the Correspondent Node, the Home Agent, the $\text{Eavesdropper}_{\text{HAMN}}$, or the $\text{Eavesdropper}_{\text{LinkMN}}$. Whenever a trace can be aggregated, this is also true for a tracelet. Personal attributes can be revealed in both classes. Only class 2 is unique for the user as shown in 3.2.2.6. To get a more concise view, transitive inferences are listed under the root of the inference chain. The inferences from the VID-identifier are not listed. The inferred fact types would be doubled, because they can also be observed directly.

The following list examines the facts and the fact sets that are to be protected according to the protection goals defined in 3.2.2.2.

Observation	Inference	Trace/Tracelet	User Unique or VID Unique
home address		-	user unique
	home network	-	-
	home location	-	-
	home provider	-	-
care-of address		trace, tracelet	user unique
	roaming network	trace, tracelet	-
	roaming location	trace, tracelet	-
	roaming provider	trace, tracelet	-
	personal attributes		user unique[a]
	real name		user unique
VID-identifier[b]	-	-	VID unique
MAC address	-	-	user unique

Table 3.4: Threats regarding Correspondent Node et al.

[a] If only personal attributes of class 1 can be inferred, this is not user unique.
[b] Only the correspondent node can observe the VID-identifier.

- Fact F1 Home provider

  The home provider is disclosed to all attackers, because the attacker can observe the home address, which allows for inference of the home provider.

- Fact F2 Home location

  The home location is disclosed to all attackers, because the attacker can observe the home address, which allows for inference of the home location.

- Set S1 Large location trace

  A large location trace is revealed. This means that the real name can be inferred. It must be recalled, that this is not a 100% certain conclusion. The functional dependency from the location trace fact type to the real name fact type was included, because it is assumed to be very likely. If an attacker can observe a user's location over a long time, it is nearly sure that some of the locations will disclose the user's identity, e.g., when returning each night to the same postal address. Knowing a user's identity allows for a great number of threats. Thus, this threat is not acceptable.
  Moreover, Table 3.4 shows that personal attributes are revealed. The reason is the large location trace. Thus, those personal attributes can be very sensitive, e.g., visits at the doctor's. This threat is not acceptable.

- Set S2 Location and VID-identifier

  This fact set is only known by the Correspondent Node. The Correspondent Node is the only attacker knowing, which VID it wants to address. This implies a non-acceptable threat.

  Against the other attackers, this set is protected.

- Set S3 n VID-identifiers

  Only the Correspondent Node can see any VID-identifier. If the user is using several VIDs with the same Correspondent Node, these VIDs can be linked by revealed user unique facts, e.g., by the identical home address. This directly undermines the privacy approach of separating the disclosure of facts into several subsets. Thus, it is not acceptable.

  Against the other attackers, this set is protected.

Home Agents, $Eavesdroppers_{HAMN}$, and $Eavesdroppers_{LinkMN}$ only know either all instances of a fact type or none at all. Thus, a collaboration among several identical attackers does not bring any benefit.

Several Correspondent Nodes instead, can observe several VID-identifiers and link them by one of the user unique linking candidates, e.g., by the common home address. Thus, a group of Correspondent Nodes increases the threat regarding Set S3.

$Eavesdropper_{CNHA}$

Table 3.5 shows the fact types being known by the $Eavesdropper_{CNHA}$. For this evaluation, it is assumed that the eavesdropper is not directly located at the Home Agent. There, it would also observe the care-of address in packets destined to the Mobile Node. This would be an example of a partitioned knowledge.

Observation	Inference	Trace/Tracelet	User Unique or VID Unique
home address		-	user unique
	home network	-	
	home location	-	
	home provider	-	

Table 3.5: Threats regarding $Eavesdropper_{CNHA}$

The following list examines the facts and fact sets that are to be protected according to the protection goals defined in 3.2.2.2.

- Fact F1 Home provider

  The home provider is disclosed, because the attacker can observe the home address, which allows for inference of the home provider.

- Fact F2 Home location

  The home location is disclosed, because the attacker can observe the home address, which allows for inference of the home location.

- Set S1 Large location trace

  This set is protected against the attacker, because not even a single care-of address can be observed. Thus, also the real name and personal attributes are not revealed.

- Set S2 Location and VID-identifier

  This attacker cannot see this fact set, because no VID-identifiers are revealed.

- Set S3 n VID-identifiers

  This set is protected against the attacker, because no VID-identifiers are revealed.

If several such eavesdroppers between the Correspondent Node and the Home Agent cooperated, they still would not see any care-of address or any VID-identifier. Single facts are either known to one attacker—and thus, all groups containing this attacker—or they are not known at all. Thus, there is no change in the protection, when attacker groups are considered.

Summarizing, single attackers can be enough to break every single protection goal. Consideration of homogeneous attacker groups exhibits only one additional threat. Collaborating Correspondent Nodes can disclose more VID-identifiers of a user than one Correspondent Node alone.

3.3.2.2 Heterogeneous Attacker Groups

Here, only additional attack possibilities to 3.3.2.1 must be collected. The evaluator considers only smallest possible attacker groups. Of course, each attacker group containing a smaller attacker group, which can already successfully mount an attack, can mount the same attack, too.

As a first step, the evaluator must develop a diagram showing threats, how attackers can cooperate and merge their knowledge. Knowledge can be merged like fact sets of a single attacker, i.e., by finding simultaneously identical facts of a fact type, which functionally determines the real name fact type.

This diagram shows all potential attackers. For each attacker, there is a list of the fact types this attacker knows. Fact types, which can be used for linking fact sets, are marked. Here, a "#" is used for a user unique fact type and a "*" is used for a VID unique type. If an attacker can see several partitioned fact sets, those sets have to be listed separately.

As a next step, the evaluator connects fact types, which can be used for linking fact sets, being in common for several attackers by lines. The lines show which fact sets the collaborating attackers can combine by the fact types, which can be used for linking fact sets.

Then, the evaluator creates a table for each asset of the protection goals. This table lists, which attacker groups have to collaborate in order to aggregate the respective sensitive fact set or to deduce the sensitive fact.

Finally, the evaluator must discuss the result of those tables. To this aim, the evaluator has to rate the sensitivity of the merged fact sets and the probability that the respective attacker groups indeed are finding each other and are cooperating. This results in the final threat analysis result.

Figure 3.5 shows the diagram of linking possibilities for the attackers. For the matter of simplicity, the linking possibilities by the personal attributes fact type and the real name fact type are not included. From the threat perspective they are equivalent to the threat with links by the care-of address. They allow the same entities to link the same fact sets. The Correspondent Node is separated from the others, because in this evaluation the additional VID-

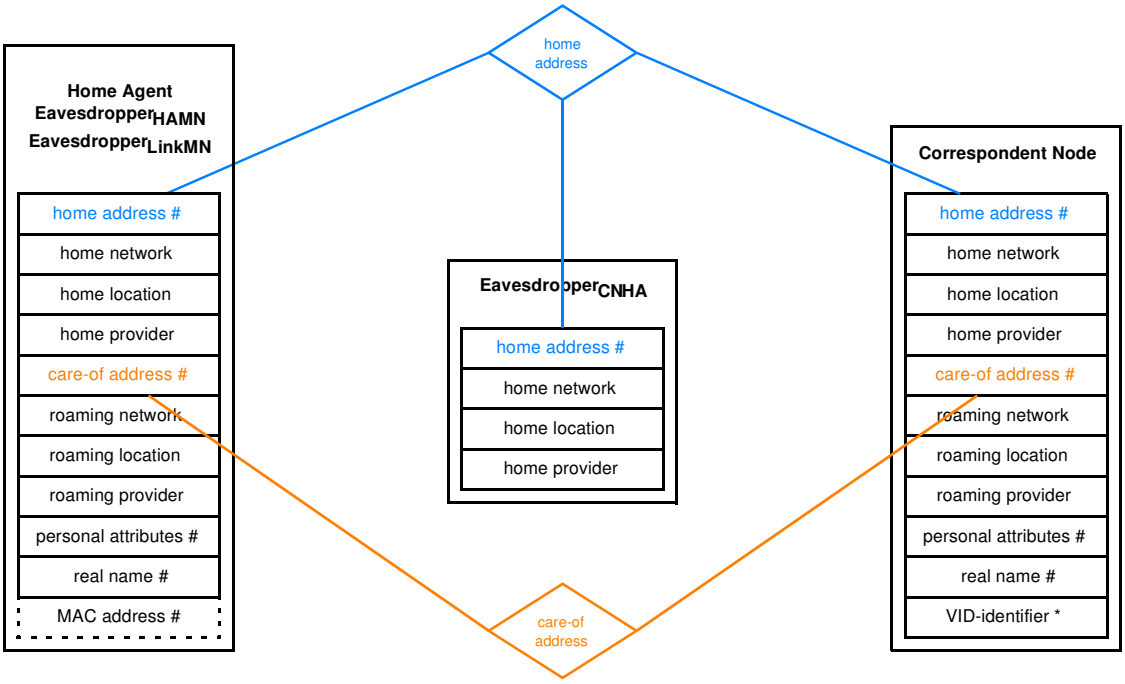


Figure 3.5: Linking diagram for Mobile IPv6

identifier is crucial. The MAC address is in a dashed box, because it can only be seen by the $\text{Eavesdropper}_{\text{LinkMN}}$.

Basically, there are two linking possibilities. Via the identical home address, all attackers can merge their knowledge. Via the care-of address, the personal attributes, or the real name, only the Home Agent, the $\text{Eavesdropper}_{\text{HAMN}}$, the $\text{Eavesdropper}_{\text{LinkMN}}$, and the Correspondent Node can exchange their knowledge.

Evaluation of the protection goals results in the following list:

- Facts F1 and F2

  Single facts are either known to one attacker—and thus to all groups containing this attacker—or they are not known at all. Therefore, there is no change in the protection, when considering attacker groups.

- Set S1 Large location trace

  For a large location trace to be revealed, there cannot be any attacker group without any attacker observing the care-of address. Every attacker knowing the care-of address knows already alone a location trace. Therefore, there are no heterogeneous attacker groups not being a superset of already identified attackers. Thus, there are no new threats. For the sake of shortness, the table detailing the collaboration possibilities is omitted.

- Set S2 Location and VID-identifier

  To evaluate protection of S2, the evaluator searches for attackers, who only know the location as start, and for a path via any linking candidate to another attacker, who knows the VID-identifier. It is also possible, that an attacker with none of both can collaborate with another attacker or a group of attackers who knows both facts. Table 3.6 lists the possibilities.

Attacker 1	Attacker 2	Link By
Home Agent $Eavesdropper_{HAMN}$ $Eavesdropper_{LinkMN}$	Correspondent Node	home address care-of address personal attributes real name
$Eavesdropper_{CNHA}$	Correspondent Node	home address

Table 3.6: Collaboration possibilities for S2

It can be seen that each possibility requires collaboration of the Correspondent Node. The Correspondent Node alone already knows set S2. Thus, those collaboration possibilities are just supersets of already discovered threats, i.e., supersets of the Correspondent Node. Thus, no new threat is found by considering heterogeneous attacker groups.

- Set S3 n VID-identifiers

  There is no attacker besides the Correspondent Node knowing the VID-identifier. Thus, there cannot exist any group knowing several VID-identifiers without collaboration of at least one Correspondent Node. Thus, the heterogeneous groups can only be supergroups of the one already identified in 3.3.2.1. For the sake of shortness, the table detailing the collaboration possibilities is omitted.

The evaluation shows that heterogeneous attacker groups do not bring any additional threats. Only supergroups of already known attackers can reveal the assets to be protected.

The next section gives a concluding summary of the whole evaluation.

3.4 Summary of Evaluation

The evaluation result is alarming. For each protection goal, there is at least one possibility to break the goal. For all assets to be protected except for S2, there are even several kinds of attackers, which can break it. The potential attackers are very diverse ranging from eavesdroppers over the Home Agent provider up to the Correspondent Node as communication partner of the user. It is virtually impossible to keep control over all those potential attackers.

Even if assuming absence of all eavesdroppers, every protection goal still can be broken. Moreover, one could assume that a Correspondent Node is likely to be either a private communication partner or a service provider. In the first case, it might be valid to assume trustworthiness. But even this would only protect several VID-identifiers from being linked and in this case, it would not make sense to use different VIDs towards a trusted communication partner, anyway. In the second case, trust is not an option. The benefit of gained knowledge about the user is very high. Thus, it cannot be concluded that all providers will overcome this temptation. In some cases, the user might have the choice between different equivalent service providers and thus can choose a trusted one. But this cannot be assumed to be always possible.

The threats towards single attackers are already so numerous, that consideration of attacker groups does not bring many more threats. Only the threat of aggregating several VID-identifiers is aggravated.

If the application layer handles that sensitive data, so that the approach of VIDs is to be used—which can be assumed for future applications—an improved communication system is needed. This must be tailored for the use with VIDs.

There are signs that improvement is possible. For communication, it is, e.g., not necessary that the Correspondent Node sees the user's location—an arbitrary identifier would be enough. Moreover, this identifier does not have to point to sensitive information like the home address does. Considering the Home Agent, there is no need for it to see the user's location in terms of the care-of address all the time, but only when indeed packets for the user arrive.

The next section gives a discussion of the presented methodology in the light of related work.

3.5 Related Work

Related work for this chapter can be divided into five sections. Section 3.5.1 starts by discussing related work in the knowledge representation area. Section 3.5.2 gives a short introduction to data mining, followed by a discussion of link detection 3.5.3 and inference 3.5.4 in databases. Finally, section 3.5.5 discusses other methodologies for security and privacy evaluation.

3.5.1 Knowledge Representation

The knowledge representation in this thesis is based on the semantic relationship graphs of [109], which discusses inference aggregation detection in databases in general and which itself bases on, e.g., [159], [158], which both are focused on inference detection in multilevel secure databases. Linking of VIDs is not considered in this work at all.

[159] establishes an entity based schema, where the relationships can be evaluated without considering how they might be stored in a database. It defines a core set of data, from which a sphere of influence can be derived by inferences.

[109] names a number of important properties of the semantic relationship graph. First, it is claimed to be well understandable for humans and thus allows for human peer review. Second, the graph is extensible when new relationships are recognized. A model of additional applications or additional parts of a system can be integrated by joining identical concepts, i.e., nodes that are semantically equivalent. Third, the graph is independent from concrete database realization techniques. All these aspects are important for this thesis.

There are a number of differences as compared to this thesis. First, the application domain is different. The work described above focuses on data to be stored in a database and searches for flaws in the schema of the database. This thesis instead focuses on the evaluation of a communication system and thus searches for flaws in the system design. This results mainly in an increased intellectual effort to derive the entities of the model.

Second, the mentioned work distinguishes between classified—i.e., sensitive—and unclassified—i.e., non-sensitive—relationships. This means, that the relationships exist anyway, but they may be concealed towards requestors of the database. In this thesis, relationships are existing or not existing. If they are to be concealed, they must be erased.

Third, the graphical representation in this thesis slightly differs in that there is only one relationship between two entities, whereas [109] distinguishes between both directions. Moreover, the cardinality is represented differently.

Fourth, this thesis only considers certain relationships, whereas the mentioned related work also considers uncertain relationships and thus provides for a notation for the certainty.

Finally, the representation in [109], [159], [158] does not express, whether an attacker knows the mapping of an inference or not. It assumes, that every existing inference possibility can also be computed.

[238] and [236] evaluate attacks to anonymity and location privacy in mobile communications. The representation in this work is focused on value-level. It only contains four basic data types, the location, the user, the device, and an action being caused by the user's device. The entities are connected by a full mesh of arrows, whereas both directions are represented unidirectionally. It additionally provides for a notation of imprecise knowledge.

The arrows there also mean relations between the sets of all possible entities being connected by the arrow like in this thesis. It is intended to denote an attacker model in showing that a potential attacker might know the relation between two instantiations of the connected entities. If an attacker knows the relation, it is assumed that it also knows its mapping. No distinction between known and unknown mappings is possible like it is in this thesis. For instance, the existence of the relation between the entities user and device means that an attacker might know which user might use which device, i.e., could infer the device from the user. The opposite relation means that an attacker might know which device is used by which user, i.e., might infer the user from the device. Because the system works on value-level, it may still be possible that the relation comprises only the empty set. In that case the system protects the sensitive information although the relation is contained in the model.

The main differences of [238] and [236] to this thesis are that they work on the value-level instead of the type-level like this thesis does. Moreover, [238] and [236] only consider four types of data and do not consider virtual identities.

3.5.2 Data Mining

Data mining is a technique for gaining new knowledge from large amounts of data. For data mining to work, the data has to be cleaned and has to be in a good state. Therefore, sometimes a data warehouse is named as a prerequisite. According to [131], which is a textbook about database design, data mining looks for previously unknown coherences in the data. It works on a value-basis.

[131] names three aspects of data mining. The first one is the classification of objects. The goal is to predict future values of objects. To this aim, the existing base of objects is classified according to one or more of their attributes. It is possible to further refine the classification in subclasses by so-called decision trees. The classification needs a representative

amount of data to be mined in order to achieve good results. The attributes for the classification can be given by the administrator or can be automatically derived by the data mining tool itself.

The second aspect is association. This aims at finding correlations between objects by deducing rules between the objects. An example for a rule is that if any person buys a computer, he or she will also buy a mouse for it.

The third aspect is called clustering and aims at grouping objects according to previously undefined attributes. An example is to group all objects about workers with an income of more than 40.000 EUR per year. This groups together logically related objects and allows also for detection of so-called outliers, which do not fit into the detected scheme.

[219], a primer about data mining, names three more aspects of data mining. Prediction aims at forecasting trends for the future, e.g., about the estimated income of engineers in the year 2020. Estimation aims at analyzing trends for deduction of other characteristics, e.g., for deduction of the number of children from the spent money. Deviation analysis finally compares the values of objects to given norms in order to detect anomalies.

Data mining usually uses machine learning techniques like neural networks, inductive logic, or k-nearest-neighbor techniques. [219] classifies the data mining methods into top-down methods, where a hypothesis stands at the beginning, which is validated in the following. In contrast, bottom-up approaches start with examples and deduce the hypothesis.

Because data mining is inherently aimed at finding new knowledge in stored data, it often violates privacy. There is a growing research community trying to control privacy even in data mining environments, e.g., [225], [181], or [4].

The work on the value-level and the exploring of previously unknown coherences in the data are differences to this thesis working on type-basis and considering only well-known coherences between fact types. Nevertheless, data mining is often used in a more general way, just saying to examine the stored data more closely. Then, this thesis can be seen as data mining work.

While the clustering of data mining resembles the step of an attacker to group together facts about one user in this thesis, the attacker here can only use known attributes and group together facts according to identical values of those attributes. This is a difference to clustering in the data mining context, which works on previously undefined attributes according to [219].

3.5.3 Link Detection in Databases

With the growing amounts of data stored in databases, the importance of processing it and keeping it in a clean state increases. Therefore, it is crucial to identify duplicate entries that in fact are referring to the same real world entity. Details regarding state of the art and current research directions can be found, e.g., in the surveys in [66], [228], or [92]. Here, a brief introduction is given.

According to [147], a framework for entity resolution modelling, and according to [66], a survey on duplicate record detection, there are similar link detection problems in different application domains without a generic unified solution and without a common terminology, so far. Therefore, [147] provides a framework for modelling those related problems and

[19] formulates a generic approach to entity resolution, which is the umbrella term according to those works. Basically, [147] separates two concepts relevant to this thesis—record linkage and deduplication.

The origin of record linkage lies in the merge of different databases, where there are subtle differences in the entries belonging to the same entity, e.g., a typo in the spelling of the name. In this thesis, only one database with correct entries is assumed, i.e., the attacker's knowledge.

The difference to record linkage is, that in deduplication problems, duplicate records of the same entity are searched in a single database [147]. This is the concept underlying this thesis.

In both occurrences, the underlying problem is having several instantiations of a database entry—which can be a set of values in fact—stemming from the same underlying concept. According to [146] focussing on how to identify matching data trails and [147] it is assumed, that there exist functions, which map the considered entries to the underlying concept. The result of these functions with the entries under consideration as input must result in the same real world entity. Then, both entries can be linked. Thus, a common focus of related research is on searching these mapping functions.

[153], which evaluates privacy effects of authentication and [226], which focuses on matching criminals' identities, consider the record linkage problem for multiple identities of a user. In [226], the focus is on linking the record based on application knowledge without a known and common data structure behind all entries. [153] reviews the trade-off between multiple identities and authentication. Privacy protection regarding entity resolution often is achieved by multi-party operation where several parties have to cooperate in order to fulfill a certain functionality and no party alone knows the link, e.g., [146], [40], which discusses privacy preserving data mining.

The difference to the linking problem evaluated in this thesis is that here only different instantiations of the same fully specified entity-relationship model are evaluated. In contrast, most of the related literature does not consider the merge of two databases with the same underlying schema but the merge of similar databases with similar but different schemas.

A related problem is identification. In identification problems, it is tried to identify the user to which a database entry belongs. This is especially relevant in cases, where the database entries are intended to be anonymized, e.g., in genome databases or weblogs [146]. In this thesis, it is only necessary to realize, that different VIDs are from the same user, irrespective of the real identity.

3.5.4 Inference in Databases

Security and privacy in databases is a prominent topic, cf. e.g., [205] and [206] modelling semantics relevant for security, and [220] giving an overview on the status of R&D in this field. The relevant aspect for this thesis is the so-called inference threat according to [72], [73] containing surveys and [11], a technical report of the National Computer Security Center. An inference threat exists when an attacker can infer sensitive information from non-sensitive data and metadata [73]. Metadata are, e.g., database dependencies and integrity constraints or outside information.

In the context of related research this means that an attacker is allowed to query a database for certain information, which is non-sensitive regarding this attacker. From this information, the attacker can infer information, which should remain concealed and which must not be queried directly. Often, an inference problem exists if there are multiple paths from a non-sensitive information to a sensitive information and the direct path is concealed against the attacker, but indirect paths—via different queries or other queried information—are not protected according to [182], focusing on detection and elimination of inference channels.

Inference problems are often evaluated together with aggregation problems, e.g., [109]. An aggregation problem exists if a collection—an aggregate—of several pieces of information together reveals new knowledge, which should not be revealed. This can be compared to the definition of a trace of facts in this thesis. Sometimes, the boundaries between inference and aggregation problems are blurred like [144] discusses.

[73] gives a general overview on the database inference research topics and subdivides research according to the database nature—statistical databases, multilevel secure databases, or general purpose databases. [124] gives an overview on multilevel secure databases, which is closest to the problem in this thesis.

Proposed solutions can also be classified, whether they aim at detecting inference during database design or during query time. The first kind of solution relates to the type-based evaluations in this thesis and works purely on the intension of the database, whereas the latter has to consider the single values stored in the database, i.e., its extension. [231] describing inference on the value-level claims that more inferences can be detected when considering the values. Work on type-level instead often results in more restrictions than would minimally be possible [73].

There are several possibilities for inferring additional information from known ones. For this thesis, only inference by functional dependencies is relevant. [214] states that for such an inference threat, a functional dependency must exist and the attacker must know the mapping of the function. Nevertheless, this work considers the mapping being always known, which is a difference to this thesis. Other, more complicated algorithms can be used to exploit multilevel dependencies [214] or join dependencies. Inference attacks on the value-level can, e.g., mean to issue several queries to a database and to build an intersection of the responses. For more complicated inference evaluations [52] and [53] introduce a methodology based on conceptual graphs.

[52] structures inference research into four questions, the first three coming from [110] and the last one from [83], both IFIP working group reports. The first question relates to this thesis and aims at discovering fundamental possibilities for undesirable inferences. This is based on the type-level and on integrity constraints, i.e., on the schema of the database. The second question aims at automatically discovering inference rules from fundamental relationships. The third question aims at automating the inference process itself, whereas the last question aims at jamming an attacker's knowledge with false information. The latter three questions are not relevant for this thesis.

More recently, research aims at providing assurance of the detected inferences [73]. This comprises, e.g., techniques to handle imprecise inference [96], [97], [158], [159] and a formal evaluation of the correctness of the detected inferences [72], [26]. Both is out of the scope of this thesis.

3.5.5 Methodology

The privacy approach of using multiple virtual identities is a rather new approach. This might be the reason that there are no comparable methodologies for evaluating threats to the VID approach known to the author. Nevertheless, there exists a variety of security evaluation methodologies and partly also privacy evaluation methodologies. They are surveyed, e.g., in [140] and [135]. They are briefly discussed in the following.

Generally, security and privacy evaluation methodologies can be separated into informal ones and formal or semi-formal ones. The boundary between the latter two is not sharp. Thus, they are discussed together.

[135] gives an overview on informal assurance methods of IT security with a focus on privacy aspects. Therein, the authors distinguish between assurance methods for products, e.g., SmartCards, and assurance methods for security processes or security management usually focused on IT infrastructures. The same distinction is done in [8]. Important for the first class are, e.g., the Common Criteria [42], its predecessors ITSec [55] and TCSec [222], or the privacy seal of the Independent Centre for Privacy Protection [117].

Important representatives of the second class are, e.g., the German IT-Grundschutz [28], the British BS-7799 [25], or the respective ISO/IEC 27000 family [120]. [69] from the European Network and Information Security Agency, ENISA, gives an inventory of the latter methodologies under the umbrella of risk assessment and risk management. On [70] those methodologies can be compared online.

Those informal methodologies are usually based on a structured catalogue of questions. By this, it is assured that an evaluator considers a standardized set of security issues. Often, the methodology to answer the question is left to the evaluator. [43], the Common Evaluation Methodology of the Common Criteria, e.g., describes an overall security evaluation framework. It specifies, that the evaluator has to assess the vulnerabilities of the target of evaluation. The evaluator is free in choosing the methodology during the vulnerability assessment step. The proposed methodology of this thesis is a candidate for using in the Common Evaluation Methodology in order to assess vulnerabilities regarding the VID approach. For some concrete questions, where methodologies are established, the methodology to be applied is specified, e.g., that penetration testing by a certain tool is to be applied in order to set the target of evaluation under well-defined attacks.

Beneath the above mentioned methodologies for overall security of IT infrastructures, there are also focused methodologies for certain problems, e.g., for cryptographic security primitives [179] or for evaluation of an IP based network stack [178].

At the other extreme of the formal-informal scale, there are the mathematically formulated methodologies. There is a multitude of approaches, which are surveyed, e.g., in [152], [170], [90]. [229] contains an overview on formal security methods with a focus on applicability to anonymity systems. Therein, it is stated that even those fully formal methods can never prove total security, because every model is based on a certain abstraction and assumptions, which might be incomplete or even contain errors.

According to this overview, the first approaches, e.g., [30], [88], [216] are based on formal logics—modal logics describing necessity and possibility, epistemic logics describing knowledge, uncertainty, and ignorance, or temporal logic describing time aspects. Later,

more detailed methods are based on process calculi describing communicating processes, e.g., CSP [111], [191], CCS [154] and its follower the pi-calculus [155]. There are also other formal techniques, e.g., [113] is based on function views and formalizes definitions around anonymity, or [29] systematically models security threats using data flow diagrams. [10] develops a tool for automated validation of security protocols using a high-level descriptive language, which is translated into an intermediary language, which then can be translated into low-level model checking languages like [16].

Most close to this thesis are approaches, which are often called semi-formal, but which are also called formal by some authors. Among them, there are graph-based research approaches focussing on privacy aspects. Such approaches are discussed in the following.

[161] and in short [162] discuss a conceptual methodology to evaluate and to design controlled anonymous applications. Controlled anonymity means that the users are not unconditionally anonymous, but that some actions, e.g., malicious ones, can be linked to a user identity, if certain conditions hold true. Therefore, requirements, e.g., anonymity from a user perspective and accountability from a provider perspective must be balanced. For the evaluation of a system regarding those properties, they use a so-called flow graph and petri-nets.

By this methodology, it is possible to see which potential attacker sees which information under which conditions. Thus, it can be derived, whether privacy flaws are present and whether control requirements such as identity escrow possibilities are fulfilled. It is also possible to model several solutions, thus supporting the choice for one of them. While the basic approach of evaluating threats and using this model for improving the system is similar to this thesis, the goals as well as the modelling techniques—flow graphs and petri nets—are different to this thesis. This work is defined on a type-basis and is suitable to support controlled anonymity in systems at design stage.

[236], [238] introduce an attacker model for anonymity systems and location privacy in mobile communications. They show the basic coherences between four data types: Users, their actions, devices, and locations. The attacker model defines, which inferences between the four data types the attacker can do. They provide an algorithm for the attacker to reason on its knowledge for attacking users using anonymizing techniques. The model provides for a formalization of known attacks on anonymizers. The model can be extended to handle imprecise information by the technique of possible worlds, in a way like the technique was adopted by [151] in the area of trust relationships.

As compared with this thesis, [236], [238] have a different goal. They provide for a model and algorithm to reason on an attacker's instantiated knowledge. Therefore, they work on a value-base and aim at evaluation after an observation for attacking an anonymized user. They do not aim at evaluating systems at design stage. They do not consider virtual identities. In contrast to this thesis, they also handle session mobility and user mobility, i.e., users changing their device.

[45], [46] formalize the privacy definitions from the Common Criteria. They provide for a model with nodes and connecting paths to represent properties of a value-base regarding unobservability, unlinkability of actions, anonymity, and pseudonymity. The nodes represent propositions, an attacker can do about a user. The paths represent the probability that the propositions being connected by a path are related. They define weights of paths between nodes of a model for the cases when anonymity etc. is protected. For the actual

evaluation the weights of the value-basis under consideration are compared to the defined absolute weights. Thus, statements can be made, whether anonymity etc. is protected in the value-base.

As compared to this thesis, [45], [46] have different goals. They are mainly focused on questions around anonymity. Linking VIDs and inferring new information is not tackled. They consider imprecise information and work on a value-level. Moreover, the direction of the paths is not distinguished like in the model of this thesis. They work with the concept of a functional mapping from propositions about a user to the user's identity, like this thesis does.

[223] focuses on electronic transactions and evaluates unidentifiability and accountability. It is based on modal logic and uses graphs to represent semantics of anonymous transaction protocols. It also has different goals than this thesis has.

Chapter 4

Improvement Methodology and its Application to Mobile IPv6

For the protection of a user's private sphere, it is important to evaluate the existing systems with respect to their vulnerabilities to the VID approach. Once these vulnerabilities are known, the natural next step is to improve the system. These two steps can also both be carried out during the initial design phase of a system.

Often, privacy engineers develop privacy enhancing improvements solely based on their intuition. This sometimes makes it hard to understand the rationale behind design decisions and yields arbitrary results. Moreover, the privacy engineer must be a full expert having a good enough intuition for this process.

A strongly formalized methodology for improving systems serves for two purposes. First of all, the improvements are well justified and the rationale behind each improvement can easily be understood. Secondly, a methodology allows also for privacy engineers without deep VID knowledge to improve systems, by benefitting from the deep expert know-how, which the designer of the methodology has put into it. The privacy engineer applying the methodology must still be a privacy expert, because the problem of protecting VIDs is too complex to map it to a fully formalized model allowing for automatic execution of the improvement by a non-expert or even a machine. The methodology gives the privacy expert a guideline at hand, how to proceed for improvement. But there are still degrees of freedom, which require a privacy understanding.

The model from the evaluation in chapter 3 describes the system in the relevant aspects regarding the VID approach and exactly defines the vulnerabilities regarding the VID approach. This is the natural basis for an improvement. The model defines the relevant fact types including their relations. Secondly, it exactly details the nature of the vulnerabilities, thus showing where and how to tackle the problems. The methodology presented in this chapter will thus build directly upon the evaluation methodology presented in chapter 3.

Virtually every privacy improvement bears costs at other aspects, e.g., performance or scalability. Thus, improvement here means improvement with respect to the protection of the

VID approach. Other aspects are considered only with second priority, e.g., for the selection of an improvement step if several choices for an improvement are possible. An overall improvement searching the optimum with respect to all aspects would require an overall system metric covering all those aspects, which is not feasible for all thinkable systems.

In section 4.1 the methodology is introduced. Section 4.2 shows the improvement of Mobile IPv6 by applying the methodology. Then, in section 4.3 the new architecture is presented as a result of the improvement process. Finally, section 4.4 shows related methodological approaches and compares the new architecture to other privacy-enhancing communication systems.

4.1 Methodology

The idea of the methodology starts with the evaluation shown in chapter 3, which resulted in a collection of the vulnerabilities and threats. The evaluation process separated the vulnerabilities in vulnerabilities about inferring new facts and in vulnerabilities about merging known fact sets. Both vulnerability classes require certain subconditions, which must hold true in the model for the vulnerability to exist.

Consequently, the vulnerabilities are eliminated, if one of the subconditions does not hold true. Thus, the system improvement methodology aims at making the subconditions of each vulnerability false. In order to assure, that a certain subcondition does not hold true, usually several different architectural building blocks are available. This chapter only names some of them without a claim of completeness. It is up to the privacy engineer's intelligence, to select the best suitable building block. Intended spreading of false information is not considered, because this would negatively affect the correct functionality of the system to a certain degree.

The procedure for system improvement is as follows. At first, the privacy engineer tries to avoid sensitive observations completely. If this is not possible, the engineer partitions the possible observations into smaller, non-sensitive observable fact sets. In some cases, avoidance of observation vulnerabilities is either not possible without affecting the functionality of the system or it is too expensive. Then, the privacy engineer steps through each remaining interpretation vulnerability and chooses one subcondition to make false by an architectural building block.

Because this procedure is not a fully formalized one, it cannot be concluded to a 100% that the resulting system is secure. Therefore, the methodology has to be accompanied by a full evaluation of the new system at the end. This will be done in chapter 5.

After protection against a vulnerability, the system has been changed. Thus, the privacy engineer must do a new evaluation with a new model and iterate through the already passed steps of the improvement methodology. This has to be done until no improvements are feasible any more by the already passed steps of the procedure.

Often, the introduced changes in the system are small and privacy engineers can try to do the required steps in their minds. This approach is also followed here, because a new model and evaluation after each step would not bring enough new insights to justify the decreased readability flow. If the engineer has made a fault, this will be identified by the final evaluation.

The following sections detail the methodology first to avoid observations and then to avoid interpretation possibilities. Interpretation covers both ways, the inference of new facts and the merging of fact sets.

The sections will follow a common structure. At first they are defining the goal being followed by a reasoning of the goal. Then, candidate fact types and candidate relations in the model are identified. They are the model elements, for which a protection being discussed in the respective section makes sense. Finally, each section gives exemplary protection possibilities first by description of the concept and then by a chosen realization of the concept.

4.1.1 Observations

In this section the protection steps dealing with avoidance of observations are detailed. In the next subsection the complete avoidance of observations is described. In the next but one subsection the partitioning of sensitive observations, which cannot be avoided completely is presented.

4.1.1.1 Avoidance of Observations

Goal

The goal is to minimize the disclosed data about the user. Therefore, the privacy engineer avoids disclosure of as many facts as feasible with respect to the implied trade-offs.

Reasoning

Facts, which potential attackers cannot observe, are not known by the attacker. The facts cannot serve as sources for further interpretations.

Candidates

The target for this type of protection are fact types whose facts are directly or indirectly sensitive. Moreover, attackers must be able to observe those facts. According to the definition in chapter 3, facts of directly sensitive fact types are containing sensitive information. Facts of indirectly sensitive fact types can be used by attackers either to infer facts containing sensitive information or to merge fact sets, which then contain sensitive information.

These are the same criteria like when identifying which fact types to include in the model. Therefore, the definition of the candidates here can be shortened. They are the fact types of the model, which potential attackers can observe.

Protection

It is often possible to avoid disclosure of all facts of a certain type. Sample mechanisms are the following:

Concept	Change the system design so that the sensitive information is no longer contained.
Example	Privacy extensions for IPv6 autoconfiguration [163] find another way of assuring uniqueness of automatically created IP addresses without including the unique MAC address. Thus, the MAC address can no longer be used to track users across several IP addresses.
Concept	Use encryption
Example	Conceal fact types of, e.g., signalling data by using IPSec between nodes of the control plane.
Concept	Use a proxy to shield sensitive information from untrusted nodes.
Example	Use a bidirectional tunnel in Mobile IPv6 between the Home Agent and the Mobile Node. Then, the care-of address containing sensitive location information is not disclosed to the Correspondent Node.

Sometimes, it is not possible to conceal all the facts of a certain fact type. Then, the disclosure of the facts should be minimized and not be possible until it is really needed for the operation of the system. Sample mechanisms are the following:

Concept	Store facts at a trusted entity and grant untrusted entities only access to them if they are really needing the facts.
Example	Untrusted location based services are only allowed to fetch a user's location from a trusted location server, if the user is indeed requesting the location based service.
Concept	Split a sensitive fact in several non-sensitive shares and store those shares at different Shareholders. For recombination of the sensitive fact, all Shareholders will have to cooperate. Allow for recombination only when the fact is really needed.
Example	Split the care-of address in several shares and store them at different Shareholders. The shares must only be recombined, when indeed a packet to the Mobile Node arrives and must be delivered to the care-of address.

Some observations cannot be concealed without affecting the functionality of the system. Then, an additional approach is to partition the disclosed fact sets into several smaller sets, which do not contain that much sensitive information like the original fact set did. This is the topic of the next section.

4.1.1.2 Partitioning of Observations

Goal

The goal of this section is the restriction of the maximum fact set a potential attacker can gain about a user.

Reasoning

Often, small amounts of data are less sensitive than large amounts. Thinking about location traces as fact sets, a large location trace bears more sensitive information than a short tracelet. Similarly, a fact set containing location and real name of a user is more sensitive than a fact set containing only the location. It is the task of the privacy engineer to identify, which fact sets are sensitive and which fact sets can be tolerated. This idea is comparable to the idea of VIDs.

Candidates

The candidates here are again based on all fact types being directly or indirectly sensitive, i.e., all fact types of the model. Further, the candidates are restricted to those fact types, whose facts attackers can group with other facts. Such resulting fact sets can be homogeneous if they contain facts of one single fact type or heterogeneous if they contain facts of different fact types. Homogeneous fact sets can only contain facts of variable fact types.

Protection

Protection is based on the distribution of trust. The user's revealed facts are distributed among several not fully trusted entities. Thus, each entity knows only a smaller set of facts and if this entity is malicious, only a part of the user's facts will be disclosed.

Sample mechanisms to partition homogeneous fact sets are the following:

Concept	Change the agent, which is observing the facts of the variable fact type over time.
Example	Change the Home Agent, which is observing the location, over time. Thus, each Home Agent gets only a short tracelet of locations.
Concept	Change the identity of a user over time. Thus, appear as somebody different.
Example	Have several eMail accounts and use them alternatingly with the same service provider. Thus, the service provider will only see small fact sets per eMail address.

A sample mechanism to partition heterogeneous fact sets is the following:

Concept	The sensitive fact set is split into several parts and disclosed to different parties. If the full fact set is necessary for the operation of the system, those parties have to cooperate. In this case they need an anonymous handle to refer to the fact sets of one user. If cooperation between both parties is to be avoided completely, a trusted third party can be placed in between.
Example	Use different providers for different services. Thus, e.g., the navigation service provider knows the location and the shopping mall provider knows the identity, but nobody knows both, location and identity.

4.1.2 Interpretations

This section aims at avoiding the interpretation of the disclosed facts by potential attackers. It considers both relevant interpretation possibilities: Inferring new facts and merging fact sets. This is also the structure of the two following subsections.

4.1.2.1 Avoidance of New Fact Inferences

Goal

The goal is to avoid that attackers can infer new facts, which are directly or indirectly sensitive, from disclosed facts.

Reasoning

If the privacy engineer avoids that disclosed facts allow for the inference of new facts, potential attackers only know, what they really need to know for system operation. Thus, attackers cannot extend the known view on the user by the inference of new facts.

Candidates

The candidates for protection in this section are relations with solid arrowheads, which are pointing from fact types, whose facts an attacker can see, to fact types, whose facts are directly or indirectly sensitive. Moreover, the facts, from which those relations originate, are candidates.

Protection

The protection aims at making false one of the subconditions from section 3.3.1.3 being necessary for an inference vulnerability to exist. Thus, each subcondition is considered separately in the following. From the pure point of view of VID protection, falsifying of the different subconditions is equivalent. Thus, the privacy engineer has usually several possibilities for protection and should choose the best suitable one in the context of the concrete system under construction.

Subcondition 1: Fact type A functionally determines fact type B. In order to assure that subcondition 1 is false, the privacy engineer must make facts of fact type B ambiguous with respect to the mapping from facts of fact type A. Sample mechanisms are the following:

Concept	Reduce the exactness of information so that it corresponds to several users.
Example	Use broadcast addresses, which point to several users.
Example	Render time uncertain by delaying events and executing them in a bulk. Thus, the events are building a kind of uncertainty set with respect to the actual time and thus, the users triggering the events are building an anonymity set.
Example	Store only uncertain location information. Thus, several users are possibly located in the area and are building an anonymity set.
Example	Render behavior uncertain by disclosing only short location tracelets instead of large location traces. Thus, several users can cause the tracelet and build an anonymity set.

Subcondition 2: Facts of fact type B are sensitive. In order to assure that subcondition 2 is false, the privacy engineer must assure that facts of fact type B are not sensitive.

Concept	Design facts of fact type B such that they are no longer sensitive.
Example	Use the home address not from user's home network, but from any arbitrary network without relation to the user.

Subcondition 3: The attacker knows the mapping of the functional dependency. In order to assure that subcondition 3 is false, the privacy engineer must avoid that the attacker can know the mapping. A sample mechanism is:

Concept	Avoid the availability of mapping functions in publicly accessible directories.
Example	Assure that a telephone directory can only be accessed in one direction.
Concept	Include facts of fact type B in an encrypted way in facts of fact type A, so that only trusted nodes can decrypt facts of fact type B, i.e., can know the relation between facts of fact type A and facts of fact type B.
Example	Encrypt the user's identifier, which is shared with a trusted location server in identifiers given to untrusted location based services. Then the location based services do not know to which identity the transaction is mapped. Such a concept is described in [102], [103].

Another possible protection would be not to disclose the respective fact to the attacker. If this were possible, this would already have been handled in 4.1.1. The same argumentation holds true for the next section, which handles the protection against merging interpretations.

4.1.2.2 Avoidance of Fact Set Links

Goal

The goal is to avoid that disclosed facts allow for the merging of several fact sets about the same user.

Reasoning

Linking several fact sets is another way by which attackers can enlarge their knowledge about a user. Examples for such fact sets are several care-of addresses or several location tracelets.

Candidates

Candidates for this section are fact types, which map uniquely to the real name fact type and whose facts can be contained in fact sets, which should remain unlinked.

Protection

The protection aims at making false one of the subconditions of section 3.3.1.4 being necessary for a linking vulnerability to exist. Thus, each subcondition is considered separately in the following. From the pure point of view of VID protection, falsifying of the different subconditions is equivalent. Thus, the privacy engineer has usually several possibilities for protection and should choose the best one in the context of the concrete system under construction.

Subcondition 1.1: There exist two instantiations of an identical fact type in two fact sets. In order to assure that subcondition 1.1 is false, the privacy engineer must avoid the disclosure of two instantiations of the same fact type in several fact sets about the user. A sample mechanism is the following:

Concept	Partition fact sets in a way that none of the considered attackers or group of attackers can see different fact sets containing facts of the same fact type.
Example	Consume all services requiring the location from the same provider and with one VID. Then no other provider observes the location and there are not two VIDs containing the location.

Subcondition 1.2: Both instantiations are indeed the same fact. In order to assure that subcondition 1.2 is false, the privacy engineer has to make sure, that if facts of the same fact type have to be revealed in several fact sets, these facts must not be identical. A sample mechanism is the following:

Concept	Assure, that one instance per VID of the given fact type exists.
Example	Use a different home address for each VID.

Subcondition 1.3: The fact type functionally determines the real name fact type. In order to assure that subcondition 1.3 is false, the privacy engineer must make facts of the real name fact type, i.e., users, ambiguous for facts of the considered fact type. This is a specialization of subcondition 1 in 4.1.2.1. Thus, similar mechanisms can be used and applied to the real name fact type as the functionally determined fact type.

Subcondition 2.1: Condition 1 is true with all subconditions. Assuring that subcondition 2.1 is false, is equivalent to assuring that one or more subconditions of condition 1 are false.

Subcondition 2.2: The real name fact type functionally determines the considered fact type. In order to assure that subcondition 2.2 is false, the privacy engineer must make facts of the considered fact type ambiguous with respect to the relation from the fact of the real name fact type, i.e., for the user. A sample mechanism is the following:

Concept	Split the user's context into several independent contexts.
Example	Use several VIDs. Then the real name does no longer functionally determine facts of a single VID. It could also map to the facts of every other VID of this user.

After description of the general methodology, the next section applies the methodology to Mobile IPv6. The result of this process is an improved architecture for mobility management, which is specially designed for VID protection.

4.2 Application of the Methodology to Mobile IPv6

For improving the system, this section uses the identified vulnerabilities of Mobile IPv6 being discovered in chapter 3 and protects the system against them one by one. The protection goals as well as the attacker model are identical to those in chapter 3.

The goal of the new architecture is, that the communication system does not reduce the protection of the VIDs. Thus, the necessity of the change of a VID can be decided purely on application level, not taking the communication system into account. If the communication system could trigger the change of a VID due to a sensitive amount of disclosed information, some things could be realized differently.

The result of this section is a conceptual architecture of a mobility management building block. This means, that it can be mapped to various concrete system architectures, e.g., Mobile IPv6, the Host Identity Protocol [160], Flying Freedom [68], the Internet Indirection Infrastructure [212], its mobility extension [235], Hierarchical Mobile IPv6 [207], or many others. Here, it is mapped to Mobile IPv6.

According to the chosen concrete system architecture for realization, the concepts will map to different instantiations. The care-of address is in principle an instantiation of the concept of an arbitrary locator indicating the user's network attachment point for routing. The home address is an instantiation of the concept of an arbitrary identifier uniquely identifying the user in the communication system. In the same way, the entities of the communication system can map to agents of different concrete system architectures and symmetric encryption, for instance, can map to IPSec or other equivalent technologies as well. Regarding encryp-

tion, the only relevant property here is, whether it is symmetric—thus, fast and with a unique identifier indicating the receiving host, which key to use for decryption—or whether it is asymmetric—thus, slow but without unique identifier, because the receiving host can decrypt it with its single private key.

In section 4.2.1 the improvement is started by minimizing observation possibilities. Then, section 4.2.2 shows the limitation of interpretation possibilities. In section 4.3 this chapter is summed up by showing the resulting architecture and by discussing it.

4.2.1 Observations

The first step for avoiding observations is to apply symmetric encryption between the Correspondent Node and the Mobile Node. This avoids observation of facts of the fact type "application data", which is a representative fact type as hook to models of applications. With this encryption, the application layer does no longer disclose any facts or fact sets. Those application layer facts are not in the scope of this thesis. Nevertheless, the encryption is relevant here, because such a symmetric encryption typically introduces an identifier, indicating the receiving node which key to use for decryption. In IPSec this identifier is called Security Parameter Index, SPI. This abbreviation will be used from now on.

There might be more packet header information that allows to link packets of the same flow between the Correspondent Node and the Mobile Node, e.g., the packet sequence number. From a VID perspective, this information is equivalent to the SPI. They are allowing to link packets to the same user but they are not containing any more sensitive information. Thus, in the remainder, only the SPI is considered as a deputy fact type for all those fact types.

The next subsection describes the avoidance of the disclosure of facts being revealed by the communication system. After that, 4.2.1.2 presents the partitioning of observations, which cannot be concealed completely.

4.2.1.1 Avoidance of Observations

This section starts by describing the avoidance of all facts of a given fact type. After that, the disclosure of facts of those fact types is minimized.

Candidates

The target for this type of protection are fact types, which are directly or indirectly sensitive and which potential attackers can observe. According to Table 3.1, these are the following fact types; the list additionally contains the SPI between the Correspondent Node and the Mobile Node, which was introduced by the symmetric encryption between both nodes.

- home address
- care-of address
- VID-identifier
- MAC address
- SPI of packets between Correspondent Node and Mobile Node, SPI_{CNMN}

Protection

The system needs most of the listed fact types for operation. The only fact type, whose facts can sometimes be concealed, is the care-of address. The care-of address is not necessary to be visible to the Correspondent Node as potential attacker and to the $\text{Eavesdropper}_{\text{MNHA}}$.

The chosen concept for protection against the $\text{Eavesdropper}_{\text{MNHA}}$ is to use asymmetric encryption between the Mobile Node and the Home Agent for signalling the care-of address. Asymmetric encryption is used here, because only small amounts of data have to be exchanged and because this type of encryption does not require the Mobile Node to be identified in the non-encrypted part of the message, i.e., it does not require an SPI.

The chosen concept for protection against the Correspondent Node as potential attacker is to change the system design so that disclosure of the care-of address is avoided. This is achieved by forcing the Mobile Node to use anonymous sender addresses for packets sent to the Correspondent Node. For indicating to the Correspondent Node, which communication partner is the sender of the packets, the Home Address of the Mobile Node is contained in the sent packets in an encrypted way.

The use of anonymous sender addresses might cause problems with network layer filtering. Network layer filtering is not done by all providers, today. If such a filtering was done, it would be necessary, e.g., to register the anonymous address by the access router doing the filtering. This is a problem similar to firewalling and the use of the home address in roaming networks [172]. Another drawback is the fact that ICMP error messages cannot be received, which is not followed any further.

This protection possibility is chosen, because it is the simplest one. There are other possibilities, e.g., using anonymizers like JAP [125]. Another possibility is to use changing proxies for outgoing communication and changing sender IP addresses for receiving incoming error packets. Because anonymization of outgoing communication is a long known problem with many solution approaches, cf. section 4.4.2, the simplest approach is chosen here in order to focus on the methodologies.

There are no more observable fact types, whose disclosure can be completely avoided. Nevertheless, it is possible to minimize disclosure of the care-of address to the Home Agent. The Home Agent does only have to know the care-of address, when a packet to the Mobile Node arrives indeed and must be forwarded. In silent phases, there is no need for the Home Agent to know the care-of address.

The chosen protection approach here is secret splitting or secret sharing [192]. In order to show the principle, again a simple mechanism is chosen. Thereby, the Mobile Node splits the care-of address into several so-called shares. These shares are stored at different *Shareholders, Shh*. For recombination of the care-of address, all shares have to be combined by XOR operations. Thus, no share alone contains any sensitive information, but all shares together allow for recombination of the care-of address. More advanced schemes requiring only a certain number of Shareholders to cooperate are also possible [192].

When a packet for the Mobile Node arrives at the Home Agent, the agent gathers the shares from the Shareholders, recombines the care-of address and delivers the packet. Here, the agents are considered trustworthy with respect to correct functionality. Moreover, access

control to the shares is neglected. This is a standard problem outside the scope of this thesis. Some approaches regarding share management can be found in [121] and [167].

Iteration

The secret sharing mechanism introduces a new vulnerability. A long trace of share update facts functionally determines the user, because the share updates directly depend on the care-of address updates and thus on the user's unique behavior. No attacker can know the mapping of this dependency.

Furthermore, the protection introduces some new potential attackers. At first, these are the Shareholders as agents of the communication system itself. Moreover, there are two new potential eavesdroppers—a set of $\text{Eavesdroppers}_{\text{ShhMN}}$ listening between the Shareholders and the Mobile Node and an $\text{Eavesdropper}_{\text{HAShh}}$ listening between the Home Agent and the Shareholders. If located near the Mobile Node or the Home Agent respectively, they could eavesdrop all shares and thus recombine the care-of address.

For the protection against the $\text{Eavesdropper}_{\text{ShhMN}}$, the Mobile Node encrypts asymmetrically the packets to the Shareholders. Asymmetric encryption is possible, because the Mobile Node only sends share updates, which are small amounts of data. Moreover, asymmetric encryption does not need an identifier indicating the sender to the recipient. Thus, there is no hint to the sending Mobile Node in the non-encrypted part of the message.

For the protection against $\text{Eavesdropper}_{\text{HAShh}}$, the Home Agent and the Shareholders encrypt exchanged packets symmetrically. Asymmetric encryption would also be an option here, but the SPI being disclosed by symmetric encryption does not bring any drawback here. Thus, the faster encryption process is feasible.

Figure 4.1 shows the new architecture after these steps. The boxes are indicating components of the communication system, i.e., the Home Agent and the Shareholders. Circles are showing the communication partners, i.e., the Correspondent Node, CN, and the Mobile Node, MN. The arrows are indicating the message flow between the entities with the direction of the flow. The arrows are marked by the fact type—here by the addresses—by which the Mobile Node is addressed by the entities participating at the respective message flow.

The message flows are as follows. The CN sends packets destined to the home address. Those packets are intercepted by the Home Agent. The Home Agent asks the Shareholders about the current shares and reconstructs the care-of address. Subsequently, the packets are delivered to the care-of address of the MN. The MN updates the shares at the Shareholders. Thereby, the MN is addressed by the home address. Moreover, the MN can directly update the care-of address at the Home Agent when a communication flow is enduring during a handover and thus it is already foreseeable that the Home Agent will also have to know the new care-of address. For sending packets the MN uses an anonymous sender address. For description of the encryption, it is referred to section 4.3.1.

The next section shows how to partition observations, which cannot be avoided. It will improve the architecture further.

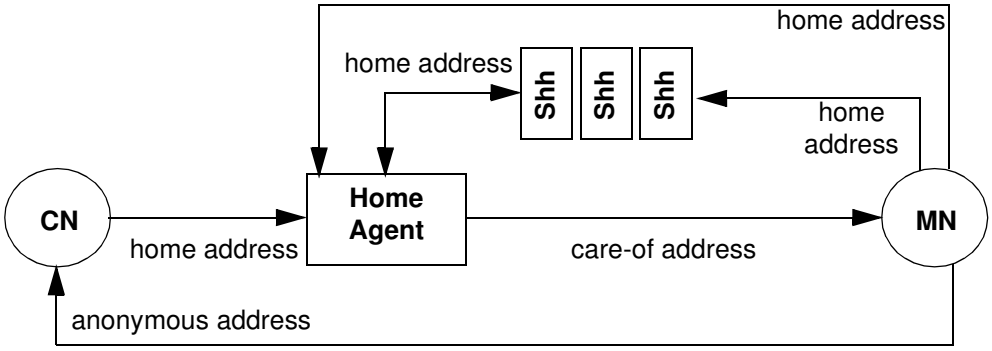


Figure 4.1: Step 1 after avoidance of observations

4.2.1.2 Partitioning of Observations

The partitioning aims at reducing the fact sets about a user, which one single potential attacker can see. Again, the starting point is a list showing the candidates for protection by partitioning.

Candidates

According to 4.1.1.2, the candidates are observable fact types, whose facts can be grouped with other facts. The list below shows which components as potential attackers can observe which fact sets in the current step of the architecture:

- Correspondent Node: VID-identifier and home address

  The Correspondent Node needs both facts in a fact set, i.e., needs to know by which home address to address the VID being identified by the VID-identifier. Therefore, no partition of this fact set is possible.

- Shareholder: Home address and trace of shares

  This fact set is not sensitive. The threat originating from the uniqueness of the share trace is also caused by the home address. Thus, the fact set is not more sensitive than the single fact of the home address.

- Home Agent: Home address, Shareholder addresses, and trace of care-of addresses

  The trace of care-of addresses is no longer permanent without interruptions, because care-of addresses are not revealed during silence times. But still, the trace is not restricted with respect to the overall cardinality.

  The large location trace being inferred from the care-of address trace is sensitive. This is amplified by the knowledge of the home address, which is unique and constant for the user. Those facts have to be separated. For now it must be assumed that the Shareholder address will contain sensitive information, because they might be located near the Mobile Node in order to reduce signalling delay. The Shareholder addresses cannot be separated from the other facts by feasible measures and will be protected in later steps.

There are two partitions to be made. First of all, the home address is to be separated from the care-of addresses, which is protection of a heterogeneous fact set. Second, the long care-

of address trace must be split into smaller tracelets, which is protection of a homogeneous fact set.

Protection

The Home Agent needs the home address and the care-of address for two distinct functionalities. With the home address, packets destined to the Mobile Node are received. With the care-of address, the Home Agent delivers packets to the roaming Mobile Node. The chosen protection therefore, is distribution of trust by separating functionalities.

This splits the Home Agent into two parts. The first part—from now on called *fixed Mobility Agent, fMA*—receives the packets addressed to the home address. Therefore, it must be placed at a fixed place in the network topology, i.e., in the network of the home address. The second part—from now on called *variable Mobility Agent, vMA*—receives the packets from the fMA and delivers them to the care-of address. For this, it collects the shares from the Shareholders.

All three agents need a handle to refer to the Mobile Node. This handle should be anonymous not disclosing anything about the user's identity. It will be called *Temporary Handle, TH*, because it does not need to remain constant forever. There are no more requirements on the TH. Thus, it can be designed arbitrarily.

It is also possible that the fMA requests the shares simultaneously with sending the packets to the vMA. Assuming an equal distance between all agents, this would save one hop delay, in which the vMA requests the shares. This solution is not followed, because it would require to modify both agents, the fMA and the vMA, by adding signalling capabilities. In the chosen realization, the fMA can remain nearly a traditional Mobile IPv6 Home Agent in the sense that it can almost be used unmodified from standard systems. If the TH is realized in form of an IP address from the network of the vMA, it can even be a completely common Mobile IPv6 Home Agent, given the vMA has also Home Agent functionality of collecting packets destined to other nodes.

Figure 4.2 shows the modified architecture after this second step. The message flow is equivalent to Figure 4.1. Still, the vMA is able to receive an infinitely long care-of address trace, given that the Correspondent Node sends packets over a long time. Protection against this threat is the goal of the next step.

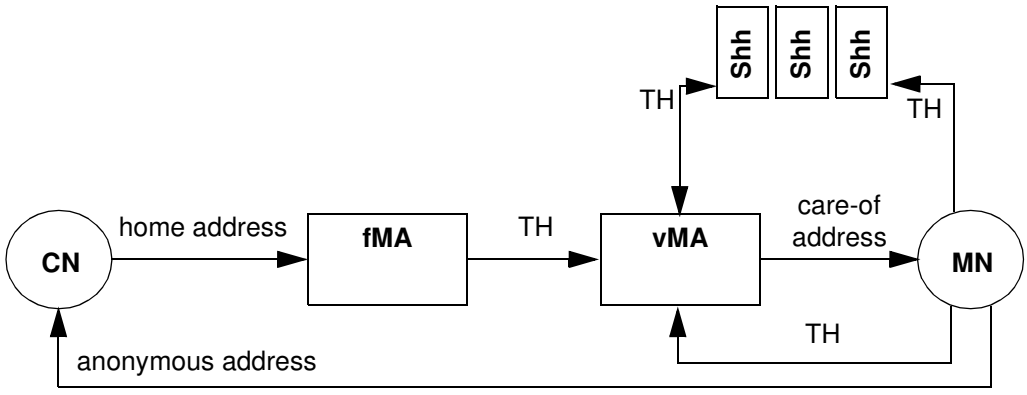


Figure 4.2: Step 2 after partitioning of heterogeneous fact sets

The chosen protection approach against the long care-of address trace is to change the entity, which can observe the variable fact type over time. Here, this means to change the vMA over time. Then, each vMA can only observe a tracelet of care-of addresses. The Mobile Node has to signal the new vMA to the fMA. Dependent on the access control realization of the share requests, it might also be necessary to indicate the Shareholders the current vMA in charge. The new vMA must receive the TH and the addresses of the Shareholders from the Mobile Node.

Iteration

Some things are changing by those partitioning steps. First of all, new potential attackers enter the field. This is the fMA, the vMA, the $\text{Eavesdropper}_{\text{fMAvMA}}$ in between, and the $\text{Eavesdropper}_{\text{MNvMA}}$. The $\text{Eavesdropper}_{\text{HAMN}}$ turned into the $\text{Eavesdropper}_{\text{fMAMN}}$. The Home Agent does no longer exist as potential attacker.

For protecting against eavesdroppers, the agents encrypt exchanged messages. Between the fMA and the vMA, symmetric encryption is possible. Encryption between the Mobile Node and the vMA is asymmetric in order not to identify the Mobile Node in the non-encrypted part of the messages. The asymmetric encryption between the Mobile Node and the Home Agent translates into asymmetric encryption between the Mobile Node and the fMA. The symmetric encryption between the Shareholders and the Home Agent translates into symmetric encryption between the Shareholders and the vMA.

Another changed issue is that the Shareholders do no longer see the home address. Thus, the functional dependency of the user from the long share trace—and thus, the share update trace—is relevant. This is protected by the same measure as the long care-of address trace. The Shareholders will be changing over time. A share tracelet can be longer than a location tracelet before becoming sensitive. Therefore, Shareholders may change more rarely than vMAs. The MN has to tell the new Shareholders to the vMA.

There are no more new vulnerabilities, which could be protected by avoidance of disclosure or partitioning of disclosed facts. Figure 4.3 shows the modified architecture after this step. The Shareholders and the vMAs are indicated as changing over time by the three dimensional notation. The Mobile Node now needs to communicate with the fMA in order to signal the new vMA. The rest of the message flow is equivalent to the first steps of the architecture.

4.2.2 Interpretations

This section describes the reduction of the possibilities of potential attackers to interpret the facts, that are still being disclosed. Both relevant ways of interpretation are considered, inference of new facts in 4.2.2.1 and linking of fact sets in 4.2.2.2.

4.2.2.1 Avoidance of New Fact Inferences

The topic of this section is to avoid that facts and fact sets, which the architecture discloses, allow for inferences of additional facts. After identifying the candidate relations for evaluating with respect to inference vulnerabilities, this section will improve the system accordingly.

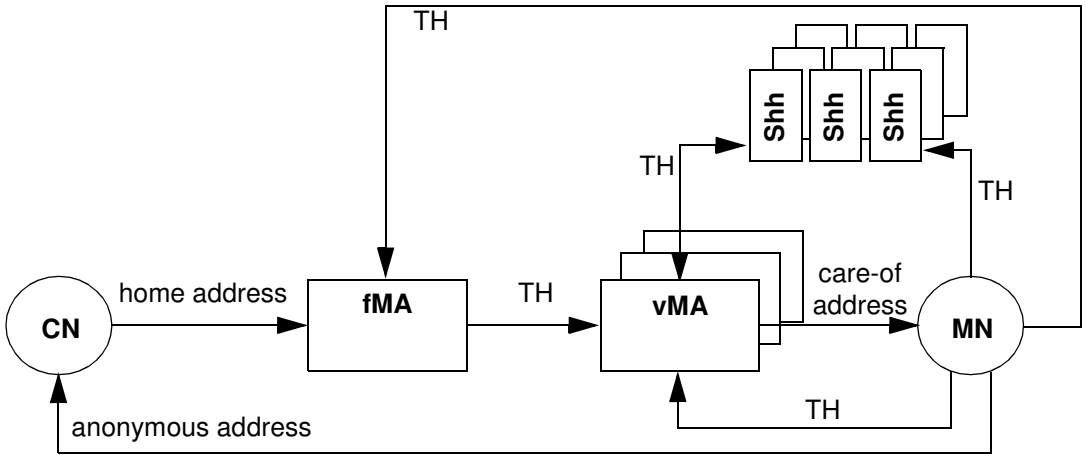


Figure 4.3: Step 3 after partitioning of homogeneous fact sets

Candidates

According to 4.1.2.1, the candidates here are relations with solid arrowheads, which are pointing from fact types, whose facts an attacker can know—either by observation or by inference from other known facts—to fact types, whose facts are directly or indirectly sensitive.

Table 4.1 shows the candidates. The location of the Shareholders and of the vMA do not necessarily map to the roaming location. Nevertheless, in network engineering, it is a common approach to locate such agents at least to a certain degree near to the roaming location in order to reduce packet delay, e.g., in [207]. Thus, the coherence of the agents and the roaming location is assumed for now.

The inference of the roaming location from the care-of address instead, is an inherent problem of a locator address, which serves for routing. Therefore, no protection of this inference is chosen here. Inference from the VID-identifier to the home address is necessary for the operation of the system. Thus, this inference is not prohibited, either.

Protection

Protection here aims at making the second subcondition for inference vulnerabilities false, i.e., avoiding sensitivity of the inferred fact. Inference of the network from an address cannot be avoided without changing the addressing structure of the Internet Protocol. Thus, inference from the network to sensitive information must be avoided.

To achieve this, the user is supposed to choose the agents of the communication system from networks being arbitrarily distributed over the Internet. Whereas this increases delay times, it prevents attackers from inferring any information from those addresses. The addresses of the vMA and of the Shareholder do no longer relate to the roaming location. Similarly, the address of the fMA as well as the home address, which is from the network of the fMA, do no longer relate to the user's real home. Thus, the home address becomes a pseudonym of the user on the network layer and is called *permanent address, PA* from now on.

Known Fact Type	Inferred Fact Types
home address fMA address	home network
home network	home location home provider
care-of address[a]	roaming network
roaming network[a]	roaming location roaming provider
vMA address[a]	vMA network
vMA network[a]	vMA location vMA provider
vMA location	roaming location
Shareholder address[a]	Shareholder network
Shareholder network[a]	Shareholder location Shareholder provider
Shareholder location	roaming location
VID-identifier	home address
location trace	location tracelet personal attributes, both classes
personal attributes, class 2	real name
real name	personal attributes, class 2

Table 4.1: Candidates for inference of new facts

[a] similarly for tracelets and traces

The latter three inference possibilities of Table 4.1 are outside the communication system and thus cannot be protected here. They are only relevant if large location traces can be revealed. Therefore, section 4.2.2.2 will avoid that attackers can link location tracelets to a large location trace.

Iteration

The protections in this section do neither introduce any new potential attacker, nor any new fact type. The attacker model states that all entities of the communication system are untrusted. Thus, the change of the agents into arbitrary networks—and consequently to arbitrary providers—does not change the situation. Therefore, no iteration is necessary. The next section avoids linking vulnerabilities on the same architectural basis from Figure 4.3.

4.2.2.2 Avoidance of Fact Set Links

The final step of the methodology is to avoid that attackers can merge fact sets, which should remain separated. These fact sets can be separated into two different kinds of sets.

The first kind of fact sets are sets from different VIDs of a user. Linking those fact sets is most sensitive, because it directly undermines the VID approach. The second kind of fact sets are only about one VID. These fact sets are tracelets or traces about one VID, which attackers must not be able to link to a large trace. Whereas such a vulnerability does not allow for linking different VIDs of a user, it still may allow for collecting a too large fact set.

Candidates

Along with 4.1.2.2, the candidates are fact types, which map uniquely to the real name fact type and whose facts can be contained in fact sets, which are to be kept unlinked. At first, the candidates for linking fact sets of different VIDs are listed. Then, a list containing candidates for linking fact sets of one VID is presented.

Candidates for certain links of fact sets of different VIDs are the following ones:

- care-of address
- shares and share-updates
- permanent address
- fMA address
- TH
- SPI_{CNMN} if several VIDs are used with the same Correspondent Node
- MAC address

There are several fact types, which are not 100% uniquely mapped to the real name fact type. Nevertheless, the functional dependency is very strong, i.e., only a few users can have caused facts of the respective fact types. Therefore, those mappings are declared as being certain and are included into the methodology. This is a decision of the privacy engineer.

This argumentation holds true for the following candidates:

- vMA address

  The vMA has to process a lot of packets and has to store them during recollection and recombination of the shares. Therefore, it is assumed that only a small amount of users can be served simultaneously. Thus, an identical vMA from several VIDs will likely point to the conclusion that the same user is the owner of the VIDs.

- vMA update

  Simultaneous events are also an indicator of an identical user. It is possible, but unlikely that two users will simultaneously change their vMA. The point in time of the update event is part of a vMA update fact.

- Shareholder update

  The same arguments hold true for the change of Shareholders.

The vMA address is part of this list, whereas the fMA address is not. The fMA has fewer resource expensive actions to do than the vMA. Thus, the fMA can serve enough users for a sufficiently large anonymity set. Again, this is a rating and a decision of the privacy engineer.

Candidates for links of fact sets of the same VID are the following ones:

- TH
- SPI_{CNMN}
- MAC address

The latter three candidates appear in the lists for different VIDs and for the same VID. The goal is to avoid both, linking fact sets of different VIDs and linking too large fact sets of one VID. Thus, protection does not only aim at avoiding the latter three candidates in fact sets of different VIDs, but even in fact sets, which have to remain unlinked, of the same VID, e.g., location tracelets.

Protection

Again, protection can aim at each subcondition of the conditions for existence of a link vulnerability from section 3.3.1.4. Assuring that subcondition 1.1 is false would require to avoid disclosure of two instantiations of the same fact type in several fact sets about the user, which have to remain unlinked. This is not possible here, because each fact type is necessary for each VID and for each tracelet in order for the system to fulfill its functionality.

Assuring that subcondition 1.2 is false requires that instantiations of the identical revealed fact type must be different for several VIDs or for several fact sets, which have to remain unlinked, of one VID respectively. This protection is feasible for a number of candidate fact types here. The chosen protection concept is to assure that one different instance per fact set, which has to remain unlinked, exists for the respective candidate fact types.

The first list contains the protection steps taken against linking fact sets of different VIDs of a user. There is one measure for each link candidate in the list for links of fact sets of different VIDs. The only exception is the care-of address.

- Use different shares for each VID. This holds for share updates as well.
- Use one permanent address for each VID.
- Use permanent addresses from different fMAs.
- Use one TH for each VID at a time.
- Use one SPI_{CNMN} for each VID. This transforms the SPI_{CNMN} to several SPI_{CNPA}, which are bound to the permanent addresses.
- Use different MAC addresses for the anonymous IP addresses.
- Use one vMA for each VID.
- Change the vMAs of different VIDs at different points in time.
- Change the Shareholders of each VID at different points in time.

The following list shows the protection measures to avoid links of too large fact sets of one VID. Again, there is one measure for each candidate in the list above.

- Change the TH of each VID over time, so that different tracelets contain different THs.

  The point in time should coincide with a care-of address change. Thus, no attacker will be able to link several THs by an identical care-of address. Thus, it is best to change the TH every n care-of address updates, with n representing the number of care-of addresses allowed to be aggregated in one tracelet. Thus, any attacker in the LAN cannot recognize the Mobile Node on a second visit and link both care-of addresses.
- Change the SPI_{CNPA} with the TH.
- Change the MAC address with the TH. Variable MAC addresses are also proposed in, e.g., [171], [91].

The care-of address directly relates to the user's location. Thus, using different care-of addresses for different VIDs would not bring any protection. The network of the addresses will still be identical and the location inferable. Thus, also the link possibility would not be avoided.

One MAC address for each VID would not make any sense either, because an Eavesdropper in the access network of the Mobile Node can only see the MAC address, when it sees the care-of address, too. The permanent address does not need a MAC address, because it will never be visible in the LAN.

There is no link candidate left, for which it would make sense to assure that subcondition 1.3 is false. Also for condition 2, no additional protection steps to the ones undertaken for assuring that subcondition 1.2 is false, are necessary. This means, that there are no vulnerabilities left regarding the link of fact sets.

Iteration

The protection steps introduce several new update fact types, which require an iteration for protection. Identical points in time of TH updates, of SPI_{CNPA} updates, and of share updates of different of a user's VIDs allow for linking fact sets. These links are not 100% certain, because there might be several users updating their data simultaneously. Nevertheless, protection measures are planned, because the probability of several users updating their data simultaneously is rated low. Thus, those links are declared as being certain by the privacy engineer. The protection steps are as follows:

- Choose different TH update points in time for different VIDs.
- Choose different SPI_{CNPA} update points in time for different VIDs.
- Choose different sets of Shareholders for different VIDs.

Those are the last steps for completing the methodology. The next section is summarizing the resulting architecture, which has been created by applying the methodology to Mobile IPv6.

4.3 New Architecture

At first, section 4.3.1 describes the final architecture with a focus on the applied encryption techniques. Then, section 4.3.2 shows message sequence charts of the communication path including share signalling as well as the signalling for updates of the specific data elements. After that, section 4.3.3 describes the proof of concept by a prototypical implementation. Finally, section 4.3.4 discusses the architecture from a broader view than VID protection.

4.3.1 Encryption View

Figure 4.4 shows the final architecture, which was presented in different stages [100], [101]. Orange arrows indicate asymmetrically encrypted communication. Blue arrows indicate symmetrically encrypted communication. The blue dashed arrow shows that the SPI of the symmetric encryption is changing over time. The solid arrow between the fMA and the vMA indicates that the TH must be communicated in a protected way in order to be concealed against eavesdroppers. All arrows are labelled with the address used as sender address, with the recipient address, and with the communicated piece of data after the colon. *anon* here stands for an anonymous IP address. The direction of the arrows indicates the direction of the message flow. The names of the agents and the Correspondent Node stand for their respective addresses.

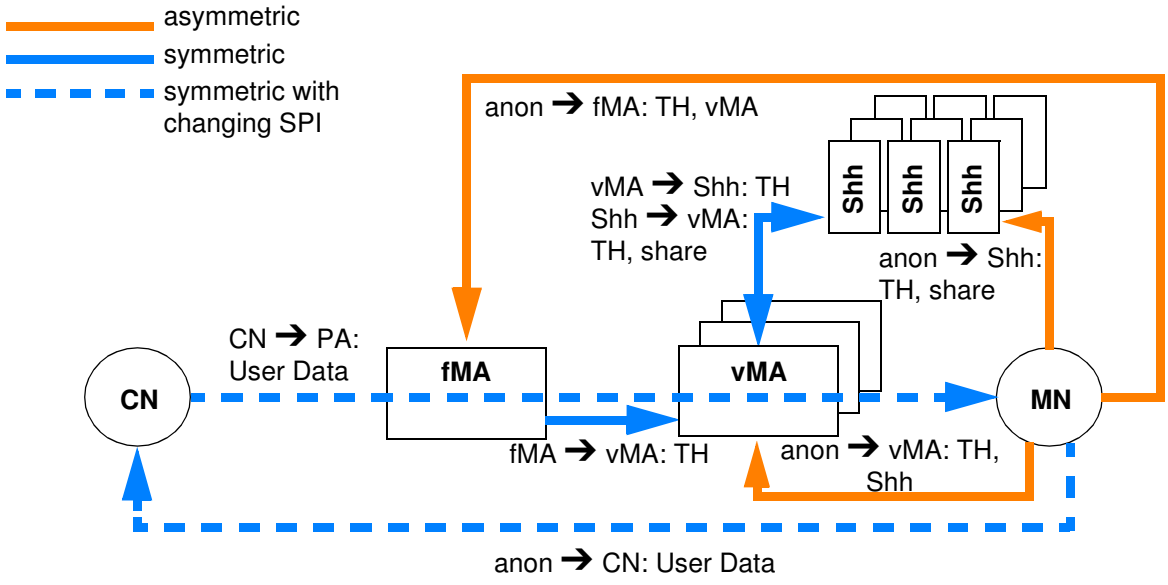


Figure 4.4: Final Architecture

In general, symmetric encryption is used between all agents of the communication system and between the communication partners. Thus, the payload of the packets is concealed against eavesdroppers between the acting entities. Whereas asymmetric encryption between the agents would be possible for the low amount of signalling data, it would not bring any benefit to VID protection. The Mobile Node encrypts signalling to the agents of the communication system asymmetrically. Thus, no identifier of the Mobile Node has to be visible in the not encrypted part of the packets. By avoiding such identifiers, attackers can espe-

cially not recognize the Mobile Node—or the respective VID used—and link several observations as belonging to the same user.

The next section shows, how the entities of the architecture cooperate. Therefore, the messages flowing between the entities are described.

4.3.2 Functional View

Figure 4.5 shows the communication flow. The Correspondent Node sends a packet to the permanent address of the contacted VID. This address is from the network of the fMA and thus routed to there. The fMA receives the incoming packet on behalf of the Mobile Node like the Mobile IPv6 Home Agent receives packets of a Mobile Node being not in the home network. Then, the fMA looks up the TH of the respective permanent address in its internal database and forwards the packet with the TH to the vMA. The vMA queues the packet, looks up the respective Shareholders of the TH in the internal database and requests the shares. On reply of the shares, the vMA recombines the care-of address and finally delivers the packet.

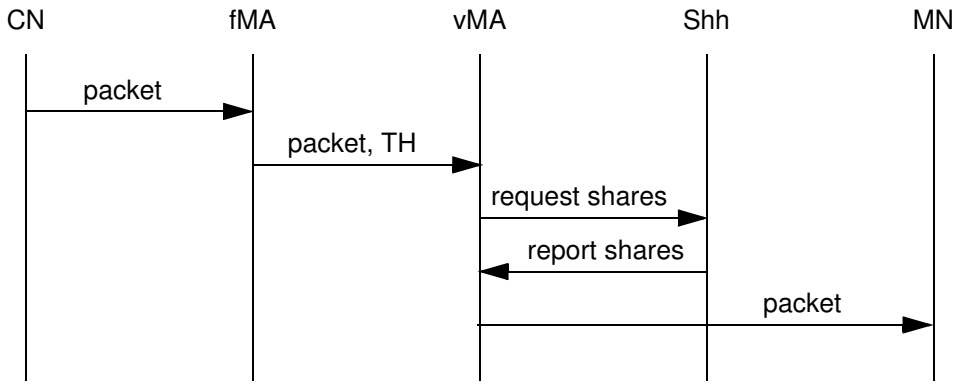


Figure 4.5: Communication Flow

Figure 4.6 shows the messages exchanged after a movement to a new network. In a first step, the Mobile Node changes several elements:

- the MAC address, which will be used for the new care-of address in the new network
- the SPI_{CNPA}, which is used for the encryption with the Correspondent Node
- the TH
- the vMA
- the Shareholders
- the shares, which are computed from the newly assigned care-of address

The vMA and the Shh represent the respective new agents after the update. The Mobile Node simultaneously sends the updated information. The Correspondent Node receives the SPI_{CNPA}. The fMA receives the updated TH and the new vMA. The vMA receives the new TH and the new Shareholders. The new Shareholders receive the new TH and the new shares. The agents are not assumed to confirm received updates. This can be included in a later step for improving robustness. The same applies for deregistration at old agents,

whereby this can be omitted intentionally in order to blur the point in time of the update towards some potential attackers.

Not all those steps are executed on all movements. The SPI_{CNPA}, the TH, the vMA, and the Shareholders only have to be updated every n movements, with n being the maximum number of care-of addresses allowed to be in a tracelet. If the Mobile Node does not change any of those elements, it is not necessary to send the respective update messages.

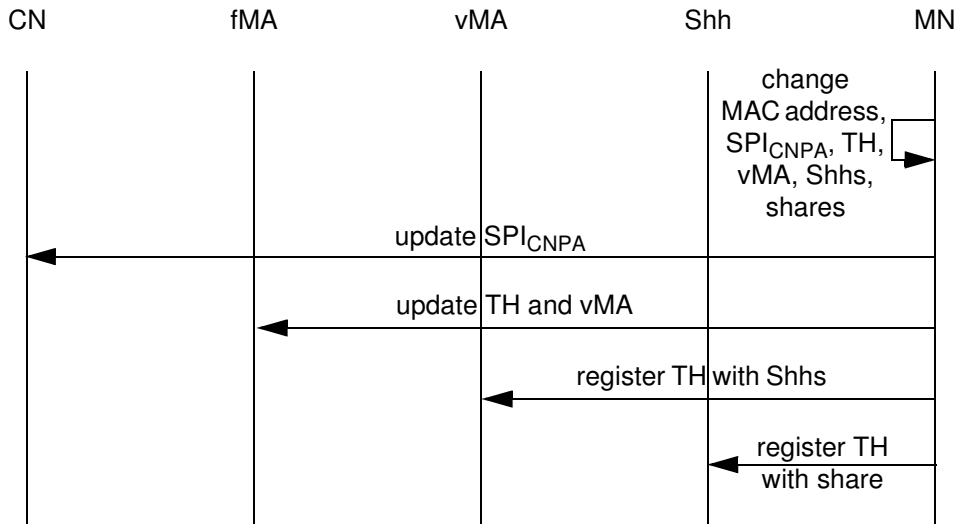


Figure 4.6: Updates on movement of the Mobile Node

4.3.3 Prototype

The core parts of the architecture have been realized in a proof of concept prototype based on Linux systems in the laboratory. This prototype shows feasibility of the communication flow for one VID. Analysis of the exchanged packets shows, that indeed only the data, which is assumed in the evaluation of chapter 5, is visible to eavesdroppers. The signalling aspects have not been considered in the prototype. They are considered a simple problem, which can be solved by new protocols above IP. Thus, the realization does not touch any existing piece of code, neither is it necessary to change anything in the network stack. The simple secret sharing technique based on XOR operation has proven feasible in an earlier prototype for message relay on application layer.

The realization showed that the Correspondent Nodes can remain common IP nodes with their native network stacks in the kernel unmodified, if not using IPSec with dynamic key exchange between the Correspondent Node and the Mobile Node. If dynamic keying is necessary, the key exchange by IKE [98] has to be modified in order not to disclose the permanent address to all eavesdroppers and the to vMA during negotiation of the security association. Then, the Correspondent Node will have to follow this adapted key exchange method.

Most of the functionalities of the agents can be achieved by configuration of standard IPv6 means, e.g., tunnels. The signalling has to be newly implemented. Moreover, the vMA must queue the packet and recombine the shares after collection, which requires new functionality. Details of the prototype can be found in [80] and in parts from an earlier prototype in [121].

4.3.4 Discussion of the Architecture in a Broader Scope

Protection of the VID approach is the number one requirement for the architecture. Nevertheless, there are several other requirements to a communication system. Often, trade-off decisions between requirements are necessary. Because of the attacker model here, which does not trust entities of the communication system with regard to abuse of sensitive information, the architecture requires a lot of compromises.

First of all, scalability is an important issue in communication systems. The number of agents, which are needed for one user is larger than in the original system Mobile IPv6. Nevertheless, a user's application data traffic is distributed among several agents. The sum remains roughly the same not considering the additional signalling traffic. Thus, all agents together can serve roughly the same number of users as in a Mobile IPv6 scenario with the same number of Home Agents. As long as several VIDs are not communicating simultaneously, they can use the same vMA, which reduces the number of necessary vMAs per user by a statistical multiplexing gain. This effect is further examined in chapter 6.

The architecture introduces longer packet delay times than plain Mobile IPv6 does. This is first of all due to the additional hop between the fMA and the vMA, secondly due to the collection of the shares, and finally due to the fact, that the agents are from arbitrary networks. The latter could require packets travelling long distances between the agents. Moreover, route optimization cannot be used unless the Correspondent Node is trusted.

Compared to those delay times, the computing time for management of the shares is assumed to be negligible. Nevertheless, if more sophisticated cryptographic secret sharing schemes are introduced and an access control is done on the share requests, there might be a considerable computational overhead, too.

Collection of the shares can be reduced to be only necessary at the first packet sent to a VID. Afterwards, the vMA knows the care-of address until it changes. Compared to some channel setup times at lower layers, especially in the radio field, this delay can then be claimed as acceptable.

The architecture introduces several handovers, due to the changes of network and data elements. The handover delays are not accumulated, but happen in parallel. Nevertheless, those delays could have a negative impact on protocols like TCP. For a decrease of handover delays, the Mobile Node can update a new care-of address directly at vMAs, which are already serving communication flows destined to the Mobile Node. This makes only sense if the vMA and the other elements are not changed on the movement.

The architecture allows for a large range of configurability. Privacy aware users can use it like it is described in this chapter. Users not taking care about their privacy, can configure the system to use less servers and to have a better performance. The agents are merely logical functions and can physically be located all on the same entity from a purely functional point of view. Whereas this would render the distribution of trust aspect useless—if not considering several virtual machines operated by different parties on the same physical machine—it would increase scalability as well as performance. Such a configuration would be very similar to plain Mobile IPv6 then. The architecture allows for arbitrary steps between those two extreme configurations, e.g., only to merge the vMA and the Shareholders on one machine, but still to keep an own fMA or to use only one Shareholder instead of

several Shareholders, or to combine the fMA and the vMA on one machine, but still to use one or several Shareholders.

With sophisticated algorithms for the time instances of agent changes, the user can more exactly tune the information, which the agents can observe. One such possibility is to consider sensitivity of the locations for determination of the allowed tracelet cardinality. The larger the tracelets may be, the more rarely the agents have to change and the fewer agents are necessary. This trade-off will be considered deeper in chapter 6. Nevertheless, algorithms are not deepened in this thesis.

For giving incentives to possible agent providers, means of charging the user would be beneficial. Standard approaches for this would require to identify the user. There are solution approaches of anonymous credentials, which could authorize a user without identification, e.g., [14], [31], thus providing the base for a charging mechanism. The same approaches could be used for controlling access to the shares.

The focus of the new architecture lays on the privacy protection. Before bringing the system into the field, it would have to be secured, as well. This is out of the scope of this thesis.

4.4 Related Work

This section sketches related work in the area of the system improvement methodology in 4.4.1 and in the area of privacy enhancing communication architectures in 4.4.2.

4.4.1 Methodology

Like for the evaluation methodology, there is no architecture design methodology to improve the VID protection of a system known to the author. There are again some methodologies from the security and privacy area but with a different focus. From those, a selection is described in this section. While there exist some proposed formal methods for small and specific problems, mathematically formulated methods cannot support the design of larger systems because of the increased complexity and the increased number of dependencies [198].

[17] gives an overview of early design methods and classifies three generations: Checklist methods, mechanistic engineering methods and logical transformational methods. [56] gives an overview in form of a roadmap of challenges towards software engineering for security.

Some approaches try to give a reasoning for deriving balanced requirements to a system. [137] and [114], e.g., discuss on an argumentative level how ubiquitous systems should be designed. [89] does the same for network security. [7] is on a comparable level of abstraction and focuses on continuous consideration of security during the complete system development life cycle. [213] gives prose guidelines for designing secure IT systems. The methodology in this thesis is stronger formalized and more focused.

Many of the approaches rely on the provision of best practices, which proved valuable in past system design procedures. They are more or less formalized with different foci, e.g., on building a Liberty Alliance [217] system [224], or on IT systems, e.g., [28], which has already been introduced in section 3.5.5. Another example in [161], [162] has already been

introduced in section 3.5.5, too. The methodology in this thesis is stronger formalized and more concentrated in its scope.

[57] defines two informal methodologies to design controlled anonymous applications. In the first one, which is inductive, the authors are starting with a specification of the actions of the system. Then, they build a so-called credential option matrix containing the scenarios over the credentials to be shown in each scenario. Finally, this matrix is evaluated by identifying accountability requirements for each scenario and comparing it with the credential matrix. The scenarios without conflicts can be built.

The second methodology, which is deductive, starts with a functional requirement analysis of the system to be designed. The second step is to identify risks in the functionality. For tackling risks, several options for countermeasures exist usually. With the chosen countermeasure, the functionality is again evaluated with respect to the contained risks. Both methodologies can be combined. The methodology in this thesis is stronger formalized and not focused on controlled anonymity.

Often these best practices consist of a high-level procedure model, which assures that the system designers consider a given set of aspects, e.g., intrusion detection or firewalling. Then, there are common patterns identified and solutions for those patterns provided. This idea is similar to the methodology in this thesis, but the procedure models usually are less formalized than here.

Patterns obtained a great attention in the research community. According to [198], patterns use expert know-how in a structured way. The pattern approach describes common problems, which are to be solved in many systems, and provides known solutions for those problems. There is a lot of pattern literature available about security patterns, e.g., [210], [197], [195], [76], [21], [95], [186], an evaluation in [138] and one of the earliest works in [232]. Privacy patterns are described, e.g., in [187], [196], [188], [194], [94].

There is also work in detailing certain steps of the system development cycle. [63], for instance, describes a programming language with adapted language concepts for systematically transforming formalized top-level specifications into executable programs. [127], [128] provide an extension to the UML modelling standard for capturing security aspects. This forces developers to consider security already in the modelling stage. These works have different goals than this thesis has.

The next section discusses related approaches of privacy enhancing communication architectures.

4.4.2 Architectures

Anonymity and location privacy are a large area in research about communication networks and resulted in a large number of proposed system architectures. Most of the systems are focusing on anonymity and on unobservability of the association of sender and recipient. The architectures are usually proposed for non-mobile users. This section gives an overview on those architectures gaining the largest attention in the community.

The related architectures can be separated in architectures not explicitly aiming at mobile users and in architectures, which have the support for mobile users as their core require-

ment. In section 4.4.2.1 the first class is described, whereafter section 4.4.2.2 shows the class for mobile users.

4.4.2.1 Privacy-Enhancing Architectures for Non-Mobile Users

The architectures in this section aim at sender anonymity and usually also at anonymity of the recipient to a certain degree. Moreover, some of the architectures aim at hiding the relationship between the sender and the recipient. As those protection goals are different from this thesis, the architectures are only discussed briefly.

The proposed architectures can first of all be structured according to the underlying technology. In earlier works, this was ISDN and GSM, e.g., [75], [134], [12]. Later proposals focus on IP technology. This thesis is based on IP and thus, only the latter architectures are discussed.

The architectures can further be subdivided in client-server based infrastructures and in peer-to-peer infrastructures. [22] compares both approaches. There are different dimensions in the comparison. The first dimension is, whether all nodes participating at the systems are symmetric, which is the case in peer-to-peer systems, or are asymmetric—e.g., with different functionalities or responsibilities—which is the case for client-server based systems. Second, the flexibility of the routing of the packets through the participating nodes is distinguished. The routing can be defined by the system either in a fixed or in a variable way. The routing can also be chosen by the user, who can be free in choice or who can be bound to certain restrictions.

There are several differences of the proposed client-server based architectures to the new architecture in this thesis. Most of those systems are focusing on unobservability of the relationship between senders and recipients, which is out of the scope here. Some systems are focused on specific applications like eMail or the World Wide Web, whereas this thesis focusing on the IP layer is of general purpose. For the peer-to-peer class, the difference to the new architecture proposed in this thesis is obvious, because they are based on the peer-to-peer paradigm, whereas the new architecture here clearly bases on dedicated servers with a different functionality than the communicating clients.

In the following, at first client-server based systems are shortly presented and then, peer-to-peer systems are overviewed. For a deeper interest, there are several surveys available in the literature, e.g., [47], [190], [23], [57].

[93] describes as the simplest anonymous communication enablers for WWW the Anonymizer [9] and the Lucent Personalized Web Assistant, LPWA [82]. Both are consisting of one server only. [93] categorizes systems consisting of several servers into rerouting based systems, which are routing the packets not on the direct path but via several anonymizing nodes, and in non-rerouting based systems, which are mainly systems based on the mechanism of Dining Cryptographers [34]. The Dining Cryptographers mechanism relies on a superposing broadcasting scheme where only the intended recipient can read the sent message. In the following, only rerouting-based systems are described, because they are closer to the new architecture of this thesis.

The first distinction of rerouting based systems is, whether they are providing for communications with high latency only or also for communications with low latency. The ones for high latency are usually based on MIXes [35]. Those architectures provide for protection of

the relationship between sender and recipient. Basically, a MIX server is an anonymizing intermediate node, mixing messages of different users, so that the sender and recipients of the messages cannot be observed.

There are basically two different flavours of MIX based anonymizing schemes [190]. In the first flavor, there is a fixed cascade of MIX servers, which is passed by traffic of all users. In the second flavor, there is a network of MIX servers and only some of them might be passed and the order of passing might change from user to user. For those MIX networks, it can be distinguished, whether the path is defined by the system or whether it is chosen by the user [22], [23], [57]. In order to improve the anonymity properties of MIX based systems, a lot of extensions were proposed, e.g., [60], [133] [49]. From the beginning, there is a strong German community for the MIXes. Early work based on ISDN is, e.g., in [173], [176]. [125] can also be sorted into the mixing based architectures.

Also widely known but only designed for eMail are the generations of remailers, which are intercepting eMail communication in order to strip off sender-related information and to relay it so that the sender is anonymized. The generations started by so-called Type-I remailers also known as cypherpunk remailers [1], which were followed by so-called Type-II remailers, e.g., [157] and finally by so-called Type-III remailers [48]. Those remailers are also MIX based systems [190], [47]. The generations have a growing protection of anonymity. Sometimes, Nym-servers, e.g., [104] are mentioned as Type-0 remailers. Those servers are not anonymizing, but are providing for pseudonymous eMail communications. The remailers go back to the work of [35].

For low latency communications, there are mainly the approaches based on [86] and its second generation [59]. They improve the performance by letting aside the mixing capability of the anonymizing nodes [229]. Basically, the packets are at first encrypted for each of the nodes on the path resulting in several layers of encryption. Each anonymizing node decrypts the packet on reception and sends it to the next node on the path until finally the packet arrives at the recipient. [24] is basically the same approach but with user chosen routes [23].

Many approaches are focused on sender anonymity in a request based scenario, e.g., in a World Wide Web surfing scenario. This thesis rather focuses on the return path, i.e., on protection of the recipient of packets. Traditional systems for sender anonymity like [35], [86] only provide for one reply to the request sent out by the user to be protected [87].

[87] provides for a system for many anonymous replies, which can even stem from different responders. Moreover, those responders do not know, that they reply to the same person. This is a similarity to the approach of the unlinkable permanent addresses in this thesis. [87] is based on a cryptosystem, where different public keys map to the same secret key. Thus, the user can give different public keys to the different responders and still decrypt all the traffic. From an architecture's perspective, this system is also based on MIXes.

[227] aims at the same goal and uses a similar cryptosystem. From an architecture's perspective, the reply is realized by multicast addresses. All members of the multicast group receive the reply, but only the user, which issues the corresponding request and thus possesses the correct private key, can decrypt the reply. The multicast group thus builds the anonymity set.

In the following, peer-to-peer systems are overviewed. [130] gives a good overview in its related work section. One of the most famous systems is [184], where each intermediate

node forwards the packets with a certain probability to the final recipient and otherwise forwards the packets to another anonymizing node. For replies, the same path is used reversely. [142] is basically the same system with another reply mechanism based on multicast. [156] also uses this routing idea. [230] relies on that principle, too, but extends it by variable forwarding probabilities. Thus, the user can configure a trade-off between anonymity—with many hops before reaching the recipient—and performance—with only a few hops before reaching the recipient. The number of hops is varied by a varying probability of forwarding packets to other anonymizing nodes instead of forwarding the packet to the final recipient.

[81] relies on the principles of [86]. [185] is very similar except for details of the tunnel setup procedure. [130] and [234] can even cope with churn of the anonymizing nodes. Some peer-to-peer based systems are based on broadcast, e.g., [203]. [85] is based on the Dining Cryptographers mechanism and thus also on broadcast. [20] achieves anonymity by making the originator looking like a common node, which only redirects traffic of other nodes. Thus, the generation of new traffic by the sender is hidden. [37] finally is broadly known but focuses on anonymous publishing rather than on generic communications.

4.4.2.2 Privacy-Enhancing Architectures for Mobile Users

The following sections describe systems being explicitly designed for mobile users. They are closer to this thesis. Among anonymity of the communication partners, those systems also provide for a certain degree of location privacy. They are grouped into four classes: Systems with a similar protection as Mobile IPv6, hierarchical systems, systems based on the Internet Indirection Infrastructure, i3 [212], and finally systems being designed for multiple VIDs to a certain degree. The capabilities of the systems regarding protection of the VID approach are compared to the evaluation result of Mobile IPv6, because that is the starting point of the thesis.

Systems Providing Similar Protection as Mobile IP

The systems evaluated in the following offer similar capabilities with respect to protection of the VID approach like Mobile IPv6 does. This is because their original purpose is not protection of multiple VIDs.

The Host Identity Protocol, HIP [160] is a proposal aiming at the integration of mobility, multi-homing and security in terms of signalling authorization. Its principle is an explicit split of the two concepts of locator and identifier. Whereas the identifier indicates, which host is to be addressed, the locator denotes the network attachment point of the host or the respective network stack. The identifier is realized as a cryptographic public key. In order to translate the identifier into a locator, i.e., in an IP address, an address discovery service is used. For mobility support, a so-called Forwarding Agent is used. It receives packets destined to the Mobile Node's fixed locator and forwards it to the current locator. Mobile Nodes are able to signal their current locator to the Correspondent Nodes.

Regarding the protection goals of this thesis, the only difference to Mobile IPv6 is the use of public keys as identifiers, which contain no sensitive information as such. But the Correspondent Node itself resolves these identifiers into the fixed IP address of the Mobile Node and thus, also see this address. This IP address is conceptually similar to the home address serving as identifier in Mobile IPv6. It points to the Forwarding Agent which is assumed to be in the user's home network. Thus, the protection result is the same as with Mobile IPv6.

The Non-Disclosure Method [74] is an extension of Mobile IPv4—which can be conceptually applied to Mobile IPv6, too—and shields the locator updates between the Mobile Node and its Home Agent against observation by third parties. This is achieved by so-called Security Agents working similar to cryptographic MIXes. Protection regarding the Home Agent and the communication partners as potential attackers is not considered. Thus, with respect to the protection goals and the attacker model of this thesis, the evaluation results are the same as with plain Mobile IPv6.

The Freiburg Location Addressing Scheme (FLASCHE) [236], [237] aims at protection against the link of several actions of a Mobile Node to each other and against the link of those actions and the user to the user's device. This is achieved by not using the same identifier over a considerable period of time. Instead, temporary identifiers are used. They consist of a random part as well as of a part containing the Mobile Node's current location. FLASCHE relies on the frequent change of the location, because thus, the identifier changes frequently.

This approach is only designed for communication initiated by the Mobile Node itself. Reply traffic can only be received as long as there are only minor location changes. For reachability, the system could be combined with Mobile IPv6, i.e., using FLASCHE between the Home Agent and the Correspondent Nodes. This in turn undermines the protection, because the identifier does no longer change. Thus, in the general scenario of communication being initiated by the Mobile Node and Correspondent Nodes, the same protection is achieved as with Mobile IPv6.

Hierarchical Systems

In this section, systems are evaluated, that consist of a hierarchy of entities which are similar to a Mobile IPv6 Home Agent. The lower the entities are placed in the hierarchy, the smaller is the area they are responsible for. The areas are smaller in terms of geographical extent as well as in terms of potentially contained users.

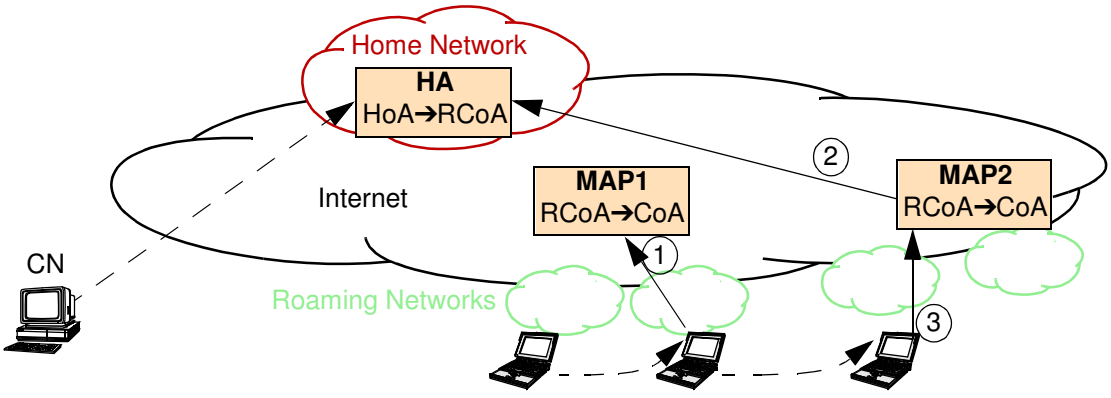


Figure 4.7: Hierarchical Systems

Figure 4.7 shows the scenario of Hierarchical Mobile IPv6 [207]. There is a *Correspondent Node, CN* sending packets to the user. On the top, there is the user's home network and below there are four roaming networks. All networks are interconnected by the Internet.

The highest level entity *HA* is similar to a *Home Agent*. Additionally to that, there may exist several levels of so-called *Mobility Anchor Points, MAPx*. The HA and the MAPx together

build up the mobility management system. A Mobility Anchor Point serves as a kind of a local Home Agent for the current roaming area of the Mobile Node. The Mobile Node does not only have a *home address, HoA* in the network of the Home Agent and a *care-of address, CoA* in the visited roaming network but also a so-called *regional care-of address, RCoA* in the network of the Mobility Anchor Point. The currently assigned address is called *LCoA, on-link care-of address*, and resembles a common care-of address in Mobile IPv6. Therefore, the term care-of address is used in order to avoid confusion.

On local movement, only the binding between the changing CoA and the RCoA has to be updated at the Mobility Anchor Point (1). Only if the area of a Mobility Anchor Point is left, the binding between the HoA and the new RCoA at the Home Agent must be updated (2) additionally to the update of the CoA (3). Thus, local movements are transparent to the Home Agent which reduces signalling overhead. This was the original intention of Hierarchical Mobile IPv6.

The Home Agent is fixed in the user's home network like in Mobile IPv6. Hence, the HoA contains sensitive information about the user's home, because of its implicit functionality as a locator of the fixed presence of the Mobile Node. The current Mobility Anchor Points at the different levels are determined due to the user's current location and change when the user moves. Thus, RCoAs and the CoA contain sensitive information about the user's current location.

Like the HoA, the addresses used between the agents, i.e., the RCoAs, serve as both, identifier and locator. For the sending entity of the higher level they are a locator, i.e., a pointer to the location where to send the packet. For the receiving entity on the lower level, they serve as an identifier, i.e., an indicator which Mobile Node is to be contacted.

Only the lowest level Mobility Anchor Point knows the current exact location of the Mobile Node. Entities at higher levels only see less accurate locations of the next lower Mobility Anchor Point's area.

The system described in [6] is conceptually similar to Hierarchical Mobile IPv6. Here, the hierarchically organized entities are called Mist Routers. They build up a tree-like overlay network. Between the Mist Routers, link identifiers and pseudonymous handles are used for path finding. No entity knows the path through all Mist routers used by a user. Only the top level Mist Router knows the user's identifier, while only the lowest level Mist Router knows the current exact locator. Higher level Mist Routers can derive sensitive information from knowledge of the next lower level Mist Router which is indicated by the outgoing network link. This is similar to an intermediate locator like the RCoAs used between the agents in Hierarchical Mobile IPv6.

Mixed Mobile IP [143] proposes an approach in which there are two MIX-like entities, *MIX 1* and *MIX 2* between the Home Agent and the Mobile Node. Figure 4.8 shows the scenario. The *Correspondent Node, CN* sends packets to the *Home Agent*. The Home Agent does not know the current locator but forwards the packets to *MIX 1*. MIX 1 in turn forwards the packet to *MIX 2*, which knows the locator and finally delivers the packet to the *Mobile Node, MN*. Originally, the approach is designed for Mobile IPv4 with a so-called Foreign Agent in the currently visited network. This Foreign Agent is the endpoint of the IP-in-IP tunnel from the Home Agent. This Foreign Agent knows the user's locator. In principle, the approach can be transferred to Mobile IPv6 by the last MIX knowing the user's locator.

This is the scenario discussed here. From a mere perspective of VID protection, [143] provides for the same level of protection like the other hierarchical systems do.

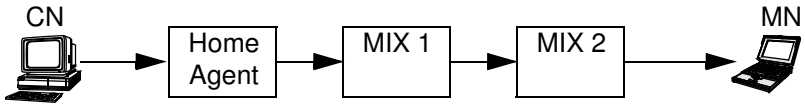


Figure 4.8: Mixed Mobile IP

To a certain degree, also [150], [2] and their base [166] are similar hierarchical approaches.

In the hierarchical systems evaluated in this section, protection regarding Correspondent Nodes as potential attackers is the same as with Mobile IPv6 used with a bidirectional tunnel. This is, because the path from the Correspondent Nodes to the top-level entity in the hierarchy is identical. Correspondent Nodes can see the identifier but not the current locator.

For evaluation of the threats regarding the mobility management system as potential attacker—which here consists of several entities—it is assumed that different hierarchical entities are operated by different parties. This assumption is necessary to distribute the sensitive information among several parties.

Nearly all protection goals are broken against any of the attackers of the attacker model in this thesis. The reason is, that those systems do not consider VIDs. The lowest entity in the hierarchy knows the location always and thus knows an infinite location trace and can infer personal attributes. The top-level entity in the hierarchy and the Correspondent Node know the home address and thus can infer the home provider and the home location. The Correspondent Node can know and link several VID-identifiers if used for different VIDs or if collaborating with another Correspondent Node. The only achieved protection goal is confidentiality of S2 as defined in section 3.2.2.2. There is no entity according to the attacker model of this thesis, which knows both, the identifier and the locator of the user's VIDs.

Systems based on the Internet Indirection Infrastructure

i3 [212] proposes a new Internet architecture which is based on the principle of indirection of communications. Thus, it is simple to include mobility, multicast and other services. Senders are sending packets to an identifier and the packets are routed to the server taking care for this identifier. Recipients are registering so-called triggers for identifiers, whose packets they want to receive. Identifiers do not have to be from the home network, thus allowing for protection of the protection goals confidentiality of F1 and of F2. Nevertheless, the server of the identifier of the Mobile Node still sees all care-of addresses. Thus, protection goals confidentiality of S1, F3, and F4 are violated. S2 and S3 also can be disclosed if a Correspondent Node collaborates with this server.

Secure-i3 [3] is an extension of i3 to provide better security and privacy. For privacy, it basically introduces a mandatory second indirection somehow equivalent to the step from one MIX to a cascade of MIXes. Protection is aimed towards third parties, and the entities of the communication network and the Correspondent Node can still attack the same protection goals like in i3.

Hi3 [169] is a merge of HIP and Secure-i3 without fundamental improvements regarding the protection of the VID approach.

VID-supporting Systems

Flying Freedom [68] extends the Freedom system [24] towards handling mobility. The Freedom system is an overlay network using MIX-like entities, so-called *Anonymous Internet Proxies, AIPs*, for pseudonymous communication. Between the AIP serving as Home Agent and the last AIP knowing the locator there may be a chain of AIPs of arbitrary length. For the discussion it is assumed that the AIP serving as Home Agent is located in the user's home network. The system shields the current locator against communication partners. The user has an IP address being part of the network in which the AIP acting as Home Agent is located.

The last AIP knowing the location is never changed and thus, protection goals confidentiality of S2, F3, and of F4 are broken. As the first AIP is assumed to be in the home network, also protection goals confidentiality of F1 and F2 are broken. Protection goal confidentiality of S2 is protected as no entity is knowing the location and the VID-identifier. Finally, protection goal confidentiality of S3 can also be met, if different first AIPs can be chosen for different VIDs. This depends on the real scenario of provided AIPs.

[149] aims at protection of VIDs in a mobile environment based on Mobile IPv6. It has a different attacker model in mind than this thesis. It assumes a trusted provider of the Home Agent. The protection is achieved by providing each VID with an own set of identifiers throughout all layers of the networking stack. Nevertheless, all home addresses are from the user's home network and are seeing the care-of addresses forever. Thus, protection goals confidentiality of F1-F4 and of S1 are violated. Moreover, the Home Agent knows the home address and the care-of address and thus can in collaboration with a Correspondent Node link a VID-identifier to the location thus violating protection goal confidentiality of S2. Finally, the home agent will be able to link all identifiers of all VIDs due to the always identical movements. Thus, in collaboration with Correspondent Nodes, even protection goal confidentiality of S3 can be broken.

For the sake of completeness, it must be mentioned, that there are also approaches in the literature relying on secret sharing for protection of the location, which are not from the area of communication networks but rather from the application domain, e.g., [148] for an application layer location server.

Chapter 5

Evaluation of Scenario-Independent Threats in the New Architecture

This chapter contains the first of two evaluations of the new architecture. Section 5.1 explains the evaluation goals and is followed by section 5.2, which describes the knowledge model of the new architecture. Section 5.3 then contains the actual evaluation, before section 5.4 closes the chapter with a summary of the evaluation result. Chapter 6 then will present the second evaluation, which is scenario-dependent. Scenario-dependent evaluation here means the evaluation of the behavior of the system dependent on the concrete behavior of the user, i.e., the movement, the communication, the number of VIDs, the number of servers, and a concrete server change algorithm.

5.1 Goals

After developing the architecture in chapter 4, its protection capabilities regarding the VID approach have to be evaluated. The evaluation in this thesis is twofold. This chapter applies the methodology proposed in chapter 3 to the new architecture and thus yields a scenario-independent privacy evaluation. Chapter 6 then will evaluate scenario-dependent privacy aspects.

The improvement methodology of chapter 4 requires the creative input by the privacy engineer. Thus, the resulting architecture is not proved by itself to fulfill all protection goals. This chapter evaluates the achieved protection. The results are independent of the scenario in which the architecture is used and gives absolute statements about which protection goals hold and which protection goals might be violated in which situations. The next section starts by describing the knowledge model of the new architecture.

5.2 Knowledge Model

At first, the protection goals and the assumptions for the evaluation of the new architecture are described in 5.2.1 and 5.2.2 respectively. Then the model, which is split into two views, the elementary fact type view and the dynamic view, is built. Section 5.2.3 starts with the first view.

5.2.1 Protection Goals for Evaluation of the New Architecture

The protection goals are the same as in chapter 3.2.2.2. This is necessary for a proper comparison of both systems. The goals are confidentiality protection of the following assets:

- Fact F1 Home provider
- Fact F2 Home location
- Fact F3 Real name
- Fact F4 Personal attributes
- Set S1 Large location trace
- Set S2 Location and VID-identifier
- Set S3 n VID-identifiers

F1 and F2 are principally no longer contained in the new architecture. The user still has fixed positions in the network—the permanent addresses and thus the fMAs—but the user has several of them and thus no single prominent provider as in a traditional GSM or Mobile IP environment. Moreover, the permanent address and thus the fMA address do not relate to the user's physical home in any way. Therefore, neither the home provider nor the home location can be inferred from those facts. There is no other possibility how F1 and F2 could be revealed in the new architecture. Therefore, also the model of the new architecture do not contain those fact types anymore, cf. 5.2.3 and 5.2.4. F1 and F2 are no longer followed in this chapter.

F3 and F4 here can only be inferred from S1. Facts of those fact types are not stored or processed by the system but can only be inferred from a large location trace. Thus, it is sufficient to evaluate the model with respect to S1, S2, and S3. The next step in the methodology is to specify the potential attackers which are to be considered.

5.2.2 Assumptions for Evaluation of the New Architecture

The assumptions are in general the same as in section 3.2.2.3. Some assumptions are translated from Mobile IPv6 to the new architecture. Those adapted assumptions are listed here.

The following assumptions originate from the considered system under evaluation as well as its configuration and usage:

- Each interface can have several MAC addresses
- Each interface can have several IP addresses—the permanent addresses and one care-of address.
- IPSec or a similar encryption protocol is used.

- The new architecture does not use an equivalent to the Mobile IPv6 route optimization.

The following assumptions originate from the considered attackers and define the attacker model:

- Every attacker knows the mapping to get the user's permanent address from the VID-identifiers.
- No attacker knows the mapping to get a VID-identifier from a permanent address.

The next section develops the elementary fact type view of the new architecture.

5.2.3 Elementary Fact Type View on Model of the New Architecture

Some simplifications and assumptions lay the ground for the model. First of all, the fact type "network" here reflects a template, which contains the network fact type itself, its location fact type and its provider fact type. Figure 5.1 shows this abstraction which is used for all networks in the model. Moreover, the device fact type and the interface fact type are neither directly nor indirectly sensitive and thus are neglected. This is the simplification announced in section 3.2.2.3.

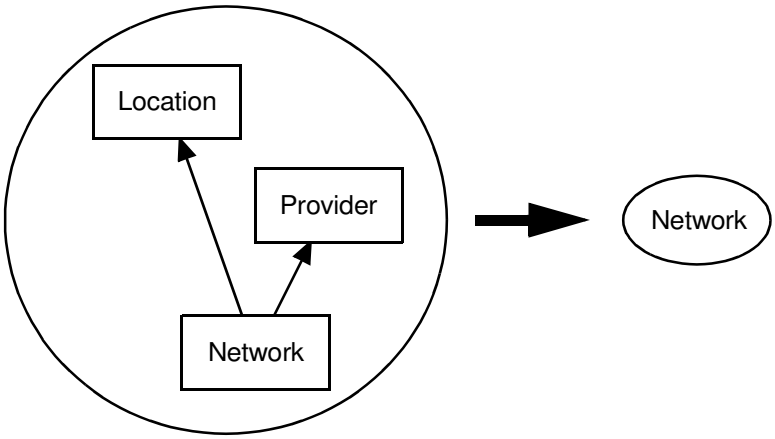


Figure 5.1: Abstracted network fact type

Figure 5.2 shows the model of the elementary fact type view, which is discussed in the following starting at the right hand side of the figure. The figure now contains both aspects of fact set links—links of fact sets of different VIDs and links of fact sets of the same VID, i.e., links of different location tracelets of one VID.

Fact Types and Relations

The fact types and the relations are explained from right to left. The VID-identifier fact type and the *Permanent Address, PA* fact type share a 1:1 relation, because each VID has its own permanent address. There is no MAC address fact type associated to the PA fact type, because the PA is never visible in the access network of the Mobile Node. The *fixed Mobility Agent, fMA* address and the PA share the same network. The network does not have any relation to the user. This also holds true for the *variable Mobility Agent, vMA* network and the *Shareholder, Shh* network.

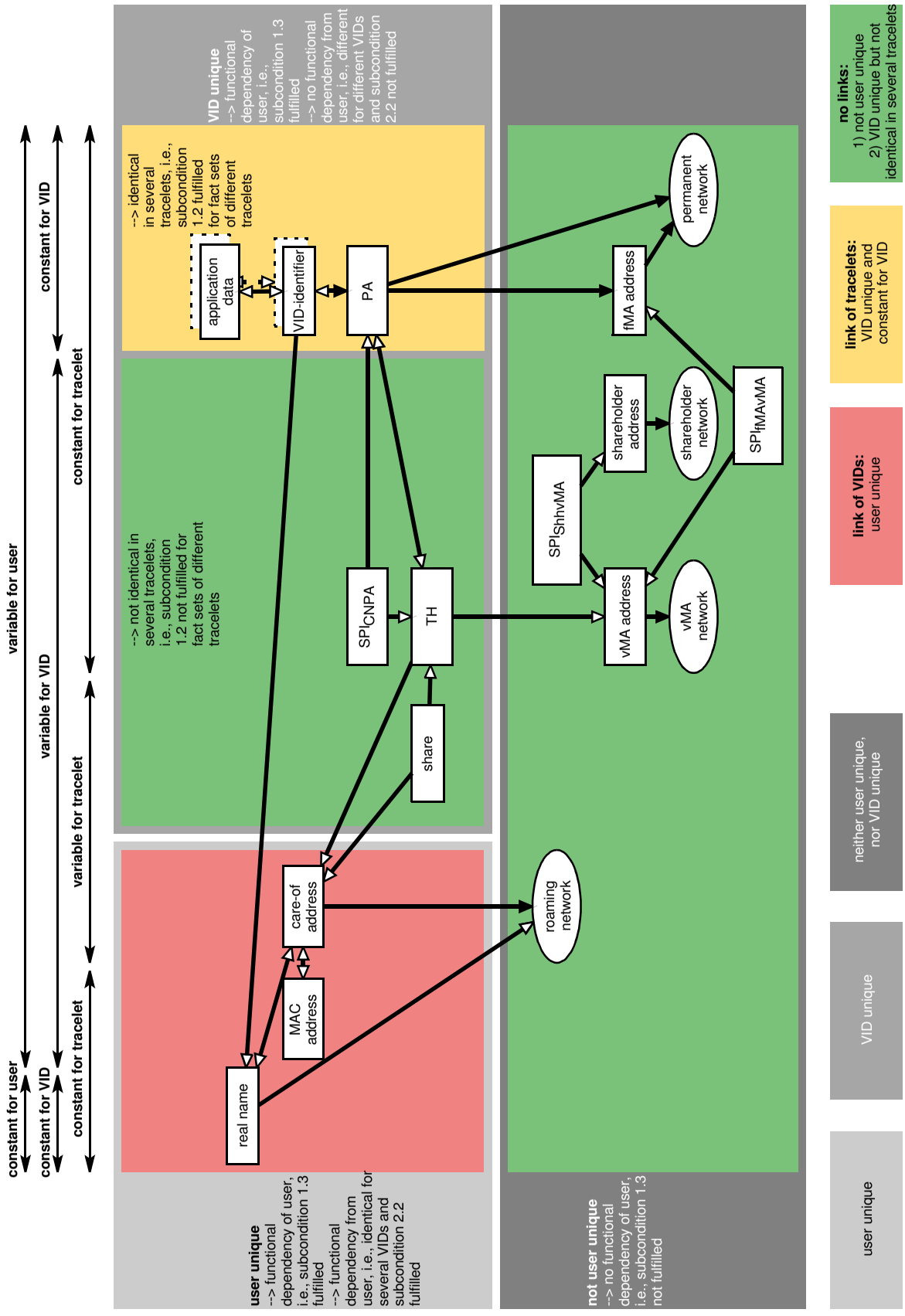


Figure 5.2: Elementary fact type view on model of new architecture

The *Security Parameter Index, SPI* fact types are separated according to the agents between which they are applied. Whereas this is not visible for eavesdroppers, the agents themselves as potential attackers will know these semantics. This difference between the attackers could result in a separate model for eavesdroppers, in which only one type of SPI would occur. For simplicity, the model is not split here but used for both classes of potential attackers. Thereby, the model reflects the stronger of both attacker classes. Each SPI fact type is uniquely related to the fact types of the two agents between which the SPI is applied. Each agent generally shares several SPIs with other agents.

The SPI_{CNPA} is related to the Correspondent Node. The Correspondent Node does not show up as a fact type of its own, because the respective fact type is considered as being personal information of the Correspondent Node and not of the Mobile Node whose privacy is under evaluation here. On the side of the Mobile Node, the SPI_{CNPA} is uniquely related to the *Temporary Handle, TH* in the model, because both are changing simultaneously. There is one TH for each VID at a time and each TH is changing over time. The SPI_{CNPA} fact type shares an N:1 relation to the PA fact type, because the same permanent address can be used with several Correspondent Nodes, simultaneously.

The user changes the TH simultaneously with the vMA and the SPI_{CNPA}. The same care-of address is assumed to possibly result in different shares. An identical share instead is assumed to be from an identical care-of address. It is considered unlikely that two different care-of addresses result in an identical share. With a changing secret sharing algorithm, this assumption may change.

THs are assumed to be unique for one VID at a certain point in time. The shares are different for each VID having its own TH and thus, an identical share means an identical TH and the share fact type is uniquely related to the TH fact type. On the other hand, simultaneous identity of the TH does not imply identity of the shares, because each TH refers to several shares.

The TH fact type maps uniquely to the care-of address fact type. The TH fact type shares a 1:1 relation with the permanent address fact type. Every vMA serves several VIDs at a time, thus its fact type maps ambiguously to the TH fact type.

The care-of address fact type shares a 1:1 relation with its MAC address fact type as well as with the real name fact type. The real name fact type shares an N:1 relation with the roaming network fact type. Each user is in exactly one network at a time according to the configuration, but this network serves several users simultaneously. Therefore, the roaming network fact type maps ambiguously to the real name fact type.

The only relations, whose mappings are known to potential attackers, are from the addresses to the respective networks and from the VID-identifier to the permanent address. These are the only solid arrowheads in the model.

Grey Boxes

Several areas indicated by grey boxes group the fact types. Fact types in the top left box, which is colored in light grey, are unique for one user. Fact types in the top right box, which is colored in middle grey, are unique for one VID. Finally, fact types in the bottom box, which is colored in dark grey, are neither unique for one user, nor unique for one VID.

The grey boxes are equivalent to the grey boxes in Figure 3.1. Each grey box subsumes fact types, which have equivalent properties from a VID linking perspective.

All relations between fact types of one grey box and fact types of another grey box are of equivalent cardinality. The relations into the dark grey box are all of nature 1:N. This relation reflects the anonymity set of users or VIDs respectively being possible as originators of facts of the fact types in the dark grey box. The relations between fact types of the middle grey box and fact types of the light grey box are of N:1 cardinality, because the user unique fact types in the light grey box are identical in the context of all VIDs, whose fact types are in the middle grey box.

It is possible to have an abstracted view on the model based only on the grey boxes. This view is indicating that a user as abstraction of the fact types in the light grey box shares a 1:N relation to VIDs as abstraction of the fact types in the middle grey box. Moreover, users as well as VIDs are sharing an N:1 relation to networks as abstraction of the fact types in the dark grey box. This reflects the setting in which one user has several VIDs. Both are connected to one network at a time. Each network serves several users and thus several VIDs at a time.

Colored Boxes

The next structural element is represented by the colored boxes. They reflect the vulnerabilities to the VID approach. All fact types in a box share the same vulnerabilities. There are three distinctions: Green, yellow, and red.

Facts of the fact types in the green box at the bottom can never be used to link sensitive fact sets. They will always build a kind of anonymity set for the users and VIDs. Thus, they are not fulfilling the linking subcondition 1.3.

Facts of the fact types of the middle grey box, i.e., of the top green box and of the yellow box, are fulfilling the linking subcondition 1.3. Nevertheless, they are not fulfilling the linking subcondition 2.2, i.e., they can be chosen differently for different VIDs and thus can only be used for linking fact sets of one VID.

Facts of the fact types in the top green box can only be used to link facts during one location tracelet, because they are changing when the tracelet changes, i.e., when the user moves the respective VID to a new vMA. Thus, there is never the identical fact in several tracelets and linking subcondition 1.2 is only fulfilled during the duration of one location tracelet. The aggregation of fact during one location tracelets is allowed, because this amount of information is rated as non-sensitive. Therefore, the color is green.

The difference of fact types in the yellow box to the fact types in the top green box is that the fact types in the yellow box are constant during the lifetime of a VID, i.e., during the lifetime of several location tracelets. Thus, facts of these fact types can be used to link several location tracelets of one VID.

Finally, attackers can use facts of the fact types in the red box to link several VIDs of a user. Those facts are unique for a user and are identical for all VIDs. The user can only have one instance of them at a time. Those fact types are functionally determining the user and vice versa. Thus, the fact types fulfill the linking subconditions 1.2 and 1.3 as well as the linking subcondition 2.2.

The figure shows two hierarchies of linking properties. The grey boxes are indicating properties regarding the link of fact sets of different VIDs, whereas the colored boxes are separating fact types according to their properties regarding the link of tracelets of one VID.

Arrows at Top of Figure

The next structural element is represented by the arrows at the very top of the figure. Those arrows are structuring the figure in columns. They are indicating, in which way—related to the VID or to the user—the fact types below the respective arrow are constant or variable, i.e., the fact types have only one or several possible instantiations.

Fact types under the very left arrow can only have one instantiation for the user, whereas all other fact types can result in different facts for the same user. Fact types under the very right arrow can only have one instantiation for a VID, whereas all other fact types can result in different facts for the same VID. Fact types under the middle right arrow can only have one instantiation during one location tracelet, whereas the other fact types can result in different facts during the duration of the same location tracelet, i.e., can result in different facts in the fact set of one location tracelet.

A location tracelet here has the meaning of a unit. It is the fact set, that an attacker can gain before the user moves the VID to a new vMA. The duration of the location tracelet thus is the serving time of one vMA.

5.2.4 Dynamic View on Model of the New Architecture

This section shows the dynamic view on the model. Again, there are simplifications. Besides the fact type representing the aggregated fact types around the network fact type from 5.2.3, there are more fact types, which are aggregated to represent the contained fact types. This grouping of fact types increases visibility of the relevant statements in the dynamic view. In fact, the tracelets and traces exist for each single fact type contained in such an aggregation of fact types. Figure 5.3 shows the aggregated fact types and the resulting symbols. The share fact type is not aggregated with the group around the TH, because shares are changing more rapidly.

Figure 5.4 shows the dynamic view on the model. Update fact types are omitted. They do not contain more relevant information than the originating fact types.

Fact Types and Relations

Figure 5.4 does not contain the colored boxes like Figure 5.2 does. The colors are used differently here. Constant fact types surrounding a variable fact type are of yellow color. They will be evaluated based on the elementary fact type view and are not relevant in the dynamic view. The same holds true for the elementary variable fact types in dark green. Tracelet fact types not being more sensitive than the respective elementary fact types are light green. Trace fact types are of red color, because they contain additional sensitive information.

The relations of the fact types in the group around the TH type to surrounding fact types are aggregated to a dotted line. They expand to the same relations as in the elementary fact type view and are identical in the elementary fact type and in the tracelet fact type. Also the rela-

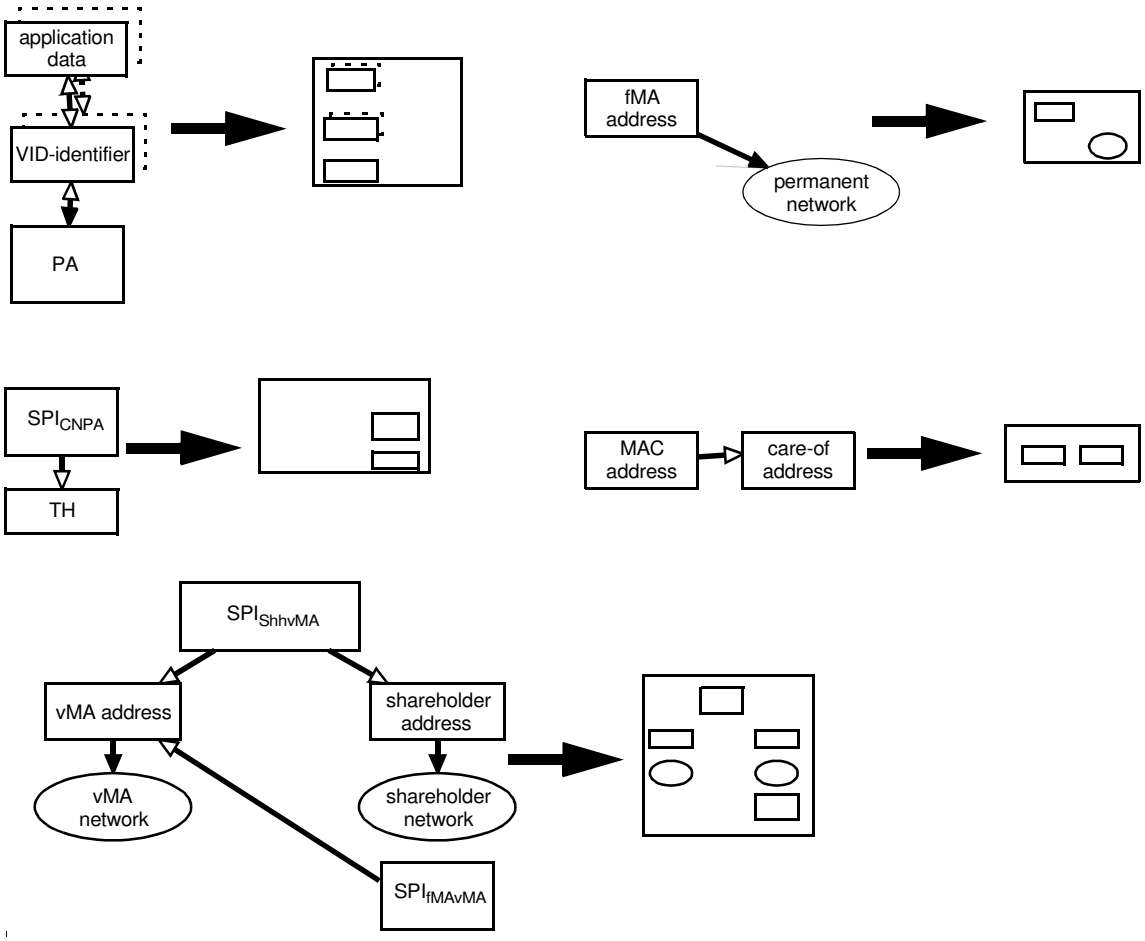


Figure 5.3: Simplification symbols of dynamic view

tions between variable fact types in a certain group remain the same if considered as elementary fact types, as tracelet fact types or as trace fact types.

The relation between the group around the TH fact type and the group around the vMA address fact type depends on the tracelet cardinality of the group around the TH fact type. It can either be a relation between both groups as tracelets or between the group around the TH type in the form of a tracelet and the group around the vMA address fact type in the form of a trace.

The trace of the group around the vMA address fact type becomes unique for a VID, when it is growing too long. Therefore, the trace fact type of this group is located in the middle grey box. It is marked in red, because it represents an additional vulnerability as compared to the elementary fact type view. The same holds true for the relation between this group trace fact type and the yellow surrounding fact types. The architecture prevents attackers from the observation of all other traces. Therefore, no other trace fact types exist in the model.

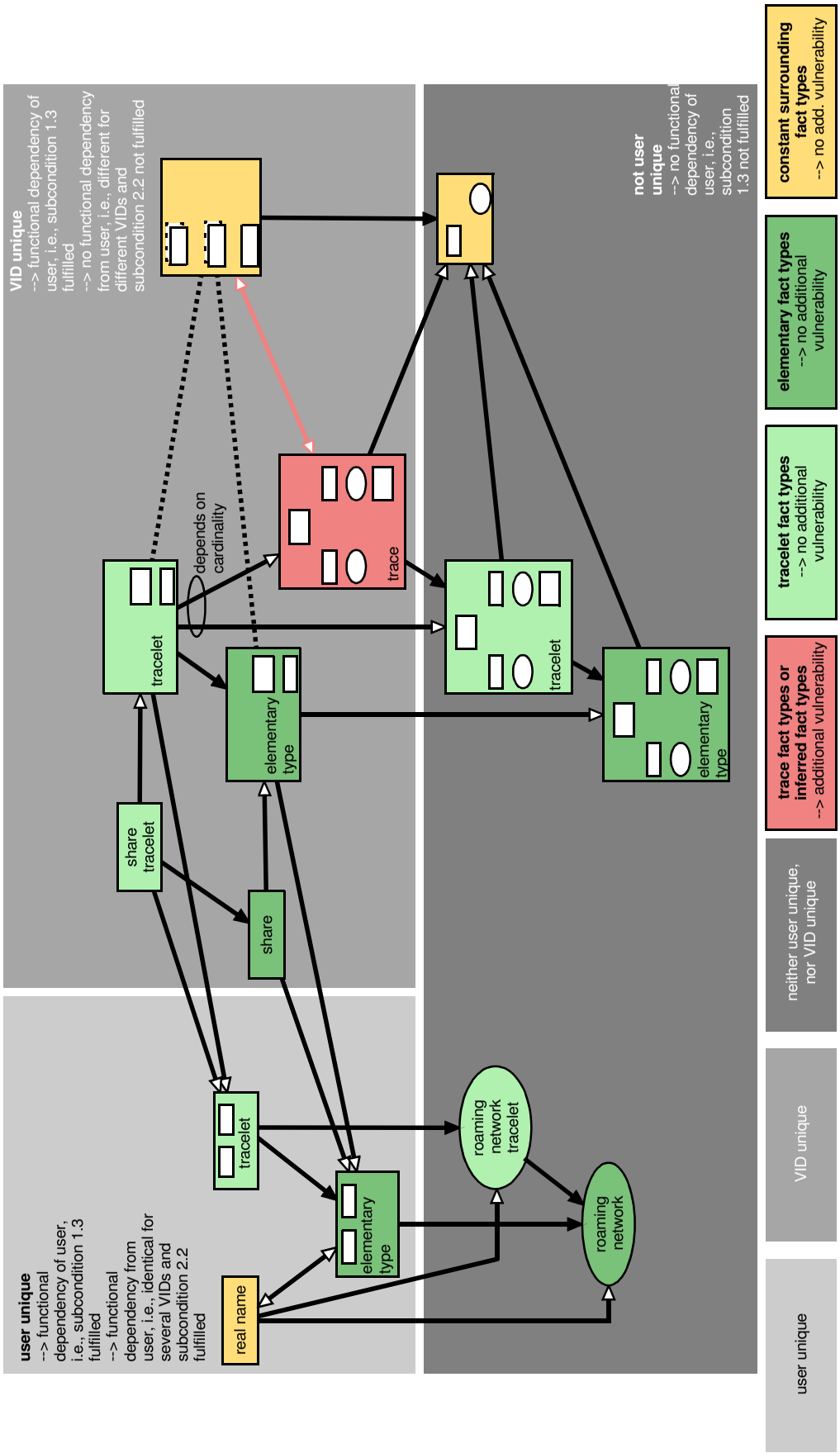


Figure 5.4: Dynamic view on model of new architecture

Grey Boxes

The grey boxes are equivalent to the grey boxes in the elementary fact type view. In the dynamic view, only vulnerabilities about linking VIDs are relevant. Evaluating the link of tracelets is not relevant here, because the facts of all variable fact types change at the end of a tracelet and constant fact types are not considered in the dynamic view. Therefore, the arrows from Figure 5.2 are not contained in Figure 5.4.

The next section shows the evaluation of the model. Therefore, both views are considered.

5.3 Evaluation

Like in chapter 3, at first preparatory information is collected in 5.3.1. Then, the final evaluation starts in section 5.3.2.

5.3.1 Preparations

The preparations start with a recapitulation of the potential attackers. Then the observations, which are possible, are collected in section 5.3.1.2. Finally, sections 5.3.1.3 and 5.3.1.4 show possibilities for inferring new facts and linking fact sets respectively.

5.3.1.1 Potential Attackers

The potential attackers are similar to the ones in chapter 3. They are an instantiation of the generic attacker model from section 5.2.2. The attackers are the communication partner, the agents of the communication system and eavesdroppers in between. All attackers are assumed to be capable of doing the same interpretations, once a certain fact or fact set is known. In detail, the following attackers are considered:

- Correspondent Node
- fMA
- vMA
- Shareholders
- $\text{Eavesdropper}_{\text{CNfMA}}$
- $\text{Eavesdropper}_{\text{fMAvMA}}$
- $\text{Eavesdropper}_{\text{ShhvMA}}$
- $\text{Eavesdropper}_{\text{MNShh}}$
- $\text{Eavesdropper}_{\text{MNvMA}}$
- $\text{Eavesdropper}_{\text{MNfMA}}$
- $\text{Eavesdropper}_{\text{MNCN}}$
- $\text{Eavesdropper}_{\text{LinkMN}}$

Each potential attacker can observe a number of facts. The following section collects these possible observations.

5.3.1.2 Observations

Each agent of the communication system is assumed to know its network as well as the provider and location of this network. This is grouped to the observations, here in the sense that it is not subject to an interpretation, but that it is directly known. Table 5.1 subsumes the observations. Thereby the $\text{Eavesdropper}_{\text{MNCN}}$ is listening to packets flowing from the Mobile Node to the Correspondent Node and not in the other direction.

Generally, the attackers observe data packets. Often, there is one fact, e.g., the TH, which is contained in each packet and which allows for the link of all packets to the same VID. Then, the attacker can combine all observations. There is also the possibility that an attacker can observe different kinds of packets and those packets do not contain a VID unique or user unique fact allowing to link the packets. Then, the observations of such an attacker are partitioned. Table 5.1 separates the partitions by grey bars. This partitioning of the observations is only necessary for the $\text{Eavesdropper}_{\text{MNvMA}}$ and for the $\text{Eavesdropper}_{\text{LinkMN}}$. This is detailed in the following.

The $\text{Eavesdropper}_{\text{MNvMA}}$ can observe two kinds of packets. The first one is flowing from the vMA to the MN. These packets contain the address of the vMA, the care-of address of the Mobile Node as well as the SPI_{CNPA}. Packets from the Mobile Node destined to the vMA contain the address of the vMA and the anonymous sender address of the Mobile Node.

The $\text{Eavesdropper}_{\text{LinkMN}}$, can observe five different kinds of packets. Packets from the Mobile Node to the fMA, to the Shareholders, and to the vMAs contain the address of the respective agent as well as an anonymous IP address of the Mobile Node and an anonymous MAC address of the Mobile Node. Packets from the Mobile Node to the Correspondent Node contain the IP address of the Correspondent Node, an anonymous IP address of the Mobile Node, an anonymous MAC address of the Mobile Node, and the SPI_{CNPA}. Finally, packets from the vMA to the Mobile Node contain the address of the vMA, the care-of address of the Mobile Node as well as the currently used MAC address of the Mobile Node.

The next section shows, which new facts attackers can infer from their observations.

5.3.1.3 Inference of New Facts

Table 5.2 shows the possibilities of inferring new facts from already known ones. All attackers are capable of doing the same inferences. Each network allows for inference of the corresponding location and the provider. Moreover, each trace allows for inference of the respective tracelet and each tracelet allows for inference of the respective elementary fact.

5.3.1.4 Linking of Fact Sets

Table 5.3 shows the fact types, whose facts can serve attackers to link fact sets. These are all the fact types, that are not in the dark grey boxes of the views on the model in Figure 5.2 and Figure 5.4. The table distinguishes between user unique fact types and VID unique fact types. Facts of user unique types serve for linking fact sets of different VIDs. Facts of VID unique fact types serve only for linking facts and fact sets of one of the user's VIDs. The user unique fact types are in the light grey boxes, whereas the VID unique fact types are in the middle grey boxes.

Attacker	Observation
Correspondent Node	VID-identifier permanent address SPI_{CNPA}
fMA	permanent address fMA address permanent network permanent network location permanent network provider vMA address TH SPI_{fMAvMA} SPI_{CNPA}
vMA	care-of address shares TH vMA address vMA network vMA network location vMA network provider Shareholder addresses of the resp. VID fMA address SPI_{fMAvMA} SPI_{ShhvMA} SPI_{CNPA}
Shareholders	share TH Shareholder address Shareholder network Shareholder network location Shareholder network provider vMA address SPI_{ShhvMA}
$Eavesdropper_{CNfMA}$	permanent address SPI_{CNPA}
$Eavesdropper_{fMAvMA}$	fMA address vMA address SPI_{fMAvMA}
$Eavesdropper_{ShhvMA}$	Shareholder address vMA address SPI_{ShhvMA}

Table 5.1: Observation possibilities

Attacker	Observation
$Eavesdropper_{MNShh}$	Shareholder address anonymous IP address
$Eavesdropper_{MNvMA}$	vMA address SPI_{CNPA} care-of address
	vMA address anonymous IP address
$Eavesdropper_{MNfMA}$	fMA address anonymous IP address
$Eavesdropper_{MNCN}$	IP address of CN SPI_{CNPA} anonymous IP address
$Eavesdropper_{LinkMN}$	fMA address anonymous MAC address anonymous IP address
	all Shareholder addresses anonymous MAC addresses anonymous IP addresses
	all vMA addresses anonymous MAC addresses anonymous IP addresses
	all vMA addresses care-of address MAC address of care-of address
	IP address of CN SPI_{CNPA} anonymous MAC address anonymous IP address

Table 5.1: Observation possibilities

The real name fact type would also be a user unique fact type, which attackers could use for linking fact sets of different VIDs. Nevertheless, it is not included in the table, because it cannot be observed by any potential attacker. Moreover, it cannot be inferred from any potential attacker like section 5.3.1.3 showed. The application data fact type is also not

Known Fact Type	Inferred Fact Types
VID-identifier	permanent address
permanent address	fMA address permanent network
permanent network[a]	provider of permanent network location of permanent network
Shareholder address	Shareholder network
vMA address	vMA network
care-of address	roaming network
trace[b]	tracelet
tracelet[c]	elementary fact type

Table 5.2: Inference possibilities

[a] similarly for Shareholder network, vMA network, and roaming network
[b] for all traces, tracelets
[c] for all tracelets, elementary types

Link Candidate Types	User Unique	VID Unique
VID-identifier		X
permanent address		X
TH[a]		X
Share[a]		X
SPI_{CNPA}[a]		X
care-of address[a]	X	
MAC address of CoA[a]	X	
trace of vMA address		X
trace of vMA network		X
trace of Shareholder address		X
trace of Shareholder network		X
trace of SPI_{ShhvMA}		X
trace of SPI_{fMAvMA}		X

Table 5.3: Link candidate types

[a] similarly for tracelet

included here, because it is out of the scope of the evaluation. It was only included for showing the link to models of application layer systems.

5.3.2 Evaluation

The evaluation considers single attackers and homogeneous attacker groups consisting of several identical attackers in 5.3.2.1. After that, the evaluation also covers heterogeneous attacker groups consisting of different attackers in 5.3.2.2.

5.3.2.1 Single Attackers and Homogeneous Attacker Groups

This section contains a table for each potential attacker according to 5.2.1. The table contains the fact types whose facts the respective attacker can observe or infer. Moreover, it informs, whether the attacker can only see elementary facts or also the respective tracelets and traces. Finally, there is a column indicating, whether the respective fact is user unique or VID unique. The latter states, whether the attacker can use the fact for linking fact sets, e.g., location tracelets, of one VID—in case of a VID unique fact type—or even fact sets of different VIDs—in case of a user unique fact type. If an attacker can observe and infer facts of the same type, this fact type is only shown as being observed in order to avoid duplications.

The tables containing the attacker's known fact types could be built by a simple tool, e.g., based on a spreadsheet. It is a partial view on the overall observation possibilities of Table 5.1 and an extension by fact types whose facts can be inferred according to Table 5.2. The last column is a mark of every fact type according to Table 5.3. Thus, the evaluator would only have to pick those fact types, whose facts the attacker can observe. Then the tool then could create the table automatically.

This section additionally contains one table per potential attacker showing, whether the potential attacker can violate any of the protection goals. For this evaluation, it is sufficient to consider the fact sets of the protection goals as was stated there.

The subsections for the attackers contain the information, which facts of the revealed fact types can be linked to fact sets. Each subsection is closed by a discussion regarding a homogeneous attacker group consisting of the respective kind of attacker. Sometimes, the user can gain different levels of protection according to the chosen configuration of the system, i.e., according to different compromises between performance and privacy protection. This is also mentioned in the discussion, when applicable.

Correspondent Node

Table 5.4 shows facts of which fact types the Correspondent Node can gain. All observed facts are unique for the VID. Only the inferred facts about the permanent network and its fMA are not VID unique. Thus, all known facts can be grouped by the PA or by the $SPI_{C\text{-}NPA}$, which are both contained in each packet and which are also both unique for the VID.

Table 5.5 shows that a Correspondent Node cannot violate any protection goal. The Correspondent Node does not see any of the user's roaming locations, which protects S1 and S2. If the Mobile Node used one Correspondent Node with several VIDs, the respective $SPI_{C\text{-}NPA}$ could also not be linked, because the Correspondent Node does not know facts any user unique fact type in the context of those VIDs.

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
VID-identifier		-	VID unique
permanent address		-	VID unique
	fMA address	-	-
	permanent network	-	-
	permanent network location	-	-
	permanent network provider	-	-
SPI_{CNPA}	-	tracelet	VID unique

Table 5.4: Known fact types of Correspondent Node

Asset	Protection	Reasoning
S1: Large location trace	+	no location revealed
S2: Location and VID-identifier	+	location not known
S3: n VID-identifiers	+	no user unique fact type to link different VID-identifiers

Table 5.5: Protection regarding Correspondent Node

Attacker groups consisting only of Correspondent Nodes cannot reveal more information. They still cannot gain any location, which protects S1 and S2. They still do not know facts of any user unique fact type, which protects S3.

fMA

Table 5.6 shows the fact types, whose facts an fMA can gain. The fMA can link all packets concerning a given Mobile Node by either the contained permanent address or by the contained TH, which are both unique and constant for the VID. Thus, the fMA can build tracelets of facts of each variable fact type. Linked instances of the vMA address, the derived facts, as well as the SPI_{fMAvMA} will become VID unique from a certain fact set size on. Thus, traces evolve. The TH and the SPI_{CNPA} instead are already VID unique as elementary facts and thus, even large tracelets will not become more sensitive. Therefore, no traces evolve.

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
permanent address		-	VID unique
fMA address		-	-

Table 5.6: Known fact types of fMA

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
permanent network		-	-
permanent network location		-	-
permanent network provider		-	-
vMA address		trace	VID unique
		tracelet	-
	vMA network	trace	VID unique
		tracelet	-
	vMA network location	trace	VID unique
		tracelet	-
	vMA network provider	trace	VID unique
		tracelet	-
SPI_{fMAvMA}	-	trace	VID unique
		tracelet	-
TH	-	tracelet	VID unique
SPI_{CNPA}	-	tracelet	VID unique

Table 5.6: Known fact types of fMA

Table 5.7 shows that an fMA cannot violate any of the protection goals. The fMA can neither see any roaming location of the Mobile Node, nor any VID-identifier.

Asset	Protection	Reasoning
S1: Large location trace	+	no location revealed
S2: Location and VID-identifier	+	neither VID-identifier nor location revealed
S3: n VID-identifiers	+	no VID-identifiers revealed

Table 5.7: Protection regarding fMA

Therefore, even a collaboration of a number of fMAs cannot violate any of the protection goals. They still would not know any location or any VID-identifier. Because no fact of a user unique fact type is revealed to any fMA, the collaborating fMAs could not even link several permanent addresses of a user.

vMA

Table 5.8 shows the fact types, whose facts a vMA can gain. Each packet sent to the vMA contains the TH, which is VID unique and constant during one location tracelet. Thus, a vMA can group all facts of all packets during the duration of one location tracelet. After the

duration of the tracelet the user picks another vMA. If the user picks the same vMA at time later, the TH will be different and thus the packets' content cannot be linked to the facts of the first tracelet. The vMA can build tracelets of the care-of address and inferable facts as well as from the shares. The duration of one location tracelet is used as time constant, here. This is determined by the user's algorithm of changing the agents.

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
vMA address	-	-	-
vMA network	-	-	-
vMA network location	-	-	-
vMA network provider	-	-	-
fMA address		-	-
	permanent network	-	-
	permanent network location	-	-
	permanent network provider	-	-
SPI_{fMAvMA}	-	-	-
Shareholder addresses		-	-
	Shareholder networks	-	-
	Shareholder network locations	-	-
	Shareholder network providers	-	-
SPI_{ShhvMA}[a]	-	-	-
TH	-	-	VID unique
SPI_{CNPA}	-	-	VID unique
shares	-	tracelet	VID unique
care-of address		tracelet	user unique
	roaming network	tracelet	-
	roaming network location	tracelet	-
	roaming network provider	tracelet	-

Table 5.8: Known fact types of vMA

[a] For all Shareholders

Table 5.9 shows, that a vMA cannot violate any protection goal. The vMA can link several location facts, but only during the duration of one tracelet. Revelation of location tracelets is allowed here. Several tracelets cannot be linked because the vMA does not know a fact of any linking candidate, which remains constant beyond the duration of one location tracelet. Further, the vMA does not know any VID-identifier, which protects S2 and S3.

Asset	Protection	Reasoning
S1: Large location trace	+	Only location tracelet is revealed. No linking candidate, which is constant over several tracelets is known.
S2: Location and VID-identifier	+	neither VID-identifier nor location revealed
S3: n VID-identifiers	+	no VID-identifiers revealed

Table 5.9: Protection regarding vMA

Shareholder

Table 5.10 shows facts of which fact types a Shareholder can gain. Each packet to or from a Shareholder contains the TH. By this, all revealed facts can be linked. This can only be done during the duration of one location tracelet, because afterwards the TH is changed.

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
Shareholder address	-	-	-
Shareholder network	-	-	-
Shareholder network location	-	-	-
Shareholder network provider	-	-	-
vMA address		tracelet	-
	vMA network	tracelet	-
	vMA network location	tracelet	-
	vMA network provider	tracelet	-
SPI_{ShhvMA}	-	tracelet	-
TH	-	tracelet	VID unique
share	-	tracelet	VID unique

Table 5.10: Known fact types of Shareholder

Table 5.11 shows that a Shareholder cannot violate any protection goal. Neither any roaming location of the user, nor any VID-identifier is revealed.

Asset	Protection	Reasoning
S1: Large location trace	+	no location revealed
S2: Location and VID-identifier	+	neither VID-identifier nor location revealed
S3: n VID-identifiers	+	no VID-identifiers revealed

Table 5.11: Protection regarding Shareholder

A homogeneous group of collaborating Shareholders can cooperate and merge their knowledge based on the identical TH, if the Shareholders are all serving a same VID. If all Shareholders of this respective VID are collaborating, the care-of address can be recombined. This can only be done for the care-of addresses during the duration of one location tracelet. This is allowed here. After the location tracelet, the TH is changing and at a certain point in time the user will pick new Shareholders for the VID, too. Therefore, the Shareholders cannot attack one user forever. The user has a certain control here for tuning the compromise between performance and privacy.

Because the collaborating Shareholders must all serve a same VID, the formation of such a group of collaborating Shareholders is not rated as being very likely. Depending on the secret sharing or secret splitting scheme, it might even be that a collaboration of the vMA is necessary to recombine the care-of address.

$\mathbf{Eavesdropper_{CNfMA}}$, $\mathbf{Eavesdropper_{fMAvMA}}$, $\mathbf{Eavesdropper_{ShhvMA}}$, $\mathbf{Eavesdropper_{MNShh}}$, $\mathbf{Eavesdropper_{MNfMA}}$, $\mathbf{Eavesdropper_{MNCN}}$

The attackers of this section are discussed jointly in one section, because they differ only in the facts they can observe. The evaluation result is the same for those attackers. Therefore, at first the observations for each potential attacker are described and then, the result is described and discussed at the end of the section.

Table 5.12 shows the fact types whose facts an $Eavesdropper_{CNfMA}$ can gain. All observable packets contain the permanent address and thus, all the revealed facts can be merged.

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
permanent address		-	VID unique
	fMA address	-	-
	permanent network	-	-
	permanent network location	-	-
	permanent network provider	-	-
SPI_{CNPA}	-	tracelet	VID unique

Table 5.12: Known fact types of $Eavesdropper_{CNfMA}$

Table 5.13 shows the fact types whose facts an $Eavesdropper_{fMAvMA}$ can gain. No fact allows for linking facts of several packets. Each packet could stem from a different VID from the perspective of an $Eavesdropper_{fMAvMA}$.

Table 5.14 shows the fact types whose facts an $Eavesdropper_{ShhvMA}$ can gain. No fact allows for linking facts of several packets. Each packet could stem from a different VID from the perspective of an $Eavesdropper_{ShhvMA}$.

Table 5.15 shows the fact types, whose facts an $Eavesdropper_{MNShh}$ can gain. No fact allows for linking facts of several packets. Each packet could stem from a different VID from the perspective of an $Eavesdropper_{MNShh}$.

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
fMA address		-	-
	permanent network	-	-
	permanent network location	-	-
	permanent network provider	-	-
vMA address		-	-
	vMA network	-	-
	vMA network location	-	-
	vMA network provider	-	-
SPI_{fMAvMA}	-	-	-

Table 5.13: Known fact types of $Eavesdropper_{fMAvMA}$

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
Shareholder address		-	-
	Shareholder network	-	-
	Shareholder network location	-	-
	Shareholder network provider	-	-
vMA address		-	-
	vMA network	-	-
	vMA network location	-	-
	vMA network provider	-	-
SPI_{ShhvMA}	-	-	-

Table 5.14: Known fact types of $Eavesdropper_{ShhvMA}$

Table 5.16 shows the fact types whose facts an $Eavesdropper_{MNfMA}$ can gain. No fact allows for linking facts of several packets. Each packet could stem from a different VID from the perspective of an $Eavesdropper_{MNfMA}$.

Table 5.17 shows the fact types, whose facts an $Eavesdropper_{MNCN}$ can gain. The facts from all packets can be linked by the SPI_{CNPA} during one location tracelet. It is assumed that a Correspondent Node communicates with more than one user. Thus, the address of the Correspondent Node does not allow for linking packets.

Table 5.18 shows that none of the eavesdroppers discussed in this section can violate any protection goal. None of them does either see any roaming location of the Mobile Node, or any VID-identifier.

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
Shareholder address		-	-
	Shareholder network	-	-
	Shareholder network location	-	-
	Shareholder network provider	-	-
anonymous IP address	-	-	-

Table 5.15: Known fact types of $\text{Eavesdropper}_{\text{MNShh}}$

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
fMA address		-	-
	fMA network	-	-
	fMA network location	-	-
	fMA network provider	-	-
anonymous IP address	-	-	-

Table 5.16: Known fact types of $\text{Eavesdropper}_{\text{MNfMA}}$

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
IP address of CN	_[a]	_[a]	_[a]
SPI_{CNPA}	-	-	VID unique
anonymous IP address	-	-	-

Table 5.17: Known fact types of $\text{Eavesdropper}_{\text{MNCN}}$

[a] depends on anonymity protection of Correspondent Node

Asset	Protection	Reasoning
S1: Large location trace	+	no location revealed
S2: Location and VID-identifier	+	neither VID-identifier nor location revealed
S3: n VID-identifiers	+	no VID-identifiers revealed

Table 5.18: Protection regarding eavesdroppers of this section

Thus, even homogeneous groups consisting of collaborating eavesdroppers of the kinds discussed in this section cannot violate any protection goal. They still do not know any roaming location or any VID-identifier.

Eavesdropper$_{MNvMA}$

Table 5.19 shows facts of which fact types an Eavesdropper$_{MNvMA}$ can gain. There are two different kinds of packets and no contained fact allows for a link of the content of both kinds of packets. Thus the table shows two partitions. The first partition reflects packets being sent from the vMA to the Mobile Node. Such packets could be linked during one location tracelet based on the identical SPI$_{CNPA}$. After that, the SPI$_{CNPA}$ changes. The other partition reflects packets being sent from the Mobile Node to the vMA. They cannot be linked to each other or to packets from the other partition.

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
vMA address		-	-
	vMA network	-	-
	vMA network location	-	-
	vMA network provider	-	-
SPI$_{CNPA}$	-	-	VID unique
care-of address		-	user unique
	roaming network	-	-
	roaming network location	-	-
	roaming network provider	-	-
vMA address		-	-
	vMA network	-	-
	vMA network location	-	-
	vMA network provider	-	-
anonymous IP address	-	-	-

Table 5.19: Known fact types of Eavesdropper$_{MNvMA}$

Table 5.20 shows that an Eavesdropper$_{MNvMA}$ cannot violate any protection goal. Regarding the care-of address, it is assumed, that when the Mobile Node moves to a new roaming network, the packets no longer pass the Eavesdropper$_{MNvMA}$. Thus, it cannot observe a tracelet of care-of addresses. Even if the Eavesdropper$_{MNvMA}$ is located directly at the vMA, which is the worst case, at maximum a tracelet of care-of addresses and respective inferable facts can be gained. After the duration of one location tracelet, the vMA changes, which will definitely put the eavesdropper out of play. The Eavesdropper$_{MNvMA}$ cannot see any VID-identifier, which protects S2 and S3.

A homogeneous group of collaborating eavesdroppers between the Mobile Node and the vMA cannot increase the disclosed tracelet of care-of addresses. After the duration of one tracelet, the SPI$_{CNPA}$ as well as the vMA and the roaming network will change and thus the

Asset	Protection	Reasoning
S1: Large location trace	+	$Eavesdropper_{MNvMA}$ is only between vMA and one or a few roaming networks. Therefore only one location or a short tracelet will be revealed. Several locations could be linked by identical SPI_{CNPA} during the maximum duration of one location tracelet.
S2: Location and VID-identifier	+	neither VID-identifier nor location revealed
S3: n VID-identifiers	+	no VID-identifiers revealed

Table 5.20: Protection regarding $Eavesdropper_{MNvMA}$

collaborating attackers do not know, whether subsequent packets belong to the same VID or to different VIDs of other users. Moreover, they still do not see any VID-identifier and thus cannot violate S2 and S3.

$\mathbf{Eavesdropper_{LinkMN}}$

Table 5.21 shows facts of which fact types an $Eavesdropper_{LinkMN}$ can gain. There are five different kinds of packets and therefore five partitions. The first partition reflects packets being sent from the Mobile Node to the fMA. Those packets do not contain any link candidate and thus, can neither be linked to each other nor to any other packet. The same holds true for packets from the Mobile Node to the Shareholders, which is reflected by the second partition and for packets from the Mobile Node to the vMA, which is reflected by the third partition.

The fourth partition reflects packets sent from the vMA to the Mobile Node. Such packets can be linked to each other during the duration of one location tracelet, because the SPI_{CNPA} remains constant during that time. Such packets can also be linked to packets being sent from the Mobile Node to the Correspondent Node, which is reflected by the fifth and last partition. In each partition the $Eavesdropper_{MNvMA}$ knows the roaming network and inferable fact types because the eavesdropper is located there.

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
fMA address		-	-
	fMA network	-	-
	fMA network location	-	-
	fMA network provider	-	-
anonymous MAC address	-	-	-
anonymous IP address	-	-	-

Table 5.21: Known fact types of $Eavesdropper_{LinkMN}$

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
roaming network	-	-	-
roaming network location	-	-	-
roaming network provider	-	-	-
Shareholder addresses		-	-
	Shareholder networks	-	-
	Shareholder network locations	-	-
	Shareholder network providers	-	-
anonymous MAC address	-	-	-
anonymous IP address	-	-	-
roaming network	-	-	-
roaming network location	-	-	-
roaming network provider	-	-	-
vMA addresses		-	-
	vMA networks	-	-
	vMA network locations	-	-
	vMA network providers	-	-
anonymous MAC address	-	-	-
anonymous IP address	-	-	-
roaming network	-	-	-
roaming network location	-	-	-
roaming network provider	-	-	-
vMA addresses		-	-
	vMA networks	-	-
	vMA network locations	-	-
	vMA network providers	-	-
care-of address	-	-	user unique
MAC address of CoA	-	-	user unique

Table 5.21: Known fact types of $\text{Eavesdropper}_{\text{LinkMN}}$

Observation	Inference	Trace/ Tracelet	User Unique or VID Unique
SPI_{CNPA}	-	-	VID unique
roaming network	-	-	-
roaming network location	-	-	-
roaming network provider	-	-	-
IP address of CN[a]	-	-	CN unique
SPI_{CNPA}	-	-	VID unique
anonymous MAC address	-	-	-
anonymous IP address	-	-	-

Table 5.21: Known fact types of $Eavesdropper_{LinkMN}$

[a] not contained in model because no estimation about privacy system of Correspondent Node

Table 5.22 shows that an $Eavesdropper_{LinkMN}$ cannot violate any of the protection goals. The linked packets of each of the partitions four and five only contain identical information and facts of the fact types from partition four and five together do not contain sensitive information. The attacker is located in the roaming network. As soon as the Mobile Node is moving, the attacker cannot observe packets of this Mobile Node any longer. Thus, it can only observe one single care-of address and infer the corresponding roaming location, which it knows anyway. Thus, S1 remains protected. S2 and S3 also remain protected, because the attacker does not know any VID-identifier.

Asset	Protection	Reasoning
S1: Large location trace	+	Attacker is only present in one network. Thus, only one care-of address is visible.
S2: Location and VID-identifier	+	no VID-identifier revealed
S3: n VID-identifiers	+	no VID-identifiers revealed

Table 5.22: Protection regarding $Eavesdropper_{LinkMN}$

Collaboration with other $Eavesdroppers_{LinkMN}$ can be possible as long as the SPI_{CNPA} does not change, i.e., during one location tracelet. By this, locations of one location tracelet can be linked. This is allowed here. For this attack, one attacker in each visited network is necessary. Such a group of attackers will still not know any VID-identifier. Thus, S2 and S3 remain protected, too.

5.3.2.2 Heterogeneous Attacker Groups

For evaluating threats by heterogeneous attacker groups, a linking diagram for the new architecture has to be drawn. This is contained in Figure 5.5. For readability purposes, only the respective address fact types are shown. They include the respective network, provider

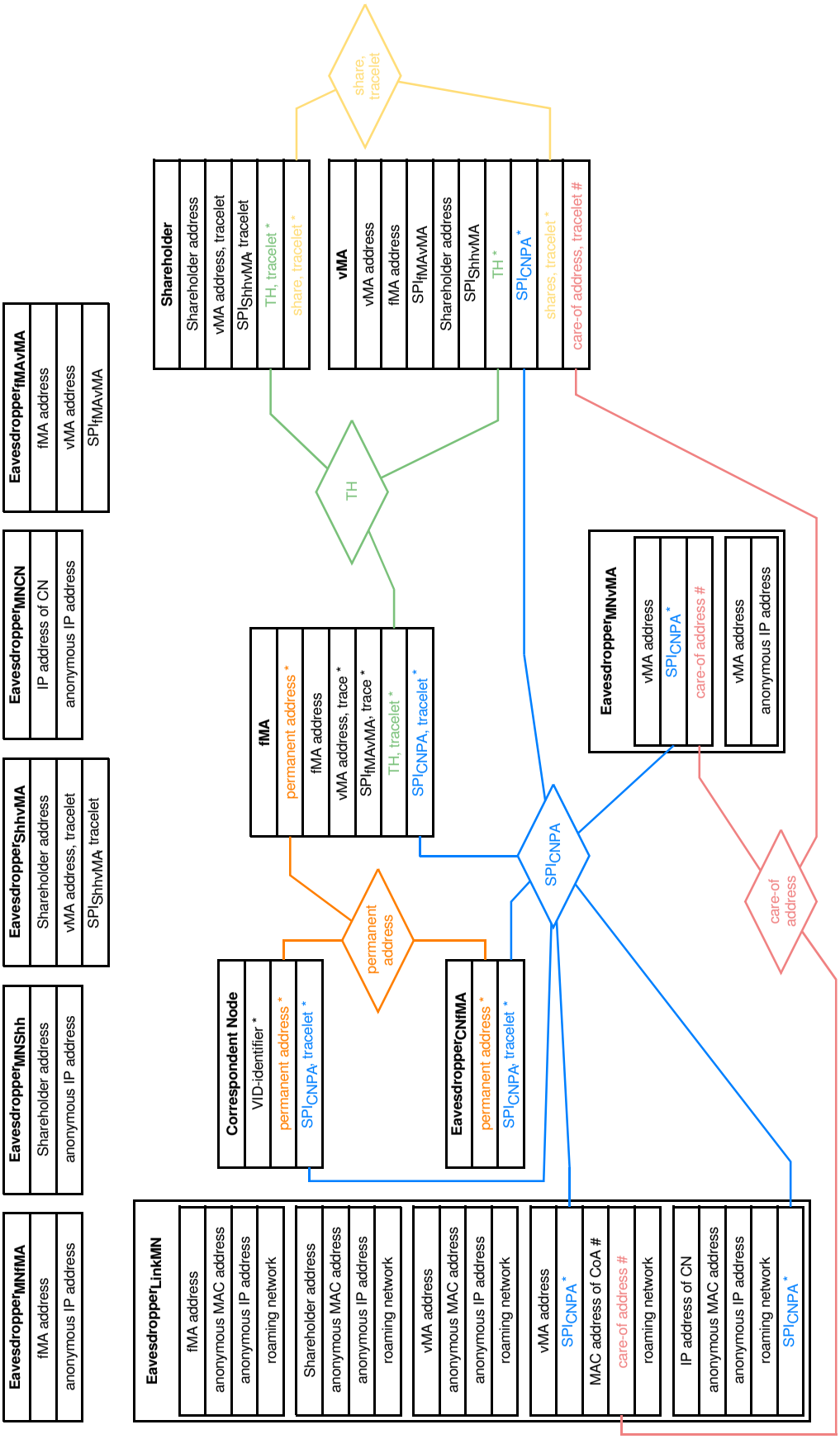


Figure 5.5: Linking diagram for new architecture

and location fact types. VID unique fact types are indicated by a "*". User unique fact types are indicated by a "#". If an attacker can build a trace or a tracelet, the respective worst case is depicted in the diagram.

For collaboration, the collaborating attackers must know an identical fact, which is either user unique or VID unique. Corresponding shared link candidate fact types are connected by an n-ary relationship. This means, that the connected attackers can collaborate if they know an identical fact of the fact type, which is contained in the relationship. There are several attackers, who cannot merge their knowledge with others at all, because they do not know any link candidate fact type.

For the evaluation it must be checked whether one of the sensitive assets S1, S2, or S3 can be revealed by several attackers, who can collaborate by at least one relationship. Each group containing a smaller attacker group already knowing the asset, knows the asset, too. Thus, only minimal groups are explored. In the following, the sensitive fact sets are explored.

S1 Long Location Trace

For violating S1, attackers knowing a location or a location tracelet of the Mobile Node must be in the collaborating group. This can be an $\text{Eavesdropper}_{\text{LinkMN}}$, an $\text{Eavesdropper}_{\text{M-NvMA}}$, a vMA, or a homogeneous group of all Shareholders of a VID. Each of those attackers is at maximum seeing one location tracelet or several unlinked location tracelets according to 5.3.2.1.

In order to link several tracelets, an additional collaborating attacker is required, which knows a VID unique or user unique fact that remains constant beyond the duration of one location tracelet. Finally, the attackers need a common linking candidate in order to link their knowledge. The only VID unique or user unique fact types, which are constant beyond the duration of one location tracelet are the permanent address and the VID-identifier. They are known by the Correspondent Node, the fMA, and the $\text{Eavesdropper}_{\text{CNfMA}}$. For collaboration, those attackers need to share a common linking candidate with at least one of the attackers knowing the location.

Table 5.23 shows the existing possibilities. In order to gain a large location trace, the attackers need to know several tracelets, which then can be linked. There are two possibilities for gaining several tracelets. The first one is that the Mobile Node uses the same vMA or the same Shareholders a second time or that the Mobile Node moves back to the same networks again, so that the same eavesdroppers can observe the care-of addresses of the Mobile Node a second time. The other possibility is that several attackers knowing one tracelet each are collaborating.

Summarizing, the privacy engineer rates the protection as being sufficient. First of all, several different attackers must collaborate, secondly they must all serve the same VID, thirdly the possibility to collaborate at all depends on the user's movement or on the agent usage profile. The user has a degree of freedom here to shape the compromise between performance and privacy protection. Still, if all requirements are met, the user's location is not revealed forever, but only for the linked location tracelets. This bears the possibility, that the linked tracelets do not yet reveal the sensitive facts F3 and F4, i.e., the user's real name and other personal data.

Attacker Knowing Location	Attacker Knowing Permanent Address or VID-identifier	Common Link Candidate
$Eavesdropper_{LinkMN}$	Correspondent Node	SPI_{CNPA}
	fMA	SPI_{CNPA}
	$Eavesdropper_{CNfMA}$	SPI_{CNPA}
$Eavesdropper_{MNvMA}$	Correspondent Node	SPI_{CNPA}
	fMA	SPI_{CNPA}
	$Eavesdropper_{CNfMA}$	SPI_{CNPA}
vMA	Correspondent Node	SPI_{CNPA}
	fMA	SPI_{CNPA}, TH
	$Eavesdropper_{CNfMA}$	SPI_{CNPA}
Group of all Shareholders of a VID	Correspondent Node	SPI_{CNPA}
	fMA	SPI_{CNPA}
	$Eavesdropper_{CNfMA}$	SPI_{CNPA}

Table 5.23: Collaboration possibilities for disclosing S1

S2 Location and VID-identifier

The only attacker knowing the VID-identifier is the Correspondent Node. Thus, a Correspondent Node must collaborate with another attacker knowing the location. This is already evaluated in the previous section. Table 5.24 shows the relevant part again.

Attacker Knowing Location	Attacker Knowing Permanent Address or VID-identifier	Common Link Candidate
$Eavesdropper_{LinkMN}$	Correspondent Node	SPI_{CNPA}
$Eavesdropper_{MNvMA}$	Correspondent Node	SPI_{CNPA}
vMA	Correspondent Node	SPI_{CNPA}
Group of all Shareholders of a VID	Correspondent Node	SPI_{CNPA}

Table 5.24: Collaboration possibilities for disclosing S2 and S3

The disclosure of S2 does not need to follow all requirements like the disclosure of S1. It is enough for the Correspondent Node to find a collaborating attacker with one location fact or one location tracelet. Nevertheless, the Correspondent Node is unconditionally required for collaboration. This can be an advantage, because the Correspondent Node will often be either a private communication partner or a service provider. In many configurations, this will be a different party, i.e., neither the communication network provider nor eavesdroppers in the network. It can be claimed more likely that eavesdroppers collaborate among themselves or that agents of the communication system are collaborating. Finally, the dis-

closure of one location or of one location tracelet with the VID-identifier is for sure annoying, but can often be tolerated.

S3 n VID-identifiers

The only attacker knowing the VID-identifier is the Correspondent Node. For linking several VID-identifiers, several VID-identifiers must be known. Thus, either the Mobile Node must use one Correspondent Node with different VIDs, or several Correspondent Nodes have to collaborate. For both possibilities, a user unique fact is necessary for linking the contexts of the different VID-identifiers. The only user-unique fact types are the care-of address, a tracelet of it, and the MAC address. Those fact types are equivalent here, because their facts are only observed together and by the same attackers. Those fact types are known by an $\text{Eavesdropper}_{\text{LinkMN}}$, by an $\text{Eavesdropper}_{\text{MNvMA}}$, by a vMA, or by the group of all Shareholders of one of the VIDs, which are intended to be linked.

The collaboration possibilities are the same as in Table 5.24. But there is a difference. For linking several VID-identifiers, not only the respective Correspondent Nodes have to collaborate, but also the attackers knowing the care-of address of exactly those VIDs have to collaborate. Only then, the attackers can exchange their knowledge about the same VIDs and detect that all VIDs have the same care-of address and thus, the same user is the owner of the VIDs. The user can tune the compromise between performance and privacy protection here. Last but not least, the attackers only see the care-of address, when the Mobile Node is indeed communicating. Otherwise, the care-of address is hidden in the shares. Thus, the VIDs to be linked have to communicate simultaneously, so that the care-of address is simultaneously revealed and can be proved for identity by the attackers.

Summarizing, again a large number of attackers has to collaborate for linking VIDs. Moreover, those cannot just be arbitrary attackers, but they must serve or eavesdrop the same VID. The agents serving a VID are chosen by the user's choice and cannot be influenced by the attackers. Moreover, a link is only possible if the respective VIDs are communicating simultaneously. Finally, even if an attack on S3 is successful, only some of the user's VIDs are linked and not all of them.

5.4 Summary of Evaluation

The new architecture fulfills the protection goals regarding single attackers being agents of the communication system, the Correspondent Node, or eavesdroppers. Moreover, homogeneous attacker groups cannot violate the protection goals, assuming a sensible configuration and usage of the system.

Only large and heterogeneous attacker groups can violate the protection goals. Even then, protection partially still holds. F1 and F2 are never revealed, because this information is nowhere contained in the new architecture. Regarding S1, some location tracelets will be linked, but not all whereabouts of the user. Therefore, revelation of F3 and F4 is unlikely. Regarding S2, only some locations are linked to only one VID-identifier. In order to disclose the location for another VID-identifier, a new attacker group would be necessary. Regarding S3, some VID-identifiers could be linked, but not all of them. Again, other or additional VID-identifiers could be linked by another or an extended group of attackers.

Collaborative attacks are rather difficult to plan, because the set of potential attackers is determined by the user. The user chooses the agents serving a VID. Moreover, the user often chooses the Correspondent Nodes and some of the eavesdroppers are dependent on the user's location. For an attack on S3, it is even required that the attacked VIDs are communicating simultaneously. This could be exploited by active attacks, in which attackers are actively sending packets. Such attacks are out of the scope here and would require additional protection.

The danger implied by those threats can be tuned by the user to a certain degree. The more often the user changes the agents, the smaller the revealed and possibly linked tracelets will become. The more often the VID is changed on application layer, the less sensitive is a revealed location in the context of this VID or a link with other VIDs.

It must be stated that it is hard for a user to rate independence of the chosen agents. Even if they appear perfectly independent from each other, they might be operated by the same party. This will hardly be detected by the user.

The new architecture brings a great improvement as compared to Mobile IPv6, which was evaluated in chapter 3. The methodology from chapter 4 has made it possible to reduce the vulnerabilities one by one and to yield a system that is protecting the VID approach. A further improvement of privacy protection can be tried, but is assumed to increase costs in an unbalanced way.

The severeness of the remaining threats depends on the cardinality of the location tracelet, i.e., of the number of location facts, which can be collected and on the occurrence of situations, when VIDs could be linked by collaborating heterogeneous attackers, i.e., when several VIDs are communicating simultaneously and are served by the same vMA. Both cannot be stated in general, but depend on the scenario, in which the system is used. Thus, the next chapter evaluates the influence of scenario parameters on the remaining threats.

Chapter 6

Simulative Evaluation of Scenario-Dependent Threats in the New Architecture

Chapter 5 evaluated the vulnerabilities and threats in the new architecture, which are irrespective of the actual scenario, in which the architecture is used. This chapter complements the evaluation by detailing the behavior of the architecture with respect to its VID protection capabilities for certain scenarios. For this aim, event-driven simulation is used. Section 6.1 introduces the goals of the chapter before section 6.2 explains the metrics for the evaluation. After that, section 6.3 describes the simulation model. Section 6.4 shows the simulation results and section 6.5 concludes the chapter by an overall interpretation of the results.

6.1 Goals

The privacy evaluation in chapter 5 pointed out fundamental statements, which are always true for the system, irrespective of its usage. It shows, against which threats the new architecture principally protects and in which collaboration scenarios some of the protection goals might still be broken. An example for such a statement is the fact that certain groups of attackers are able to link VIDs, if the VIDs are communicating simultaneously and are served by the same vMA.

The simulative evaluation in this chapter complements the evaluation in chapter 5 by going one step further and by simulating certain scenarios. For those scenarios, the VID protection capabilities of the new architecture are evaluated. By systematically crossing the space of input parameters, this chapter evaluates how the VID protection capabilities of the new architecture develop for changes in the scenarios. By doing so, the chapter gives guidelines on how to parametrize the system. For the example above, this means to evaluate how long

it takes until a situation occurs in which several VIDs are communicating simultaneously via the same vMA, i.e., until a certain group of attackers can link those VIDs.

There are several parameters having an influence on the yielded VID protection level. Some of them count to the configuration of the system, i.e., the amount of servers available for a user and the algorithm, when to change the vMA. Others define the movement of the user and the communication parameters.

The chosen methodology for the evaluation is an event-driven simulation. The simulation tool is supported by the IKR Simulation Library [116].

In order to evaluate the yielded privacy, first the metrics to be measured in the simulation have to be defined. Section 6.2 will give this definition.

6.2 Metrics

Event driven simulation is able to provide statistical data about events, which occurred in the simulation or about data values of the simulation model. Those raw metrics given from the simulation are chosen in a way that they can be easily translated to statements about VID protection understandable by users.

Basically, the user is interested in three questions:

1. How long does it take until a new VID can be linked with any of the existing ones by information being revealed by the communication system?
2. How large will a location tracelet grow, which one vMA can observe?
3. On how many servers will the care-of address be revealed on the average?

6.2.1 Privacy Metrics From Literature

There are no established metrics to measure the severeness of threats to the VID approach. Privacy research has mainly aimed at providing metrics for the anonymity of users or more generally for the unlinkability of similar items of interest, e.g., messages. The most relevant metrics of those areas, which nevertheless all have a focus different from the ones in this thesis, are shortly introduced below.

Regarding anonymity, the anonymity set, cf. 2.1.2, is surely the metric, which is used most often. Nevertheless, the anonymity set assumes a uniform probability distribution of all the subjects in the anonymity set, e.g., each subject could be the sender of a message by the same probability. Therefore, a lot of research was done on metrics improving the anonymity set by considering the underlying probability distribution.

[39] structures anonymity metrics with a focus on probabilistic anonymity metrics. The best known metrics are introduced by [58] and [200]. They measure the effective size of the anonymity set by the entropy of the probability distributions that one of the given subjects is the sender of a message. Both works are developed independently from each other at the same time and only differ in a normalization. [211] extends this metric towards an information theoretic metric regarding unlinkability of similar items of interest. All those metrics assume that the number of users is known.

[38] extends those metrics by incorporating time dependency of observations. Moreover, it focuses on how to get the information for the quantification, which was neglected by the other works.

[221] shows that an entropy based metric is not always suitable, because it only measures the amount of information that is necessary to identify a user completely. It states that attackers are already successful if they can only guess the user's identity with a good probability. For this aim, new metrics are suggested.

[215] defines a metric for anonymity of entries in releases of a statistical database. The released data is said to be k-anonymous, if the information for each person cannot be distinguished from at least (k-1) individuals, from which also information is contained in the release. [145] shows two breaches of k-anonymity based on a lack of diversity of the sensitive attribute in the revealed data and based on the background knowledge, the attacker has. A new metric called l-diversity is proposed.

[113] takes a different approach. Therein, degrees of anonymity and unobservability are defined with the help of partial knowledge of function views.

[184] provide for qualitative metrics regarding the degree of anonymity and unlinkability of the sender and the recipient. The edges of the proposed scale are absolute privacy and provably exposed. In between there are four more degrees: Beyond suspicion, probable innocence, possible innocence and exposed without a proof. Beyond suspicion means that the sender of a message does not seem to be the sender more likely than all other users seem to be the sender. Probable innocence means that the sender does not seem to be more likely the sender than not to be the sender. Possible innocence means that there is a possibility that the real sender of a message was not the subject, which seems to be the sender, but any other subject. The other degrees mean, what their names intuitively suggest. The metrics do not only hold true for sender anonymity, but also for recipient anonymity and for unlinkability of sender and recipient.

Most of the metrics mentioned above aim at the expression of uncertain knowledge. In this thesis, uncertainty is not considered as motivated in section 2.5.2. The following sections introduce the metrics used in this chapter.

6.2.2 Mean Time For the First Link

According to chapter 5, VIDs can only be linked if some attacker observes the simultaneously identical care-of address of those VIDs. Only the vMA and the $\text{Eavesdropper}_{\text{MN-vMA}}$ can observe the care-of address. Thus, they are to be considered in this chapter. Nevertheless, it is sufficient to consider the vMA, because the results are directly correlated to those of the $\text{Eavesdropper}_{\text{MNvMA}}$, who observes the care-of addresses in the same way, i.e., the same care-of addresses at the same times as the vMA. Both attackers do not have a notion of the VIDs, whose care-of address they can link. To determine the VIDs, the attackers need to collaborate with attackers knowing the VID-identifiers. Such an attacker is only the Correspondent Node.

This chapter evaluates the situations, when such a collaboration principally can happen. For such a collaboration to happen, two VIDs have to communicate simultaneously over the same vMA so that this vMA can link the care-of addresses. For the final evaluation of the

privacy threat, it must be kept in mind, that even in such a situation the user's privacy is only at risk if the observing entity collaborates with at least one Correspondent Node.

The attacker $\text{Eavesdropper}_{\text{LinkMN}}$ is neglected here. First of all, this attacker is fixed and thus can only attack if the user happens to be on the respective access link. Secondly, by encryption on layer 2 those attackers can be restricted to only the access points, which lowers the probability of an attacker's presence even more. Groups of vMAs or $\text{Eavesdroppers}_{\text{MNvMA}}$ are also not considered in this chapter. This would be for further study.

The chosen metric reflecting the links of VIDs is the mean time it takes for the first link of a new VID to any of the user's existing VIDs. More exactly, it is the first possible occurrence of the situation that any attacker group determines a link if they are collaborating. For finding a suitable name for this metric, the analogon to the metric for the failure of a system—the Mean Time Between Failures, MTBF—is taken. The metric for the first link of VIDs is abbreviated as MTFFL, which means the **M**ean **T**ime **F**or the **F**irst **L**ink.

Figure 6.1 illustrates this metric. There are two existing VIDs, VID 1 and VID 2. At a certain point in time, the user creates a new VID. The bars depict communication actions of the respective VIDs. The user wants to know, when the new VID can be linked with any existing one for the first time. The new VID can be linked as soon as its care-of address is revealed on a vMA, which knows already the identical care-of address of another of the user's VIDs. Thus, the simulation examines, when the new VID is communicating simultaneously with any other VID on the same server for the first time. The time from the creation of the new VID until the time of the first simultaneous communication on one vMA is called MTFFL.

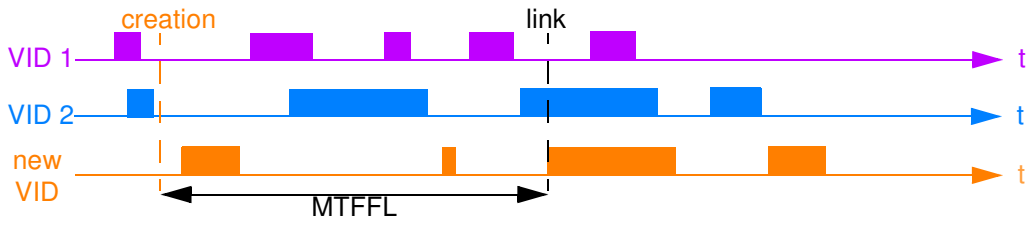


Figure 6.1: Illustration of MTFFL

The final interpretation of this value in the real world is up to the user. The user can decide that the time is too short and thus, more servers have to be used. Or the user can decide that the new VID is not very important and therefore a link with another VID can be tolerated with the given MTFFL. For the evaluations in this chapter, the MTFFL is a characteristic metric, because it allows to show how the value of this metric develops on changes of parameters of the system configuration or of the scenario.

6.2.3 Tracelet Cardinality

Locations are represented in the system of this thesis by care-of addresses. From the simulation, statistics about when the care-of address is revealed can be extracted. The number of consecutive care-of addresses, which an attacker can observe, corresponds directly to the number of locations, which the attacker can deduce from the care-of addresses. The relevant metric is the cardinality of the tracelet, which a vMA as potential attacker can gain.

There are slight simplifications, e.g., that each location is considered as being equally sensitive. This assumption is in line with the concentration of this thesis on the type-based interpretation of facts. In general, a larger tracelet cardinality, TC, will reveal more sensitive information about a user than a small TC will do. The focus is on the influence of scenario parameters on the level of VID protection. Thus, the cardinality of the tracelet is another significant metric.

The goal of the design of the communication layer here is that the tracelet cardinality is small enough in order not to influence the decision of the application layer about when to change a VID. Another possibility would be to make a cross-layer decision about when to change the VID and to consider the care-of address tracelets in the VID change decision on the application layer. The latter approach is not followed in this thesis.

6.2.4 Mean Number of Care-of Address Observers

During silent times, the new architecture hides the care-of address. On the other hand, it will disclose the care-of address on several servers, when several VIDs are communicating simultaneously. Thus, more servers can act as potential attackers and observe tracelets than this would be the case in a system with only one server. This is a natural trade-off when applying the distribution of trust concept by distributing the sensitive information on more parties. The claim is that the smaller amounts of information being revealed to one potential attacker lower the sensitivity of the potentially revealed information more than the growing number of potential attackers rise the risk of abuse.

For measuring this compromise, another metric is introduced, which measures the mean number of servers—potentially acting as attackers—which can observe the care-of address. So, the benefit of hiding the care-of address and the penalty of the multiple server system as compared to a single server system can be quantified. The metric is called **M**ean **N**umber of **C**are-of address **O**bservers, MNCO.

Question	Metric	Meaning
How long does it take until a new VID can be linked with any of the existing ones by information being revealed by the communication system?	Mean Time For the First Link, MTFFL	Mean time it takes for the first occurrence of a situation, where the new VID can be linked with any of the existing VIDs.
How large will a location tracelet grow, which one vMA can observe?	Tracelet Cardinality, TC	Number of consecutive care-of addresses a vMA as potential attacker can link to one user.
On how many servers will the care-of address be revealed on the average?	Mean Number of Care-of address Observers, MNCO	Quantification of compromise from hiding care-of address and from revealing it simultaneously on several vMAs.

Table 6.1: Summary of metrics

Table 6.1 summarizes the relevant questions and the metrics used for quantifying the answers to the questions. Moreover, the metrics are shortly explained in their meaning.

6.3 Simulation Model

This section presents how the system and the scenario are modelled for the simulation. Therefore, section 6.3.1 presents the model of the system and section 6.3.2 presents the parameters of the simulation.

6.3.1 System

A simulation model is an abstraction of reality and only reflects those parts of the system, which are necessary to evaluate the intended metrics. Therefore, the Shareholders are not modelled here, because they are neither needed for observing a tracelet nor for linking VIDs according to chapter 5. Similarly, outgoing communication originating from the user is neglected, because it does not reveal the care-of address, which is the base for all three privacy metrics.

The next subsection presents the simulation model. Subsection 6.3.1.2 then shows how the metrics are transformed to the simulation.

6.3.1.1 Model

Figure 6.2 shows the simulation model in its relevant details. Basically, there are two large blocks. The first one is the user and the VIDs of this user. The second block, on the right hand side, is the vMA reflecting the potential attacker. This could also be an Eavesdropper-$_{MNvMA}$, but for better understanding, only the vMA is mentioned throughout the chapter. On the left hand side there are the generators of the events in the simulation.

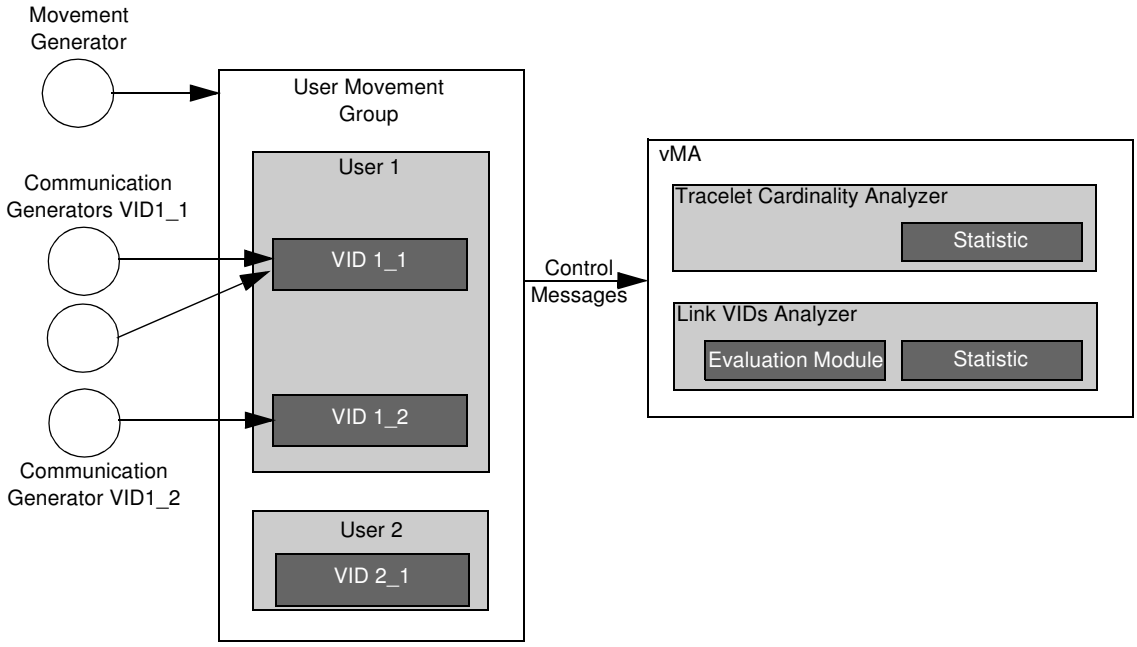


Figure 6.2: Structural simulation model

Users are members of a user movement group. This modelling prepares for extensions of the simulations in order to simulate jointly moving users, e.g., in a train or in a car. The user movement group is triggered by a generator of movement events for the contained users.

The movement events are identical for all users of the user movement group and for all VIDs of a user.

Each user has one or several VIDs. The generators of the communication events are triggering the VIDs, because different VIDs of the same user can be used with different application services. A VID can be triggered by one or more generators reflecting the fact that a user can use the same VID for one or more application services.

The movement and communication of the VIDs are signalled by control messages to the vMA indicating, e.g., when a vMA is started to be used by which VID and when this VID is communicating, i.e., when the vMA can see the care-of address. This level of abstraction omits the level of detail of packet exchanges, the care-of address management, and the secret sharing mechanisms in the simulation model.

The vMA has two different kinds of analyzers—one for determining the tracelet cardinality and another one for determining links between different VIDs. Each analyzer has its own statistic for recording the TC or the MTFFL, respectively. The analyzer for the MTFFL additionally has an evaluation module, which currently evaluates the MTFFL between the new established VID and any other existing VID. This module could be replaced by another module evaluating another metric for measuring links between VIDs.

[129] and [15] contain more details and reasoning of the simulation model and the simulation tool. The next section details determination of the relevant metrics in this model.

6.3.1.2 Metrics

Measuring the MTFFL has to be realized in a way avoiding instationarity of the simulation. When a new VID is created in reality, the existing VIDs are communicating already. The new VID is introduced in a stationary state of the system. Thus, other VIDs than the considered one keep on running in the simulation after a detected link. In order to simulate a new start of the measurement in reality, the relevant state of the evaluated VID is reset and the time measurement is restarted for the next value. If the state were not reset, a new link would be recognized immediately, because both VIDs on the given vMA are communicating for a non-zero duration.

The other VIDs are in an arbitrary state when the new measurement starts—representing a new established VID. The new VID is silent when being created, but can start either immediately or an arbitrary time later after creation. Thus, the simulation waits with the starting of the new measurement until the considered VID is in a silent phase and a new communication event is scheduled. This next communication event and can start immediately or at a later point in time—like in reality.

Figure 6.3 illustrates the MTFFL. The *new VID* is the VID, which is considered to be newly created. *VID 1* and *VID 2* are already existing and communicating every now and then. All sketched VIDs are being served by the same vMA. On the first sketched communication of the new VID, it directly collides with the communicating VID 2 and the first link is recognized. After the detected link, the measurement is stopped until a new communication event of the *new VID* is scheduled while being in a silent phase. Then, the relevant state is reset and the measurement is started again. In reality, this corresponds to a new VID being established at the time of the starting measurement. From there on the time to the next link is measured. This is the case, when the new VID is communicating and VID 1 starts to com-

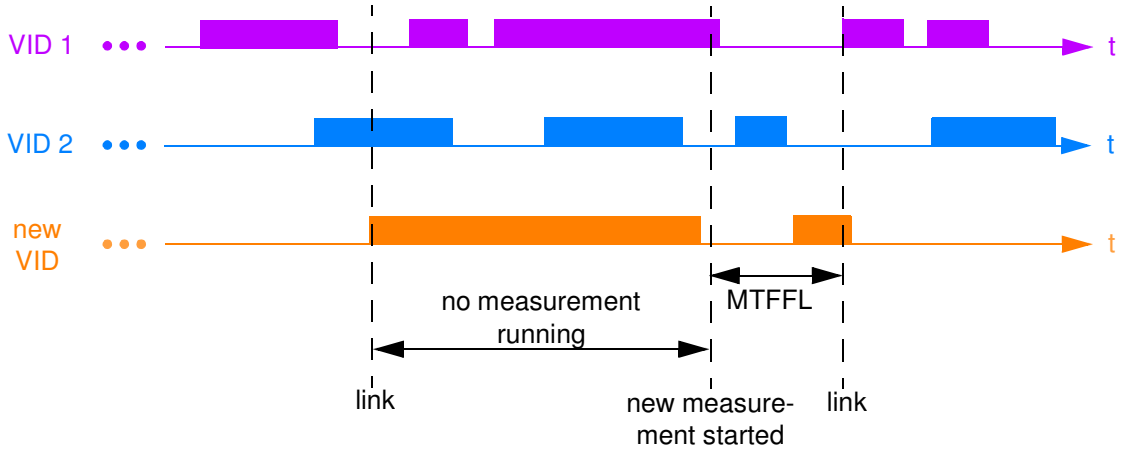


Figure 6.3: MTFFL in the simulation

municate in the example. Like will be pointed out in section 6.3.2.2 the communication here is memoryless.

The tracelet cardinality is the mean value of a counter of the disclosed care-of addresses, which one vMA observes for a VID. The counting starts when a VID starts to communicate or a communicating VID moves from another serving vMA to the given vMA. The counting ends when the VID moves away to a new vMA. In silence phases, the care-of address is not visible to the vMA and thus not counted. If the VID re-starts to communicate before moving away to another vMA, counting continues. The tracelet cardinality here does not take into account linked VIDs, i.e., does not merge tracelets of linked VIDs.

The MNCO is always updated on a user's movement. This update must be done for the user, because the MNCO is a metric of the user and not of a certain VID of the user.

6.3.2 Parameters of the Scenario

The scenario, which is stimulating the simulation, mainly comprises the movement and the application services of the user. This translates to a movement model as well as a communication model needed as stimulators.

The time is measured in the unit of seconds. In principle, this unit does not matter and the time can also be assumed to be arbitrary time units. The unit "seconds" are only used for translating the size of communication actions in bytes given by the configuration files of the generators to the duration of communication actions, which is needed for the simulation. The base for this is an available bandwidth of 2 Mbit/s, i.e., in the range of wireless links, which is considered to be the bottleneck of an end-to-end communication. Both communication partners are assumed to be attached to wireless access links.

The next sections explain the chosen user movement model in section 6.3.2.1 and the communication model in section 6.3.2.2.

6.3.2.1 User Movement

It is virtually impossible to model the behavior of users in a realistic way, because human beings do not adhere to mathematical models. Sometimes, it is possible to model aggregations of several users, whose random behaviors compensate each other. Here, a modelling

of a single user is needed. The focus of the thesis is on understanding the behavior of the system with respect to the VID protection. Therefore, a simple mobility model for the user's movement has to be employed, so that the effects of the system can be clearly identified and understood and are not blurred by effects of a complicated movement model, whose abstraction is too detailed.

Relevant for evaluating the behavior of the system is not the exact movement of the user, but the interarrival time between changes of the network membership, $T_{A,\,mov}$. Those changes are leading to a new care-of address for the user and thus, are the relevant input for the tracelet cardinality. Network changes during a communication action of a user influence the tracelet cardinality.

If a pure IP network architecture like [123] or [122] is assumed, IP subnetwork sizes come down to single wireless LANs, Ethernets, or GSM/UMTS cells. Thus, the interarrival time of care-of address changes corresponds to the interarrival time of cell handovers.

Assuming a rough form of a circle for the serving area of a network, there are more possibilities to walk through the circle by paths being shorter than the diameter. Therefore, the distribution must have more shorter interarrival times than longer interarrival times. According to [199], a negative exponential distribution is not the optimum, but is an adequate approximation.

For the goal of this thesis of evaluating the behavior of the system with respect to its stimulators, the exactness of the approximation is assumed to be adequate, because a comparison of different parameter sets is the goal. For a more exact quantification, more sophisticated models should be used. Then, also the form of the network coverage areas and the distributions of the velocity must be considered [199].

For this thesis, a rough estimation defines the corridor of the mean values of the interarrival times. Towards low values, the corridor starts close to zero, because it can be that a cell is just touched by a user, i.e., the care-of address changes happen shortly after each other. Towards large values, a UMTS macro cell with about 1000m radius being crossed by a pedestrian of about 5 km/h (1.4 m/s) directly through the diameter gives an indication. Non-moving users are not considered in this estimation.

$$T_{A,\,mov} = (2 \times 1000m) / \left(1,4\frac{m}{s}\right) = 1429s$$

Thus, the corridor for the mean values of the interarrival time of handovers is between some few seconds and some thousands of seconds.

6.3.2.2 Communication

The same arguments like for the movement models hold true for the traffic models. As motivated in chapter 1, future applications drive the need for VIDs and thus for a VID supporting architecture. It is not yet possible to foresee the traffic of future applications. Even for today's applications, this is not always possible. To complicate things, a user can use one VID for several application services, which results in a composition of traffic of several applications. Moreover, only the communication times on the last hop, i.e., on the vMA, are relevant. They can additionally be influenced by traffic shaping in the network.

Again a simple model is to be used for the simulations in order to understand the effects of the system behavior with respect to the VID protection. When more detailed traffic models will be available and a concrete system is intended to be built, the results can be revalidated with the new models of the concrete applications.

The communication here is modeled by traffic generators with a mean holding time, $T_{H,\,com}$, and a mean interarrival time, $T_{A,\,com}$. Thus, different communication actions of one generator can overlap. For the probability distributions of $T_{H,\,com}$ and $T_{A,\,com}$, negative exponential probability distributions are chosen.

Negative exponential probability distributions are generally accepted for traffic parameters in order to evaluate the principal behavior of a new system. Negative exponentially distributed holding times are known, e.g., from telephony applications [136]. The memoryless property seems to be realistic for future applications, too.

The interarrival time is also chosen with a negative exponential distribution, here. There are two reasons for that. First of all, the generators in the simulation tool do not correspond to the number of possible traffic sources in reality. The generators are abstractions of the communications of one application layer service. Such an application layer service can be, e.g., a telephony service, or a community service like a friend finder service. Behind such a modelled service there can be a large number of real users triggering a telephone call to the considered user or triggering a friend finder request. Secondly, the application services are assumed to be memoryless, i.e., the interarrival times of the communication is not dependent on the past. The application services are assumed to be independent of each other, e.g., the user does not stop using one application when starting to use another one.

The generators here follow all the same traffic model. The distinction of different generators provides for extensibility of the tool in order to simulate services with different traffic parameters in the future.

The communication is modelled on the base of communication actions with an interarrival time and a duration, not on the base of single packets. It is relevant, during which time a VID is receiving packets, because the vMA can observe care-of addresses during this time. It is not relevant, how many packets the VID is receiving. Therefore, it is assumed that the packets of one communication action, e.g., the transmission from a webcam or the reception of a friend finder update, are coming in such a short sequence that the attacker would deduce an ongoing communication activity, anyway. Moreover, the relevant base of the simulation are the revealed care-of addresses during communication actions and it is improbable that care-of addresses are changing more than once during the small pauses between packets of one communication action. This would be the only error being introduced by using the abstraction of communication actions.

For the corridor of the mean values of the traffic model, Table 6.2 shows some estimations. In order to show the behavior of the system, sometimes those ranges are surpassed in the following simulation runs.

The mean value of the overall offered traffic A is defined as $A = T_{H,\,com} / T_{A,\,com}$ and cannot be larger than one Erlang. This is due the fact, that in the simulated scenarios, the generators are always generating traffic with constantly configured parameters and do not have silent times, e.g., due to different times at a day. A scenario offering more than one Erlang would bring the system in an instable state of overload.

Application Service	Interarrival Time of Activities	Duration of One Activity
Friend Finder	600 s (10 min)	2s
File Sharing	40000 s (2 files per day)	1800 s (1/2 h)
Webcam transmission	80000s (1 transmission per day)	7200 s (2 h)

Table 6.2: Approximation of communication parameters

For explanation, the user's wireless access link has to be regarded as bottleneck of the system, where all offered traffic to the considered user concentrates. The traffic of all sources to the considered user has to be handled by this link and is thus restricted to one Erlang. The system could handle more than one Erlang per user if losses were considered. Here, it is assumed that the system does not suffer from losses, because losses would result in retransmissions of higher layer protocols like TCP. Realtime traffic, for which retransmissions do not make sense are neglected here, because they are assumed to constitute only a minor part of future applications. Thus, this part of the system can be regarded as a delay system with one server, i.e., the user's access link.

In a delay system, all offered traffic will be carried by the servers. Thus, all offered traffic must pass this single access link. If the offered traffic were more than the link could handle, the utilization would have to be higher than one or the delay times would grow infinitely. Both is not possible in reality.

6.3.3 System Configuration

Regarding the configuration of the system there are basically two relevant parameters. First of all, the user can decide how many servers to rent for the agents of the communication system. In the simulation only the number of vMAs is evaluated. This number is limiting the protection a user can afford as most likely servers will not be free of charge in a commercial scenario. The number of vMA determines to a large degree the number of necessary servers, because only one sMA is needed per VID but several vMAs changing over time are needed. The number of Shareholders is also smaller than the number of vMAs, because they are changing more slowly than the vMAs.

To reduce the number of necessary servers, it is possible to use one physical server for different logical functions of different VIDs, e.g. as sMA of VID 1 and as vMA of VID 2. Thus, in most of the sensible configurations, the number of servers is equal to the number of vMAs and the other logical functions are distributed among the available physical servers.

The other parameter is the probability, whether the vMA should be changed on a user's movement or not. This probability follows a binomial distribution. Like motivated in chapter 5, vMA changes should occur on location changes. This parameter influences the maximal tracelet cardinality, because a tracelet will be ended if the vMA is changed. More sophisticated algorithms for changing the vMA are possible but out of scope.

6.4 Evaluation

This section presents the results of the scenario-dependent evaluation. The relevant parameters presented in the preceding sections can be classified according to how easily the user can influence them. For the user it is most easy to influence the probability to change the vMA after a movement. This does not bring higher costs and is not determined by any outside event. Still easily but most likely combined with higher costs, the user can change the number of available servers.

As compared to those parameters, it is harder to influence the number of VIDs, because this number is usually determined by the application layer based on privacy constraints of the applications. Finally, it is very hard to influence the interarrival time of care-of address changes, because this is ultimately bound to the user's movement and to the available networks. It cannot be assumed that the user changes the movement patterns in order to support the privacy level offered by the communication system. The same arguments hold true for the parameters of the communication actions, i.e., the interarrival time of the communication actions and their duration. In addition, those parameters usually are determined by the communication partners and not by the user, because incoming communication is considered, here.

Most of the results contain the 95% confidence intervals based on the batch-means method. The confidence interval indicates the statistical significance of the respective value. The actual value lies with the probability of 95% in the interval according to the definition. The base for the determination of the confidence interval is the student-t distribution [139]. In some figures, the confidence interval is not contained, because they are derived from other simulated results.

Figure 6.4 shows the organization of this section. Mobile IPv6 was the starting point. There, the MTFFL is zero when considering the Home Agent as potential attacker, because all VIDs can be immediately linked by the identical home address. The tracelet cardinality is infinite, because the Home Agent sees each care-of address. The MNCO is one, because the care-of address is always revealed on exactly one server.

Section 6.4.1 evaluates the new architecture with only one server and compares the results to the results of Mobile IPv6. Section 6.4.2 evaluates the new architecture with several servers but without server changes of the VIDs and compares the results to the ones from the preceeding section. Finally, section 6.4.3 evaluates the new architecture in its full functionality and compares the results to the preceeding section.

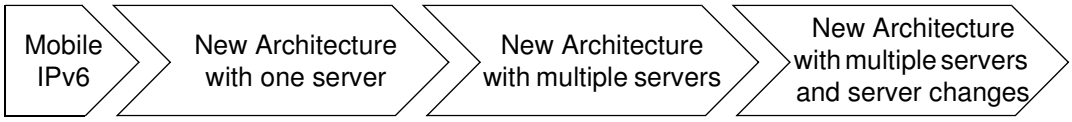


Figure 6.4: Organization of the evaluations

6.4.1 Basic Behavior

The simplest parametrization is to use only one server. Remember that only servers available for the logical vMA role are considered in this chapter. With only one available server the probability to change the server is zero. Thus, the mechanism of changing servers is dis-

abled. Moreover, all VIDs are being served by the same server. The only enabled mechanism of the architecture is to hide the care-of address during silence times, i.e., when no VID is communicating.

This configuration is comparable to a Mobile IPv6 system with an added secret sharing scheme for hiding the care-of address during silence times. This configuration serves well as a reference for evaluating the influence of the number of VIDs in section 6.4.1.1 and for evaluating the traffic parameters in section 6.4.1.2.

6.4.1.1 Influence of Number of VIDs on MTFFL

In the following simulations, the user uses 100 application services simultaneously with a different number of VIDs on which the services will be evenly distributed. The overall offered traffic is changing and is evenly distributed amongst all services. The mean interarrival time of the communication actions of the services, $T_{A,\,com}$, is changing with the overall offered traffic. The mean duration of the communication actions, $T_{H,\,com}$, is fixed.

The results are independent of the interarrival times of the user's movement, which hence is not considered in this section. The movement only triggers changes of the care-of address and thus indirectly changes of the servers, on which the VIDs are communicating. Care-of address changes do not influence the link of VIDs and server changes are only relevant when considering more than one server.

Because only one server is available, all VIDs will be served by this server. Thus, there is a possibility for linking two VIDs as soon as both are communicating simultaneously.

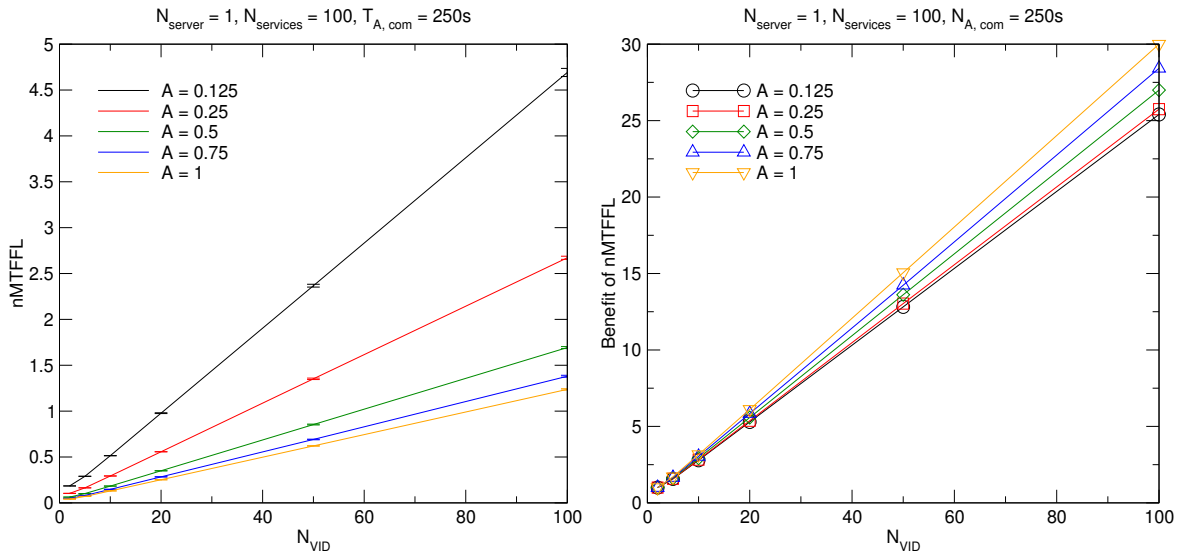
The MTFFL is growing with a growing $T_{A,\,com}$. If a VID is communicating rarely, it can only rarely be linked by the revealed care-of address. Therefore, the MTFFL is normalized to an *nMTFFL* by relating it to the $T_{A,\,com}$ in this chapter. Thus, it can easily be seen how often a VID can communicate—as a mean value—until it can be linked with another VID for the first time. A side effect of this normalization is that it is not important for the result, whether the overall offered traffic is changed by changing the $T_{A,\,com}$ or by changing the $T_{H,\,com}$.

Figure 6.5 (a) shows the behavior of the nMTFFL on a changing number of VIDs and for different overall offered traffic. The absolute numbers are in the range of nearly zero to about five $T_{A,\,com}$, i.e., 1250 seconds corresponding to about 20 minutes.

The nMTFFL rises more than linearly with a decreasing overall offered traffic. This leads to the conclusion, that the statistical multiplexing gain by serving several VIDs by only one server is significantly higher for VIDs, which are not communicating a lot.

The nMTFFL is also rising with the number of VIDs. The reason is that the offered traffic per VID is decreasing with an increasing number of VIDs. Thus, the certain VID, to which links of arbitrary other VIDs are evaluated, communicates less. For that reason, the attacker has the possibility to link this VID to any other VID on fewer occasions.

Figure 6.6 depicts this. The overall communication actions are shown over the time as black bars. The solid lined blue box contains all communication actions of the considered new VID in case of two overall VIDs, whereas the solid orange box contains the communication actions in the case of four overall VIDs. It can be seen that the attacker has fewer occasions to link the considered VID in the orange case, because there are fewer communication



(a) Absolute values (b) Benefit regarding 2 VIDs

Figure 6.5: Influence of N_{VID} on nMTFFL

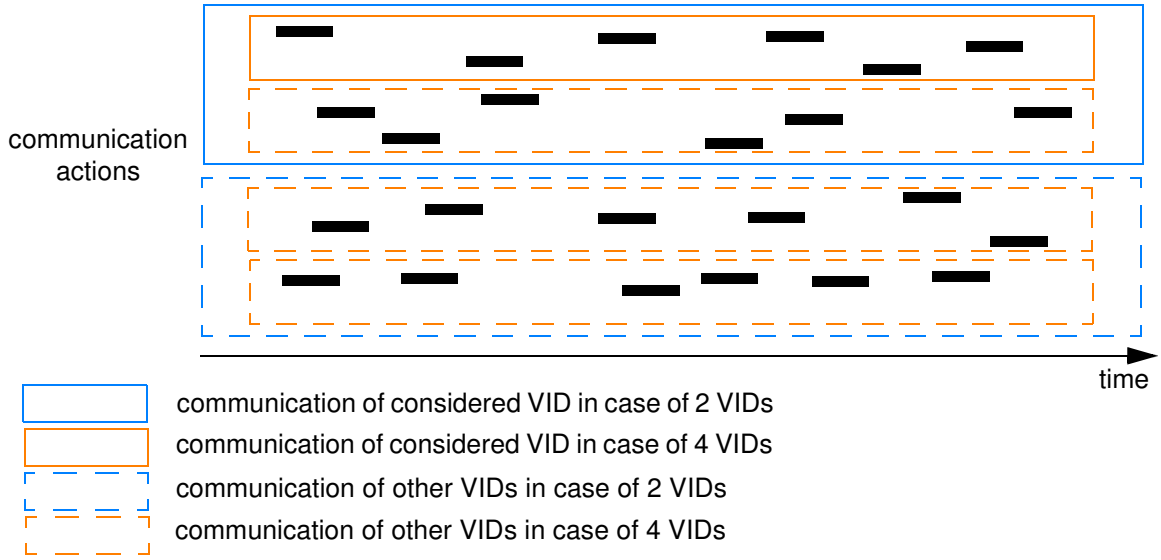


Figure 6.6: Overall offered traffic evenly distributed on a different number of VIDs

actions during which the care-of address is disclosed and can be compared to the care-of addresses of other VIDs. This effect outweighs the effect that the offered traffic of the other VIDs increases by the share of traffic, which was taken from the considered VID in the simulated scenario.

Figure 6.5 (b) shows the direct benefit of multiple VIDs by relating the nMTFFL to the nMTFFL in the scenario with two VIDs. The benefit rises linearly with an increasing number of VIDs. A 50 times higher number of VIDs yields a longer nMTFFL in the range of 25 to 30 depending on the overall offered traffic. For the highest offered traffic, the benefit is the highest, because the base value with two VIDs is very low. This is due to the more than linear decrease of the nMTFFL with increasing offered traffic.

The result of this evaluation is the fact that a user can positively influence the nMTFFL—and thus, the MTFFL—by increasing the number of VIDs and thus distributing the overall

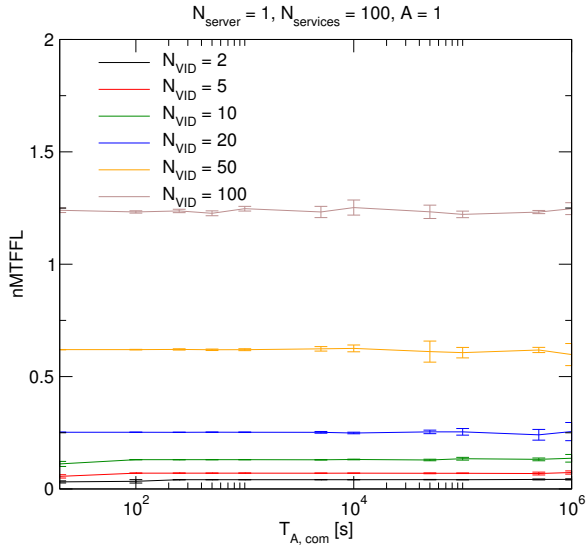


Figure 6.7: Influence of $T_{A,\,com}$ on nMTFFL

offered traffic across more VIDs. Moreover, the overall offered traffic significantly influences the nMTFFL in a way that a rising overall offered traffic lowers the nMTFFL.

A comparison to a pure Mobile IPv6 system already reveals a benefit. In a pure Mobile IPv6 system, the MTFFL would be zero. As soon as a new VID is established, the Home Agent can link it to all other VIDs, no matter whether the new VID communicates or whether it does not communicate.

6.4.1.2 Influence of Traffic Parameters on MTFFL

This section examines the influence of the traffic parameters on the nMTFFL. For this aim, the overall offered traffic is kept constant at a value of one Erlang. The nMTFFL is shown over the $T_{A,\,com}$ on a logarithmic scale for different numbers of VIDs. Because the overall offered traffic is constant, the $T_{H,\,com}$ is changed in line with the $T_{A,\,com}$.

Figure 6.7 shows the result. A change of the traffic parameters does not change the nMTFFL. This means that the pure MTFFL only rises proportionally to the growing $T_{A,\,com}$. The nMTFFL rises linearly for different numbers of VIDs as was shown in section 6.4.1.1.

6.4.1.3 Influence of Number of VIDs on MNCO

After evaluating the influence of the number of VIDs on the VID-centric metric MTFFL, this section evaluates the influence of the number of VIDs on the MNCO, which is a metric for the user. This section shows the time, during which the care-of address is revealed in a system with one server. This serves as reference to which the trade-off in configurations with multiple servers will be compared.

Figure 6.8 shows the resulting graph. The MNCO varies with the overall offered traffic, but remains constant with the number of VIDs. Even in case of an overall offered traffic of one Erlang, the MNCO is not one, because partially, the communication actions occur in parallel. Such parallel communication actions can be from the same VID or from different VIDs. Figure 6.6 visualizes that the number of VIDs on which the communication actions are distributed does not influence the MNCO.

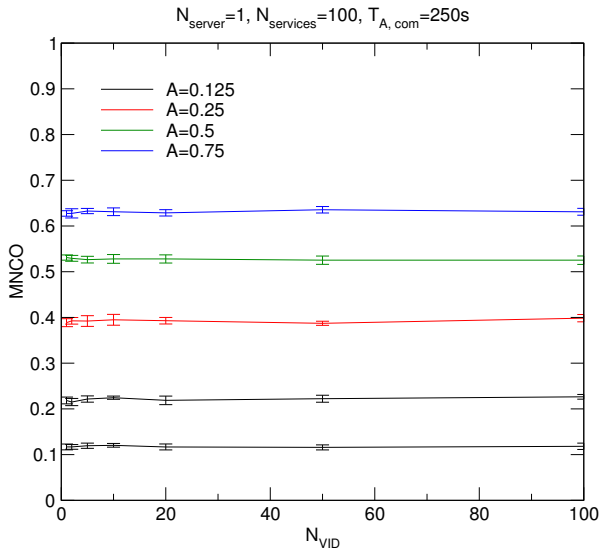


Figure 6.8: Influence of N_{VID} on MNCO

This scenario already takes the secret sharing scheme of the care-of address in account. During silent times, i.e., when no VID is communicating, the care-of address is not disclosed. As compared to a pure Mobile IPv6 system, this already brings a benefit. In a pure Mobile IPv6 system, the care-of address is 100% of the time disclosed to the Home Agent.

6.4.1.4 Influence of Number of VIDs on Tracelets

With one single server acting as vMA, the care-of address tracelets will grow infinitely long. The major difference as compared to a pure Mobile IPv6 system is that the tracelets will grow more slowly and will not be continuous, because the care-of address is hidden during communication breaks.

6.4.2 Multiple Servers Without Server Changes

In this section the complexity of the architecture is increased according to Figure 6.4. Besides the hiding of the care-of address during silence times, multiple servers acting as vMA are introduced. Thus, the user's VIDs are spread among the available servers. The mechanism of changing those servers is still disabled by setting the probability for a server change to zero.

The scenario bases on an equal distribution of VIDs on the available servers. The simulations are in fact run with one server and a changing number of VIDs on this server, which corresponds to the overall number of VIDs on all servers. In other cases, the simulation result would largely depend on the initial state, i.e., on how many VIDs are on the server of the VID, for which the links are measured.

6.4.2.1 Influence of Number of Servers on MTFFL

The simulation runs with 100 VIDs, a $T_{A,\,com}$ of 250s and different overall offered traffic. The results are normalized to the results with one server from section 6.4.1 and thus, show the benefit of multiple servers.

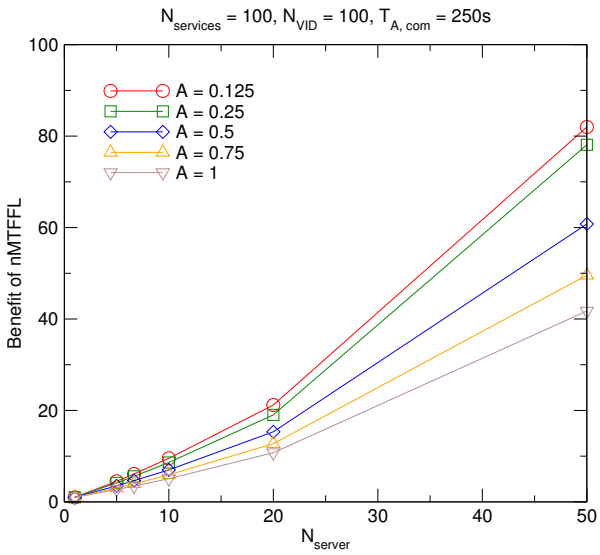


Figure 6.9: Benefit regarding 1 server

Figure 6.9 shows the results. It can be seen that the gain as compared to one server rises with the number of servers as well as with the decreasing of the overall offered traffic. With 50 servers, the nMTFFL is about 80 times higher than with one server, if the offered traffic is 0.125 Erlang. This already results in times in the range of a day.

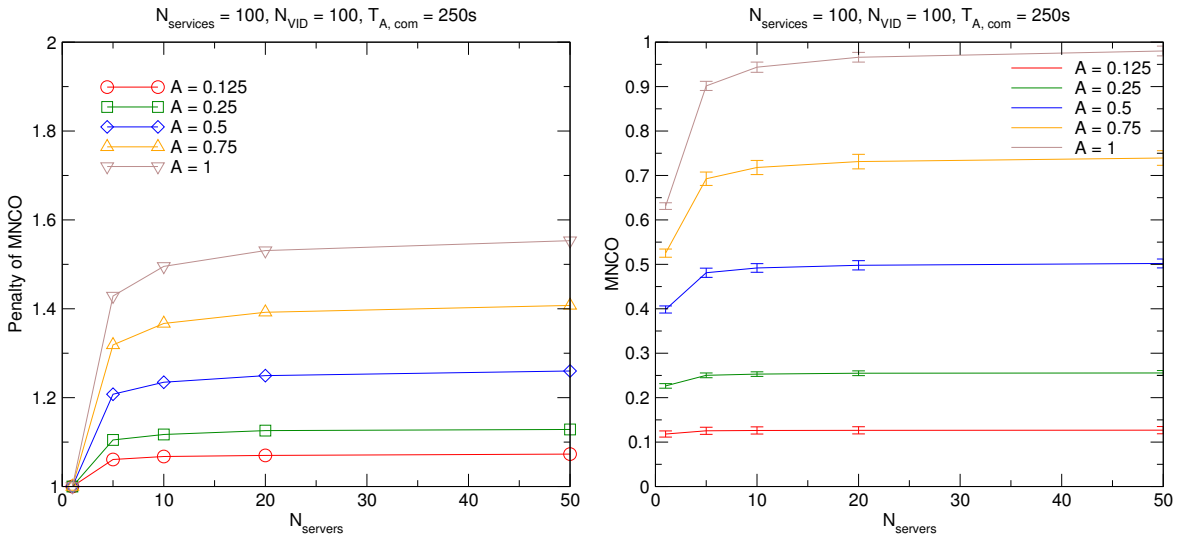
Both effects are based on the decreasing offered traffic on one server. In case of 50 servers, the offered traffic per server is 1/50 of the offered traffic in case of one server. This corresponds to a decrease of the overall offered traffic in the simulations of section 6.4.1. Thus, the nMTFFL in case of 50 servers is higher than with only one server. The more servers are available, the larger is the difference of offered traffic per server as compared to a scenario with only one server.

For low offered traffic, the difference is higher, because the nMTFFL decreases rapidly with a rising offered traffic. Thus, a reduction to 1/50 brings more absolute difference in those cases.

6.4.2.2 Influence of Number of Servers on MNCO

This section shows the MNCO as compared to the MNCO in the scenario with only one server. It refers to the penalty of disclosing the sensitive care-of address multiple times when having multiple servers.

Figure 6.10 (a) shows that the penalty approaches a constant value. From there on, an increase of the number of servers does not reveal the care-of address more than before. The reason for this maximum value is that there are nearly no overlapping communication actions taking place on the same server. The more servers are available, the fewer VIDs are served on the same server and thus, the fewer overlapping communication actions occur. Overlapping communication actions of different VIDs on different servers are counted doubled during the overlap time and thus, contribute to the MNCO doubled and do not result in a lower MNCO.



(a) Penalty regarding 1 server (b) Absolute MNCO

Figure 6.10: Influence of $N_{servers}$ on MNCO

Figure 6.10 (b) shows the absolute values of the MNCO. The maximum values approach more or less the value of the overall offered traffic. This supports the argument, that there are nearly no overlapping communication actions on the same server.

For high overall offered traffic, the maximum value is not fully reached. This is due to the fact that there are still some overlapping communication actions of the same VID and of the other VIDs on the same server. Therefore, there are also a significant amount of links measured, which is reflected towards a smaller nMTFFL in Figure 6.9. For low overall offered traffic, the maximum value is reached, because the probability for parallel communication actions—of the same VID or of different VIDs—on the same server is very low. This is also reflected in the long nMTFFL in Figure 6.9 for low overall offered traffic.

6.4.2.3 Influence of Number of Servers on Tracelets

As for the tracelet cardinality, there are no changes as compared to section 6.4.1.4. Because a VID does never change a server, this server will still collect care-of addresses forever.

6.4.3 Multiple Servers With Server Changes

This section evaluates the full new architecture. This is the final step of the evaluation according to Figure 6.4. Besides the hiding of the care-of address and multiple servers, the VIDs are changing the servers. By enabling the mechanism of server changes the user's movements and the probability to change the vMA server on a movement become relevant. Both are additional input parameters for the simulations. Moreover, tracelets of finite cardinality are evolving when servers are changed.

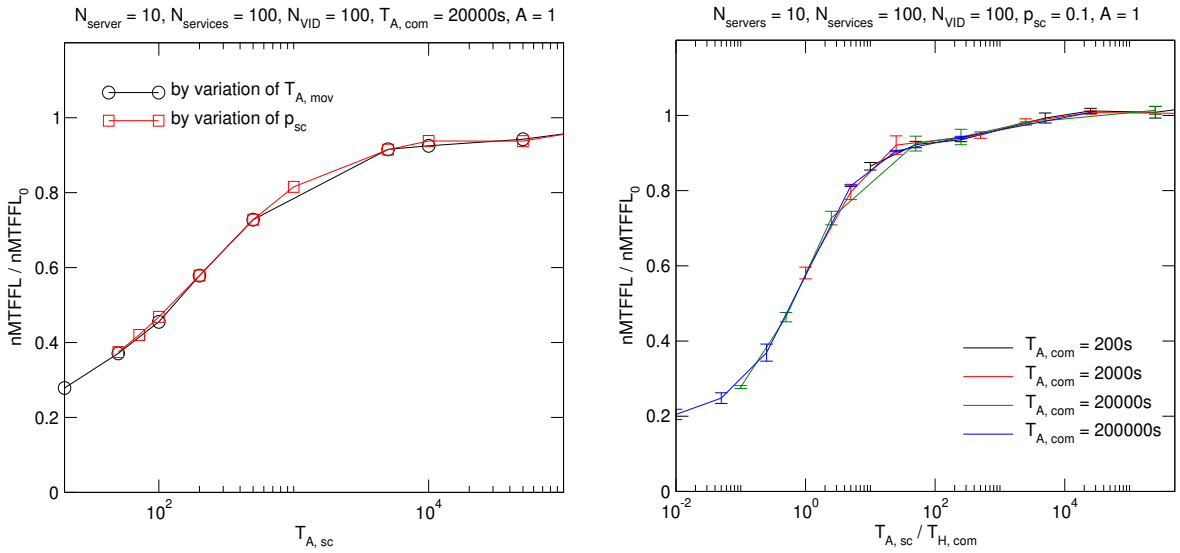
Section 6.4.3.1 evaluates the nMTFFL, whereas section 6.4.3.2 evaluates the tracelet cardinalities. The user's movement is expressed in the mean interarrival time of the care-of address changes, which is called $T_{A,\,mov}$, in order to make it obvious that it depends on the

user's movement. The interarrival time of care-of address changes also depends on the size of the IP subnetworks, through which the user is moving.

The MNCO remains like in section 6.4.2.2 on a changing user movement and server change probability. The user's movement does not have an influence on the disclosure of the care-of address. The server change effects compensate each other in the considered homogeneous scenario with identical offered traffic per VID. If a VID is the only communicating one on the old server and moves to a server with other communicating VIDs, the MNCO decreases. If instead a communicating VID moves from a server with other communicating VIDs to a server without communicating VIDs, the MNCO increases. Thus, the MNCO is not followed further in this section.

6.4.3.1 Influence of Movement and Server Change Probability on MTFFL

For the simulations of this section, ten servers are chosen with an overall offered traffic of one Erlang being evenly distributed among 100 VIDs. The following Figure 6.11 (a) shows the first results of the nMTFFL, which are normalized to the value without server changes from section 6.4.2.1. The nMTFFL of the scenario without server changes is named $nMTFFL_0$.



(a) Variation of $T_{A,\,sc}$ by different parameters

(b) Influence of $T_{A,\,sc}/T_{H,\,com}$

Figure 6.11: Influence of server changes on nMTFFL / $nMTFFL_0$

A first observation shows, that the nMTFFL only depends on the mean interarrival time of server changes $T_{A,\,sc}$. $T_{A,\,sc}$ is yielded by multiplying the mean interarrival time of care-of address changes $T_{A,\,mov}$ with the mean reciprocal of the server change probability p_{sc} $T_{A,\,sc} = T_{A,\,mov} \cdot 1/p_{sc}$.

It is not relevant, whether $T_{A,\,sc}$ is varied by varying $T_{A,\,mov}$ and keeping p_{sc} constant or by keeping $T_{A,\,mov}$ constant and varying p_{sc}. While the user will probably not adapt the movement pattern, it is easily possible to adapt the server change probability.

The second observation shows that the graph can be divided into two areas. In the first one, $T_{A,\,sc}$ is much higher than the $T_{H,\,com}$. This means that during a communication action there

will hardly be any server change. Server changes in silent times do not have any influence on the link of VIDs, because the care-of address is not revealed. Thus, the graph approaches the value of one, i.e., the value of section 6.4.2.1 without server changes, for this area.

In the second area, $T_{A, sc}$ is rather small, i.e., there are server changes during communication actions. Then, the nMTFFL is decreasing with a decreasing $T_{A, sc}$. This is due to the fact that a communicating VID affects other VIDs when it is changing the server. If the moving VID is the only communicating VID on the originating server, there is the possibility that on the new server, there is already another communicating VID. Then, those VIDs can be linked, which would not have been the case if the VID had not moved. Thus, server changes negatively affect the MTFFL and nMTFFL.

Figure 6.11 (b) shows, that in fact the nMTFFL is only dependent on the relation between the $T_{A, sc}$ and the traffic parameters. The simulation evaluates the same overall offered traffic with different $T_{A, com}$ and thus with different $T_{H, com}$. It shows that the graphs are identical when relating the $T_{A, sc}$ to $T_{H, com}$. It also shows that in the first mentioned area with a result similar to the system without server changes, this relation is much higher than one, i.e., the $T_{A, sc}$ is much longer than $T_{H, com}$.

Figure 6.11 means that the penalty for the nMTFFL by server changes is only dependent on the ratio of the interarrival time of the server changes to the duration of the communication actions. If the server changes happen rarely, i.e., if $T_{A, sc}$ is large as compared to the duration of the communication actions, there is nearly no penalty as compared to the scenario without server changes. As will be seen in the next section, this allows for scenarios with a pure benefit for the tracelet cardinality and without a negative effect on the nMTFFL.

6.4.3.2 Influence of Movement and Server Change Probability on Tracelets

For the following simulations, the setup consists of 20 servers, 100 VIDs, and an overall offered traffic of 0.1 Erlang. The tracelet cardinality *TC* refers to the number of care-of addresses, which an attacker can see and link as belonging to one user. After the user changes the serving vMA for the considered VID and until the old server will be used again some time in the future, the server is not able to realize that both tracelets belong to the same VID. Thus, a tracelet is the number of care-of addresses a server observes during one serving session, during which it serves a given VID.

The mean maximal tracelet cardinality, which a server can observe, is bound and called TC_{max}. It is determined depending on p_{sc}: $TC_{max} = T_{A, sc} / T_{mov} = 1 / p_{sc}$. On each movement, the VID issues a random experiment, whether the server should be changed or not. A sever change will definitely stop the collecting of care-of addresses by the server, i.e., restrict the size of the tracelet cardinality. Therefore, TC_{max} is called the maximum tracelet cardinality. Nevertheless, this is not a sharp maximum, because a) it depends on a random experiment and b) the same server can be chosen again as new server, in which case the collection of care-of addresses can proceed. Both is due to the simple server change algorithm chosen here. A comparison of different algorithms is out of scope.

Figure 6.12 shows the basic behavior. The tracelet cardinality is standardized by relating it to the maximal tracelet cardinality resulting in the *normalized tracelet cardinality, nTC*. Its behavior is shown over $T_{A, mov}$. There are two graphs, one with a maximal tracelet cardinality of ten and another one with a maximal tracelet cardinality of 100. The $T_{A, mov}$ is plotted

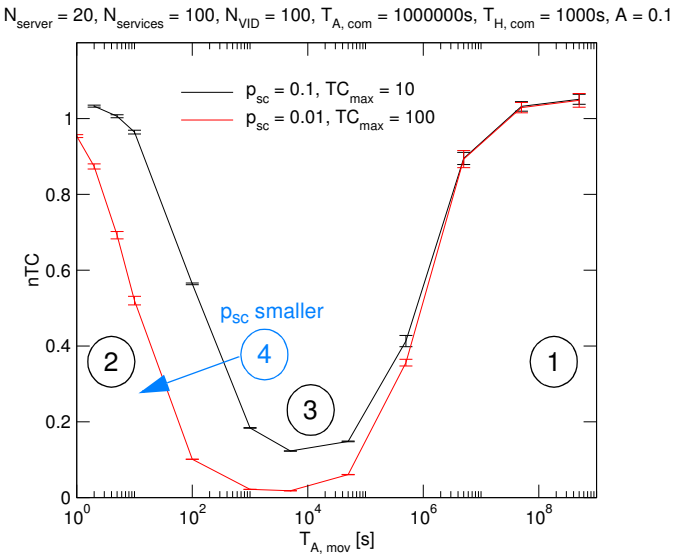


Figure 6.12: Basic behavior of tracelet cardinality

on a logarithmic scale. $T_{A,\,com}$ is very long, but chosen for showing the whole behavior of the graph. Like shown later, the behavior is similar for smaller values of $T_{A,\,com}$.

There are four observations. First, the TC approaches TC_{max} for very large $T_{A,\,mov}$. Secondly, the TC approaches TC_{max} for very small $T_{A,\,mov}$. Thirdly, TC is smaller than TC_{max} in between. Fourthly, the area of relatively small tracelet cardinalities is becoming broader with a larger TC_{max}. Both graphs remain similar towards large $T_{A,\,mov}$. The TC remains longer below TC_{max} with a growing TC_{max} towards short $T_{A,\,mov}$.

The described behavior is defined by two processes. For understanding them, the tracelet cardinality is illustrated in Figure 6.13. There are two servers, *server 1* and *server 2*. The VID is communicating in form of the orange bars. The blue arrows indicate care-of address changes. The black arrows indicate server changes. The servers cannot observe care-of addresses in silent times. Thus, server 1 is observing eight care-of addresses and server 2 is observing nine care-of addresses. It can be seen that the tracelet cardinality depends on both, the $T_{A,\,mov}$ and on the communication pattern.

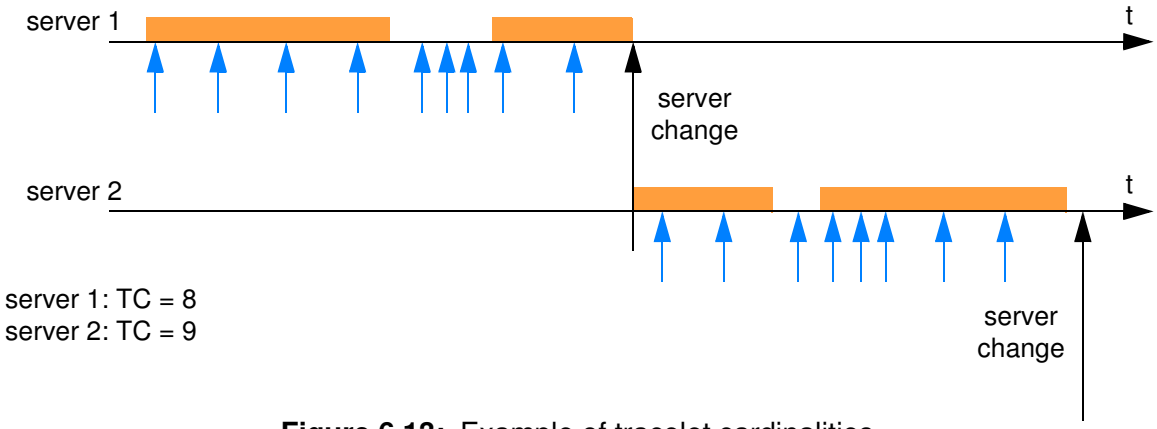


Figure 6.13: Example of tracelet cardinalities

In the area, where $T_{A,\,mov} \gg T_{A,\,com}$, it can be assumed that each care-of address will be disclosed, because during each validity period of a care-of address, there will be at least one

communication action. Thus, the tracelet cardinality is only restricted by the server changes and approaches TC_{max}.

In the area, where $T_{A,sc} = T_{A,mov}/p_{sc} \ll T_{H,com}$, it can be assumed that the care-of address is permanently disclosed during a serving session or not disclosed at all, whereby the latter case does not result in an update of the statistic. Thus, the maximal tracelet cardinality is reached. Again, TC is only restricted by the server changes.

Thus, the TC is best if $T_{A,mov}$ is not much larger than $T_{A,com}$ and $T_{A,sc}$ is not much shorter than $T_{H,com}$.

In the areas with $TC < TC_{max}$, both effects are overlapping, i.e., TC depends on the movement and on the communication pattern. The upper bound, where TC approaches TC_{max} is not dependent on p_{sc}, whereas the lower bound depends on p_{sc} and moves to the left with a smaller p_{sc}, i.e., a larger $T_{A,sc}$. This is because $T_{A,mov}$ must be smaller in order to fulfill the condition $T_{A,sc} = T_{A,mov}/p_{sc} \ll T_{H,com}$. The move to the left corresponds to a factor of about ten. This is intuitive, because ten is the factor, by which p_{sc} is smaller in the red graph than it is in the black graph. Therefore, $T_{A,mov}$ must be also 10 times lower to achieve the same $T_{A,sc}$.

The area with $TC < TC_{max}$ is defined by the parameters of the scenario, i.e., by the user's movement pattern and by the communication patterns of the used application services. By changing p_{sc}, the user can only define the absolute height of the graph and its width, i.e., the absolute TC_{max} and the area, in which TC is lower than TC_{max}.

A high p_{sc}, i.e., a small tracelet cardinality, negatively influences the MTFFL in general, which shows a compromise between the values of both metrics. Nevertheless, there can be areas, where even the nTC is already small and where the nMTFFL is not suffering from a relevant penalty by the server changes.

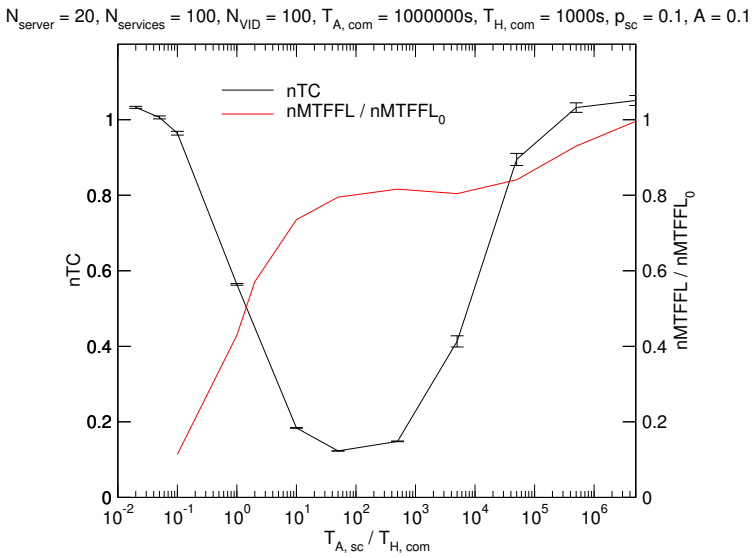
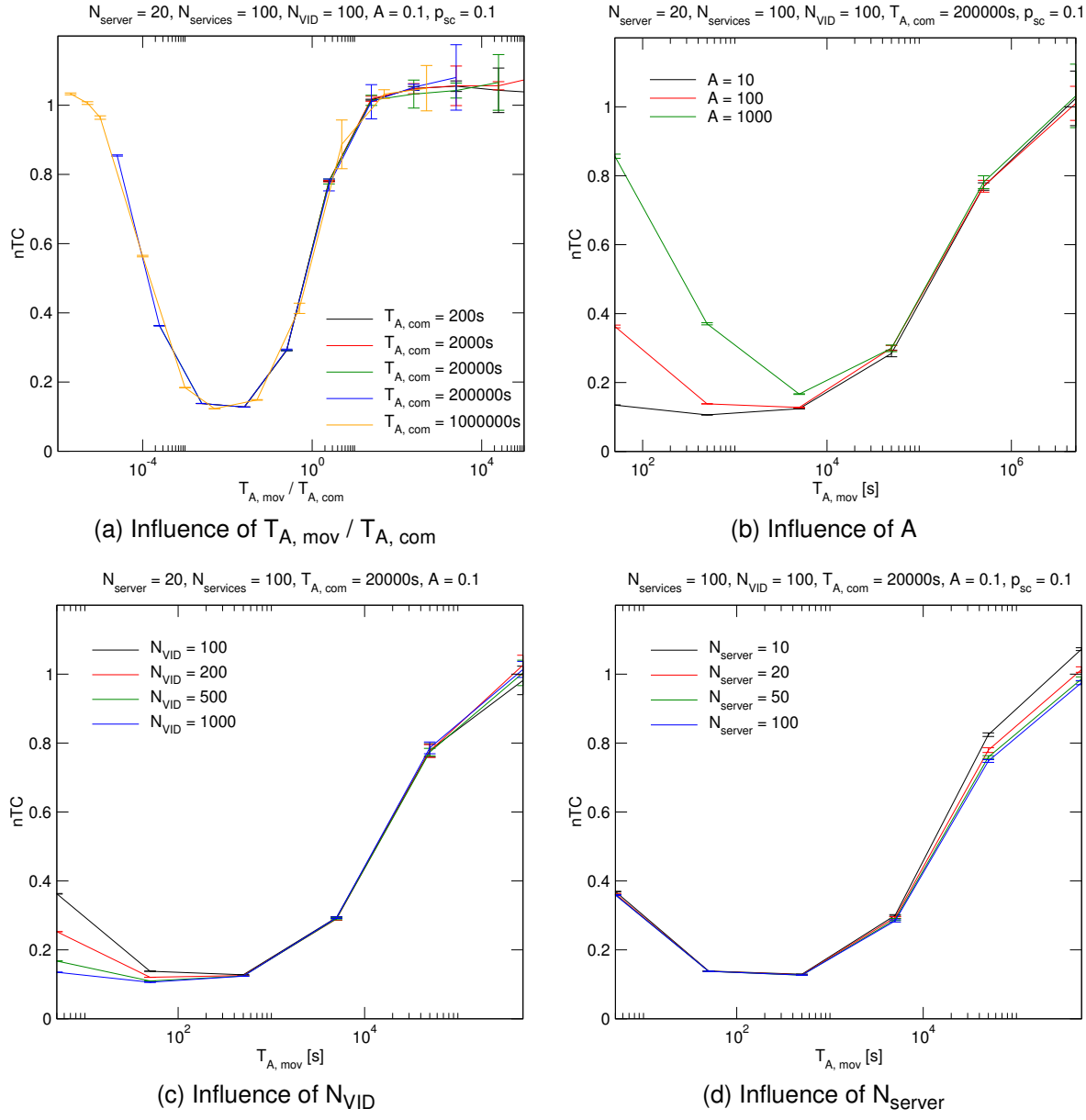


Figure 6.14: nTC and nMTFFL / $nMTFFL_0$ over $T_{A,sc} / T_{H,com}$

Figure 6.14 shows such a situation. In the black graph the nTC is shown and in the red graph the nMTFFL / $nMTFFL_0$ is shown over $T_{A,sc}$ related to $T_{H,com}$. $nMTFFL_0$ is the nMTFFL in the same scenario but without server changes. For a value of $T_{A,sc} / T_{H,com}$ of

100, the nTC is already in the low area whereas the penalty of nMTFFL is still low, i.e., the nMTFFL is very close to the nMTFFL in the scenario without server changes.



(a) Influence of $T_{A,\,mov}$ / $T_{A,\,com}$

(b) Influence of A

(c) Influence of N_{VID}

(d) Influence of N_{server}

Figure 6.15: Behavior of nTC

Figure 6.15 (a) shows the nTC for different $T_{A,\,com}$ with the same overall offered traffic of 0.1 Erlang. The nTC is plotted over $T_{A,\,mov}$ related to the $T_{A,\,com}$. It is obvious that the nTC only depends on this relation. This means that the graph over the pure $T_{A,\,mov}$ would move towards the right side with a growing $T_{A,\,com}$.

Figure 6.15 (b) shows the behavior of the tracelet cardinality on a varying overall offered traffic. It can be seen that the area of $TC < TC_{max}$ becomes smaller with a growing overall offered traffic, which is due to the growing $T_{H,\,com}$, if $T_{A,\,com}$ is constant and the offered traffic is growing. $T_{A,\,sc} = T_{A,\,mov}/p_{sc} \ll T_{H,\,com}$ is already fulfilled for larger $T_{A,\,mov}$, then.

Figure 6.15 (c) shows the influence of the number of VIDs on the normalized tracelet cardinality. For each VID, one application service is foreseen. The overall offered traffic and the $T_{A, com}$ are kept constant, so that $T_{H, com}$ changes with the number of services. This is the reason for the diverging graphs towards small $T_{A, mov}$. For small $T_{A, mov}$, the tracelet cardinality is dependent on $T_{H, com}$. Towards large $T_{A, mov}$ instead, the tracelet cardinality only depends on $T_{A, com}$, which is identical for all graphs.

Figure 6.15 (d) shows the influence of the number of servers on the normalized tracelet cardinality. With more servers, the tracelet cardinality is slightly decreasing. The reason is in the server selection algorithm, which is a simple random process, here. Thus, the same server can be chosen again. In that case, the tracelet can become longer than TC_{max}. Selection of the old server, again, happens more frequently, if fewer servers are available.

6.5 Summary of Evaluation

The results presented in section 6.4 allow for several conclusions, which are summarized and put into relation to each other in this section. First of all, in section 6.5.1 the main results from the previous sections are summarized. Then, in section 6.5.2 the influence of the mechanisms of the new architecture on the privacy metrics is qualitatively shown. In section 6.5.3 the influence of the scenario parameters on the privacy metrics is presented. In section 6.5.4 the tracelet cardinality and the MTFFL are set in relation to each other. Finally, in section 6.5.5 conclusions regarding a careful dimensioning of the system are drawn.

6.5.1 Summary of Raw Results

The main raw results are that the MTFFL can be in an acceptable range of values even if using fewer servers than VIDs. The tracelet cardinality depends largely on the interarrival time of the user's care-of address changes. The tracelet cardinality is best for an interarrival time of the user's care-of address changes being not much higher than the interarrival time of the communication actions and not much lower than the duration of the communication actions.

The protection mechanism of changing servers in order to reduce the tracelet cardinality affects the MTFFL negatively, if the interarrival time of the server changes is in the range of the duration of communication actions and lower. Thus, there is a compromise between the values of both metrics, the MTFFL and the TC. There are areas of the scenario parameters with a low tracelet cardinality, but with still a high MTFFL.

As regarding the MNCO, the care-of address is still revealed less than in plain Mobile IPv6, although it might be revealed to several potential attackers at the same time.

6.5.2 Influence of System Configuration on Privacy Metrics

This section summarizes the results regarding the influence of different mechanisms and system configurations. According to the different configurations, different mechanisms are contained in the new architecture. Table 6.3 shows the different configurations according to Figure 6.4. In the rows, different aspects are compared. The first row names the enabled protection mechanisms. The next row states, which VIDs might possibly be linked by

potential attackers. Then, the MTFFL properties of the different configurations are named, being followed by the TC properties. The last row summarizes the MNCO results.

System Configuration	Mobile IPv6	New Architecture with one server	New Architecture with multiple servers	New Architecture with multiple servers
Enabled Mechanisms	none	CoA hiding during silent times	CoA hiding during silent times Spreading VIDs among several servers	CoA hiding during silent times Spreading VIDs among several servers Server Changes
Possibly linked VIDs	all	all	N_{VID}/N_{Server}	depending on server selection algorithm
MTFFL	0	> 0	> 0	> 0
TC	∞	∞	∞	< ∞
MNCO	1	< 1	< 1	< 1

Table 6.3: Influence of mechanisms

Mobile IPv6 as starting point does not contain any of the evaluated protection mechanisms. It has the worst performance regarding all three privacy metrics. The new architecture with one server enables the care-of address hiding during silent times and thus yields an MTFFL larger than zero and an MNCO smaller than one. The new architecture with multiple servers achieves the same qualitative performance but achieves additionally that not all VIDs might be linked. Finally, the full architecture achieves in addition a finite tracelet cardinality.

The next section summarizes the influence of the configuration and scenario parameters on the MTFFL and on the tracelet cardinality.

6.5.3 Influence of Parameters on Privacy Metrics

This section gives an overview of the qualitative influence of system and scenario parameters on the privacy metrics nMTFFL and nTC. In the following Table 6.4, the first column indicates the changed parameter. It is assumed that all other metrics do not change in a given row.

Parameter	Influence on nMTFFL	Influence on nTC
higher overall offered traffic	shorter MTFFL	longer TC in the boundary of TC_{max}, smaller range of TC<TC_{max}
longer $T_{A,\,com}$ and longer $T_{H,\,com}$	no influence	dependent on value of $T_{A,\,mov}$ either longer or shorter TC in the boundary of TC_{max}
slower movement (longer $T_{A,\,mov}$)	longer MTFFL	dependent on value of $T_{A,\,mov}$ either longer or shorter TC in the boundary of TC_{max}
more VIDs	longer MTFFL	shorter TC towards small $T_{A,\,mov}$
more servers	longer MTFFL	slightly shorter TC
longer TC_{max} (smaller p_{sc})	longer MTFFL	longer TC, broader range of TC<TC_{max}

Table 6.4: Influence of parameters

It can be seen that all parameters but the communication parameters influence the nMTFFL in a clear way. The influence on the nTC depends for most parameters on the area in the graph, where the scenario is located.

6.5.4 Different Systems: MTFFL vs. Tracelet Cardinality

Different privacy aware communication systems provide for different VID linking and tracelet cardinality properties. Figure 6.16 gives a qualitative idea about those properties. Systems having only one server, e.g., pure Mobile IPv6, have an infinite tracelet cardinality and an MTFFL of zero. Systems with one server per VID, e.g., Mobile IPv6 with one Home Agent per VID, have an infinite MTFFL, but also an infinite TC. Only with mechanisms like server changes, the TC becomes finite.

The proposed system can be parametrized in a wide area colored blue. With a very high p_{sc}, the TC is very small, but the MTFFL, too, which is the bottom left corner of the figure. In order to reach the upper left corner, a sufficient amount of servers and an intelligent algorithm distributing the VIDs on the servers must be available. The proposed system can also be parametrized for being equivalent with the systems in the upper right and in the lower right corner.

The arrows above and on the right indicate the parameters making the MTFFL longer or the TC shorter respectively. The slower movement and a changed $T_{A,\,com}$ and $T_{H,\,com}$ also

influence the metrics, but their influence is dependent on the overall parameter set and thus, cannot ultimately be indicated in the figure.

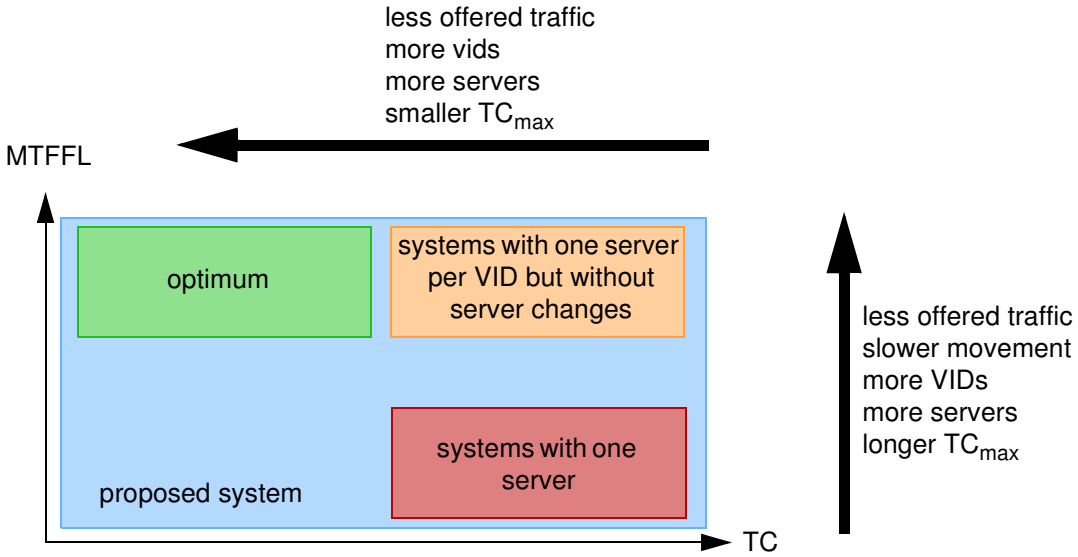


Figure 6.16: Different systems in the light of MTFFL and TC

Figure 6.17 shows the development of the nMTFFL and of the TC for a change of servers and for a change of the overall offered traffic. All other parameters are kept constant in the simulations. The graph only shows a part of Figure 6.16, whose axis are going to infinity.

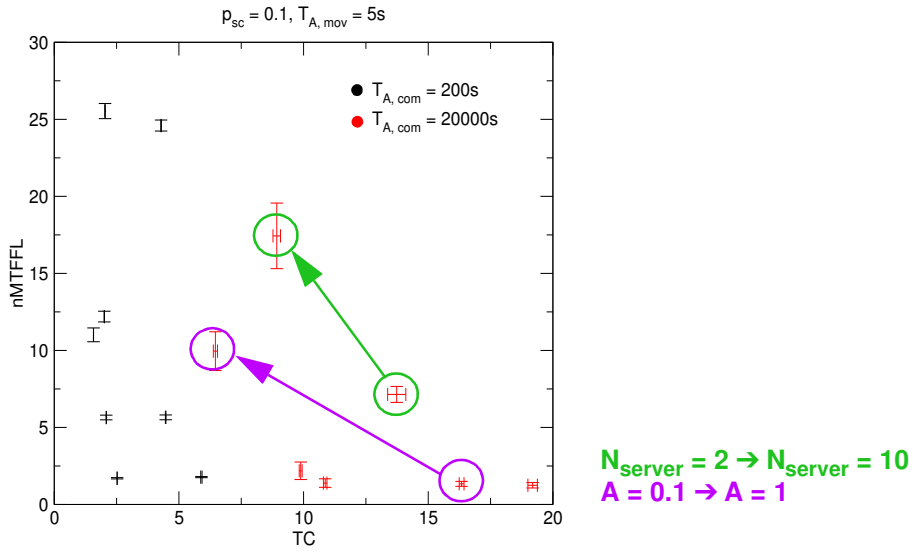


Figure 6.17: Influence of parameter changes on nMTFFL and on TC

Both metrics can also be summarized in one index defining the properties of an evaluated system with respect to its VID protection. This ProtectionIndex evaluates to $ProtectionIndex = (a \times MTFFL)/(b \times TC)$. The ProtectionIndex grows with a growing MTFFL and with a shrinking TC. The relation of the factors a and b defines the importance of the MTFFL compared to the importance of the tracelet cardinality.

Systems without server changes have a ProtectionIndex of zero because of the infinite tracelet cardinality. Systems with multiple VIDs on one server and without further protec-

tion mechanisms like the secret sharing mechanisms for instance, also have a ProtectionIndex of zero, because of the MTFFL of zero. A ProtectionIndex of infinity would correspond to perfect systems with respect to one or both metrics. Such systems either have an infinite MTFFL or a tracelet cardinality of zero or both. The proposed system is in between those extremes depending on the parameters of the system and of the usage scenario.

6.5.5 Dimensioning

One of a user's questions will be, how many servers to spend for given application services and VIDs. A qualitative answer can be divided into three classes:

- One VID per server

  This makes sense, if the VID is used for a service with a high offered traffic irrespective of the usage time. The reason is that in case of a high offered traffic, the care-of address will be revealed during most of the time. Thus, the danger of linking the VID by revealing the identical care-of address of another VID is high.

  This dimensioning can also make sense for a mid-range offered traffic, when the VID is used for a long time, because the potential damage, if this long living VID will be linked to another one is probably high.

  With this dimensioning, there is no statistical multiplexing gain yielding in fewer needed servers than VIDs.

- Few VIDs per server

  This dimensioning provides for a small statistical multiplexing gain, i.e., requires fewer servers than VIDs. It can be used for a mid-range offered traffic per VID, when the VIDs are not used for a long time. If the VIDs shall be used for a long time, it is better to use this dimensioning only for low offered traffic, i.e., if the probability for a link is low because the care-of address will only rarely be revealed.

- Many VIDs per server

  While this dimensioning provides for a high statistical multiplexing gain, it can only be used for VIDs with a low offered traffic and short usage times. Then, the damage if the VIDs are linked will usually also be smaller, because not much information is revealed during the usage time.

There is another influence on the required number of servers, i.e., the maximal tracelet cardinality. A reasonable proceeding is to define the maximum tracelet cardinality first. In a real scenario, this requires a lot of privacy knowledge or proper privacy estimation methods. This maximal tracelet cardinality defines the parameter p_{sc}. With this p_{sc}, the user can determine the number of servers in order to yield an acceptable MTFFL.

For the dimensioning the overall evaluation has to be kept in mind. Even if VIDs are communicating simultaneously on the same server and thus, they could principally be linked, this is only possible if all necessary entities are a) malicious and b) collaborating with each other. Depending on the probability of this attacker scenario, which has to be rated by the user in the specific case of the involved providers, the acceptable probability of a short MTFFL has to be determined by the user.

Chapter 7

Conclusions and Further Work

Applications of the future will increase the threat to a user's privacy. An important approach for protecting privacy is to use multiple virtual identities, VIDs. Those VIDs must be protected by the communication system. This requires a special design of communication systems for supporting the VID approach.

The design for a VID supporting communication system implies evaluation and often the subsequent improvement of several candidate building blocks for the use in the architecture. Those are recurring tasks. Therefore, sound methodologies are needed firstly to evaluate the vulnerabilities and threats of a communication system or one of its building blocks and secondly to subsequently improve the system or the building blocks.

This thesis develops both methodologies for system evaluation and for improving systems. They are introduced in a general way and applied to Mobile IPv6 as an example in order to achieve a mobility management building block, which supports the VID approach. The methodologies provide for comparable and well-reasoned results and reduce the danger of overlooking design flaws. Moreover, they are strongly formalized so that also non-experts can use them.

The application of the evaluation methodology to Mobile IPv6 shows, that all aimed protection goals are broken. The application of the system improvement methodology to Mobile IPv6 results in a new architecture for mobility management. The architecture can be configured differently for different compromises between privacy protection and performance. The relevant parts of this architecture were realized as proof of concept on the base of Mobile IPv6.

The new architecture was evaluated by two methodologies. First, the scenario-independent threats and vulnerabilities were evaluated by the introduced methodology. The result is that all protection goals are fulfilled for single attackers as well as for homogeneous attacker groups. Heterogeneous attacker groups might break some of the protection goals in specific scenarios and configurations. The second evaluation examined the remaining threats depending on configuration and usage scenarios by event-driven simulation. The main result is that there is a trade-off between the values of both metrics, the one measuring the VID linking properties and the metric measuring the amount of disclosed information.

There are scenarios, in which both goals are met well. The simulation also showed the behavior of the new architecture with respect to changes in the scenario and in the configuration. The results give guidelines for the configuration of the system.

Chapter 2 provided the fundamentals necessary to understand the thesis. First of all, this is a general background in security and privacy. Secondly the bases underlying the methodologies were introduced: Basics in knowledge engineering, mathematical functions, and functional dependencies. Because the developed methodologies were used to provide for a new architecture for mobility management, chapter 2 also introduces mobility management in IP and Mobile IPv6 as starting point of the thesis. Finally, the field, in which the thesis lies was structured and the scope was defined in this structure.

Chapter 3 motivated and introduced the methodology for evaluating systems with respect to threats and vulnerabilities regarding the VID approach. Each explained step was applied to Mobile IPv6 as an example. It was started by developing the knowledge model consisting of the elementary fact type view and the dynamic view. Then, the evaluation of the knowledge model was introduced and applied to the model of Mobile IPv6. The result was, that all aimed protection goals are broken by plain Mobile IPv6.

Chapter 4 then motivated and introduced the methodology for improving systems. Like in chapter 3, each step was first introduced and then applied to Mobile IPv6. The steps are first, reduction of observation vulnerabilities, then reduction of inference vulnerabilities, and finally reduction of vulnerabilities of linking sensitive fact sets. After each change of the system, i.e., usually after each step of the methodology, the previous steps have to be reapplied in order to identify newly introduced vulnerabilities. The result is a new architecture for mobility management, which is configurable to a large degree in order to provide for different preferences, focussing on privacy or focussing on performance. The architecture can be mapped for realization to different technologies. A prototype proved the feasibility and the adequate implementation possibility of the most relevant functionalities.

Chapter 5 evaluated the new architecture with respect to scenario-independent vulnerabilities and threats. Therefore, the knowledge model of the new architecture was built—starting with the elementary fact type view being followed by the dynamic view. The evaluation of the knowledge model showed that the architecture protects all aimed protection goals against single attackers and against homogeneous attacker groups consisting of attackers of the same type. There are heterogeneous attacker groups, which can break some of the protection goals. Those attacks are hard to plan, because the potential attacker must play certain roles regarding the attacked user and the roles are allocated by the user. Moreover, the attacker groups must consist of quite diverse attackers, which are not very likely to collaborate in common scenarios.

Chapter 6 evaluates the new architecture with respect to scenario and configuration dependent threats. Therefore, three metrics are introduced, the mean time it takes until a newly established VID can be linked with any other existing VID for the first time, the MTFFL, the cardinality of the location tracelets which a potential attacker can gain, the TC, as well as a metric quantifying the trade-off regarding revelation of the care-of address at multiple potential attackers at the same time, the MNCO. The main results are that the MTFFL can be in an acceptable range of values even if using fewer servers than VIDs. The tracelet cardinality depends largely on the interarrival time of the user's care-of address changes. The tracelet cardinality is best with an interarrival time of the user's care-of address changes

being not much higher than the interarrival time of the communication actions and not much lower than the duration of the communication actions. The protection mechanism of changing servers in order to reduce the tracelet cardinality affects the MTFFL negatively, if the interarrival time of the server changes is in the range of the duration of communication actions and lower. Thus, there is a trade-off between the values of both metrics, the MTFFL and the TC. There are areas of scenario and configuration parameters with a low tracelet cardinality, but with still a high MTFFL. As regarding the MNCO, the care-of address is still revealed less than in plain Mobile IPv6, although it might be revealed to several potential attackers at the same time.

Further work could extend the methodologies towards handling value-based evaluations and towards handling uncertain information. The architecture could be extended by aspects like security or charging, by including interoperating parts of a larger architecture, e.g., IPv6 Neighbor Discovery or ICMP messages, or by extending the scope towards handling micro-mobility. The scenario-dependent evaluation could be extended towards evaluation of groups of linked VIDs and the respectively resulted tracelet cardinalities. Moreover, groups of collaborating attackers could be simulated.

Bibliography

[1] A Cypherpunk's Maniphesto, http://www.activism.net/cypherpunk/manifesto.html, accessed on 2007/09/10.

[2] Abeille, J.; Aguiar, R. L.; Girao, J.; Melia, T.; Soto, I. & Stupar, P.: "MobiSplit in a virtualized, multi-device environment", IEEE International Conference on Communications, June 2007.

[3] Adkins, D.; Lakshminarayanan, K.; Perrig, A. & Stoica, I.: "Towards a more functional and secure network infrastructure", Report No. UCB/CSD-03-1242, UC Berkeley, 2003.

[4] Agrawal, D. & Aggarwal, C. C.: "On the design and quantification of privacy preserving data mining algorithms", Symposium on Principles of Database Systems, 2001.

[5] Akyildiz, I.; Jiang, X. & Mohanty, S.: "A survey of mobility management in next-generation all-IP-based wireless systems", IEEE Wireless Communications, Volume 11, pages 16-28, August 2004.

[6] Al-Muhtadi, J.; Campbell, R.; Kapadia, A.; Mickunas, M. D. & Yi, S.: "Routing through the mist: Privacy preserving communication in ubiquitous computing environments", ICDCS '02: Proceedings of the 22nd International Conference on Distributed Computing Systems, IEEE Computer Society, pages 74-83, 2002.

[7] Amer, S. H.; Humphries, J. W. & Hamilton, J. A.: "Survey: Security in the system development life cycle", Proceedings of the 2005 IEEE Workshop on Information Assurance, United States Military Academy, West Point, NY, June 2005.

[8] Anderson, R. J.: "Security engineering: A guide to building dependable distributed systems", John Wiley & Sons, Inc., 2001.

[9] Anonymous Proxy Servers, www.anonymizer.com, accessed on 2007/09/10.

[10] Armando, A.; Basin, D.; Boichut, Y.; Chevalier, Y.; Compagna, L.; Cuellar, J.; Drielsma, P. H.; Hem, P.; Kouchnarenko, O.; Mantovani, J.; Mdersheim, S.; von Oheimb, D.; Rusinowitch, M.; Santiago, J.; Turuani, M.; Vigan, L. & Vigneron, L.: "The Avispa tool for the automated validation of internet security protocols and applications", CAV 2005: Proceedings of Computer Aided Verification 2005, Springer Verlag, LNCS 3576, 2005.

[11] Ashby, V.; Jajodia, S.; Smith, G.; Wisseman, S. & Wichers, D.: "NCSC TECHNICAL REPORT - 005", National Computer Security Center, Number 1/5, May 1996.

[12] Askwith, B.; M.Merabti & Shi, Q.: "MNPA: A Mobile Network Privacy Architecture", Computer Communications, Volume 23, Number 18, pages 1777-1788, December 2000.

[13] Atiquzzaman, M. & Reaz, A.: "Survey and classification of transport layer mobility management schemes", PIMRC 2005: Proceedings of the 16th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2005, Volume 4, pages 2109-2215, September 2005.

[14] Backes, M.; Camenisch, J. & Sommer, D.: "Anonymous yet accountable access control", Proceedings of the 2005 ACM workshop on Privacy in the Electronic Society, pages 40-46, 2005.

[15] Batchvarov, A.: "Simulative Untersuchung von Anonymisierungsdiensten in mobilen IP-Netzen", Studienarbeit, Universitaet Stuttgart, December 2005.

[16] Basin, D.; Mdersheim, S. & Vigan, L.: "OFMC: A symbolic model checker for security protocols", International Journal of Information Security, Volume 4, Number 3, pages 181-208, 2005.

[17] Baskerville, R.: "Information systems security design methods: implications for information systems development" ACM Computing Surveys, ACM Press, Volume 25, Number 4, pages 375-414, 1993.

[18] Bellinger, G.; Castro, D., Mills, A: "Data, Information, Knowledge, and Wisdom", 2004, http://www.systems-thinking.org/dikw/dikw.htm, accessed on 2007/09/25.

[19] Benjelloun, O.; Garcia-Molina, H.; Jonas, J.; Su, Q.; Whang, S. E. & Widom, J.: "Swoosh: A generic approach to entity resolution", Stanford University, Technical Report Number 2005-5, 2005.

[20] Bennett, K. & Grothoff, C.: Dingledine, R. (ed.) "GAP - Practical Anonymous Networking", PET 2003: Proceedings of Privacy Enhancing Technologies Workshop, Springer-Verlag, LNCS 2760, pages 141-160, March 2003.

[21] Blakley, B. & Heath, C.: "Technical guide: Security design patterns", The Open Group, April 2004.

[22] Boehme, R.; Danezis, G.; Diaz, C.; Koepsell, S. & Pfitzmann, A.: "On the PET workshop panel 'Mix cascades versus peer-to-peer: Is one concept superior?'", PET2004: Proceedings of the Workshop on Privacy Enhancing Technologies 2004, Springer Verlag, LNCS 3424, May 2004.

[23] Borning, M.; Kesdogan, D. & Spaniol, O.: "Anonymitaet und Unbeobachtbarkeit im Internet", it+ti - Informationstechnik und Technische Informatik, Volume 43, Number 5, pages 246-253, 2001.

[24] Boucher, P.; Shostack, A. & Goldberg, I.: "Freedom system 2.0 architecture", 2000, http://www.cs.mcgill.ca/~splinter/Freedom_System_2_Architecture.pdf, accessed on 2007/09/10.

[25] British Standards Institute: "BS 7799: Code of practice for information security management", British Standards Institute, UK, 1993.

[26] Brodsky, A.; Farkas, C. & Jajodia, S.: "Secure databases: Constraints, inference channels, and monitoring disclosures", Knowledge and Data Engineering, Volume 12, Number 6, pages 900-919, 2000.

[27] Bronstein, I. & Semendjajew, K.: "Taschenbuch der Mathematik", Grosche, G.; Ziegler, V. & Ziegler, D. (ed.) B.G. Teubner Verlagsgesellschaft, 1991.

[28] Bundesamt fuer Sicherheit in der Informationstechnik: "IT-Grundschutz", http://www.bsi.de/gshb/ accessed on 2007/08/10.

[29] Burns, S. F.: "Threat modeling: A process to ensure application security", GIAC Security Essentials Certification (GSEC) Practical Assignment, January 2005.

[30] Burrows, M.; Abadi, M. & Needham, R.: "A logic of authentication", ACM Transactions on Computer Systems, ACM Press, Volume 8, Number 1, pages 18-36, 1990.

[31] Camenisch, J.; Hohenberger, S. & Lysyanskaya, A.: "Balancing accountability and privacy using e-cash", SCN 2006: Proceedings of the Security and Cryptography for Networks, September 2006.

[32] Campbell, A.T.; Gomez, J.; Sanghyo Kim; Chieh-Yih Wan; Turanyi, Z.R.; Valko, A.G.: "Comparison of IP micromobility protocols,", Wireless Communications, IEEE [see also IEEE Personal Communications], Volume 9, Number 1, pages 72-82, February 2002.

[33] Chaum, D. L.: "Showing credentials without identification: Transferring signatures between unconditionally unlinkable pseudonyms", AUSCRYPT '90: Proceedings of the International Conference on Cryptology on Advances in Cryptology, Springer Verlag New York, Inc., 246-264, 1990.

[34] Chaum, D. L.: "The Dining Cryptographers problem: Unconditional sender and recipient untraceability", Journal of Cryptology, Volume 1, Number 1, pages 65-75, 1988.

[35] Chaum, D. L.: "Untraceable electronic mail, return addresses, and digital pseudonyms", Communications of the ACM, ACM Press, Volume 24, Number 2, pages 84-90, 1981.

[36] Chiussi, F.M.; Khotimsky, D.A.; Krishnan, S.: "Mobility management in third-generation all-IP networks," Communications Magazine, IEEE, Volume 40, Number 9, pages 124-135, September 2002.

[37] Clarke, I.; Sandberg, O.; Wiley, B. & Hong, T. W.: "Freenet: A distributed anonymous information storage and retrieval system", Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability, pages 46-66, July 2000.

[38] Clauss, S.: "A framework for quantification of linkability within a privacy-enhancing identity management system.", ETRICS 06: Proceedings of the Emerging Trends in Information and Communication Security 2006, Springer Verlag, LNCS 3995, pages 191-205, 2006.

[39] Clauss, S. & Schiffner, S.: "Structuring anonymity metrics", DIM '06: Proceedings of the Second ACM Workshop on Digital Identity Management, ACM Press, pages 55-62, 2006.

[40] Christen, P.: "Privacy-preserving data linkage and deocoding: Current approaches and research directions", ICDMW '06: Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops, IEEE Computer Society, pages 497-501, 2006.

[41] Clark, D.: "Understanding", http://www.nwlink.com/~donclark/performance/understanding.html, 2004, accessed in November 2006.

[42] Common Criteria, http://www.commoncriteriaportal.org, accessed on 2007/07/24.

[43] "Common methodology for information technology security evaluation", Version 3.1, September 2006.

[44] Conta, A. & Deering, S.: "Generic packet tunneling in IPv6 specification", IETF, RFC2473, December 1998.

[45] Cvrcek, D. & Matyas, V.: "On the role of contectual information for privacy attacks and classification", IEEE ICDM: Proceedings of the IEEE Privacy and Security Aspects of Data Mining Workshop, Brighton, UK, November 2004.

[46] Cvrcek, D. & Matyas, V.: "Privacy - what do you mean?", Proceedings of the UBICOMP 2004 Privacy Workshop, pages 12-18, 2004.

[47] Danezis, G. & Diaz, C.: "A survey of anonymous communication channels", Journal of Privacy Technology, 2006.

[48] Danezis, G.; Dingledine, R. & Mathewson, N.: "Mixminion: Design of a Type III anonymous remailer protocol", Proceedings of the 2003 IEEE Symposium on Security and Privacy, pages 2-15, May 2003.

[49] Danezis, G. & Sassaman, L.: "Heartbeat traffic to counter (n-1) attacks: red-green-black mixes", WPES '03: Proceedings of the 2003 ACM Workshop on Privacy in the ElectronicSociety, ACM Press, pages 89-93, 2003.

[50] Public Act of the UK Parliament: "Data Protection Act 1998", http://www.opsi.gov.uk/ACTS/acts1998/19980029.htm, 1998.

[51] Deering, S. & Hinden, R.: "Internet Protocol, version 6 (IPv6) Specification", IETF, RFC2460, December 1998.

[52] Delugach, H. S. & Hinke, T. H.: "Using conceptual graphs to represent database inference security analysis", Journal of Computing and Information Technology (CIT), Volume 2, Number 4, pages 291-307, 1994.

[53] Delugach, H. S. & Hinke, T. H.: "Wizard: A database inference analysis and detection system", IEEE Transactions on Knowledge and Data Engineering, Volume 8, Number 1, pages 56-66, February 1996.

[54] Denning, D.E. & Morgenstern, M.: "Military database technology study: AI techniques for security and reliability", August 1986.

[55] Department of Trade and Industry, London, "Information technology security evaluation criteria (ITSEC)", June 1991, http://www.bsi.bund.de/zertifiz/itkrit/itsec-en.pdf, accessed on 2007/08/10.

[56] Devanbu, P. T. & Stubblebine, S.: "Software engineering for security: a roadmap" ICSE '00: Proceedings of the Conference on the Future of Software Engineering, ACM Press, pages 227-239, 2000.

[57] Diaz, C.; Naessens, V.; Nikova, S.; De Decker, B. & Preneel, B.: "Tools for technologies and applications of controlled anonymity", Anonymity and Privacy in Electronic Services, IWT APES deliverable 11, December 2004.

[58] Diaz, C.; Seys, S.; Claessens, J. & Preneel, B.: "Towards measuring anonymity", PET2002: Proceedings of the Workshop on Privacy Enhancing Technologies 2002, Springer Verlag, LNCS 2482, April 2002.

[59] Dingledine, R.; Mathewson, N. & Syverson, P.: "Tor: The second-generation Onion Router", Proceedings of the 13th USENIX Security Symposium, August 2004.

[60] Dingledine, R.; Shmatikov, V. & Syverson, P.: "Synchronous batching: From cascades to free Routes", PET2004: Proceedings of the Workshop on Privacy Enhancing Technologies 2004, Springer Verlag, LNCS 3424, 2004.

[61] Dreibholz, T.; Jungmaier, A. & Tuexen, M.: "A new scheme for IP-based Internet-mobility", LCN '03: Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks, IEEE Computer Society, pages 99, 2003.

[62] Droms, R.; Bound, J.; Volz, B.; Lemon, T.; Perkins, C. & Carney, M.: "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", July 2003.

[63] Eckert, C. & Marek, D.: "Developing secure applications: A systematic approach" SEC'97: SEC '97: Proceedings of the IFIP TC11 13 International Conference on Information Security in Research and Business, Chapman & Hall, Ltd., pages 267-279, 1997.

[64] Einsiedler, H.; Aguiar, R.; Jaehnert, J.; Jonas, K.; Liebsch, M.; Schmitz, R.; Pacyna, P.; Gozdecki, J.; Papir, Z.; Moreno, J. & Soto, I.: "The Moby Dick project: A mobile heterogeneous all-IP architecture", Advanced Technologies, Applications and Market Strategies for 3G - ATAMS 2001, 2001.

[65] Eliot, T.: "The Rock", Faber & Faber, 1934.

[66] Elmagarmid, A. K.; Ipeirotis, P. G. & Verykios, V. S.: "Duplicate record detection: A survey", IEEE Transactions on Knowledge and Data Engineering, IEEE Educational Activities Department, Volume 19, Number 1, pages 1-16, 2007.

[67] Elmasri, R. & Navathe, S.B.: "Fundamentals of database systems, 2nd Edition.", Benjamin/Cummings, 1994.

[68] Escudero, A.; Hedenfalk, M. & Heselius, P.: "Flying Freedom: Location privacy in mobile internetworking", INET2001: Proceedings of the Internet Society Conference 2001, June 2001.

[69] European Network and Information Security Agency, ENISA: "Risk management: implementation principles and inventories for risk management/risk assessment methods and tools", June 2006.

[70] European Network and Information Security Agency, ENISA, risk management section, http://enisa.europa.eu/rmra/h_home.html, accessed on 2007/08/13.

[71] Farha, R.; Khavari, K.; Abji, N. & Leon-Garcia, A.: "Peer-to-peer mobility management for all-IP networks", ICC '06: Proceedings of the IEEE International Conference on Communications 2006, Volume 5, 1946-1952, June 2006.

[72] Farkas, C.: "The inference problem in databases", PhD thesis, Information Technology, George Mason University, 2000.

[73] Farkas, C. & Jajodia, S.: "The inference problem: A survey", SIGKDD Explorations Newsletter, ACM Press, Volume 4, Number 2, pages 6-11, 2002.

[74] Fasbender, A.; Kesdogan, D. & Kubitz, O.: "Analysis of security and privacy in Mobile IP", Proceedings of the 4th International Conference on Telecommunication Systems Modeling and Analysis, March 1996.

[75] Federrath, H.: "Sicherheit mobiler Kommunikation", DuD-Fachbeitraege, Vieweg, 1999.

[76] Fernandez, E. B. & Pan, R.: "A pattern language for security models", PLoP 2001: Proceedings of the Conference on Pattern Languages of Programs, 2001.

[77] FIDIS (Future of Identity in the Information Society), http://www.fidis.net, accessed on 2007/08/22.

[78] Firesmith, D.G.: "Analyzing the security significance of system requirements", The 13th IEEE Requirements Engineering Conference 2005, 2005.

[79] Firesmith, D.G.: "Specifying reusable security requirements.", Journal of Object Technology, Volume 3, pages 61-75, 2004.

[80] Fischer, F.: "Weiterentwicklung und Aufbau eines Anonymisierungsdienstes fuer mobile IP Kommunikation", Studienarbeit, Universitaet Stuttgart, 2007.

[81] Freedman, M. J. & Morris, R.: "Tarzan: A peer-to-peer anonymizing network layer", CCS '02: Proceedings of the 9th ACM Conference on Computer and Communications Security, ACM Press, pages 193-206, 2002.

[82] Gabber, E.; Gibbons, P. B.; Kristol, D. M.; Matias, Y. & Mayer, A.: "Consistent, yet anonymous, web access with LPWA", Communications of the ACM, ACM Press, Volume 42, Number 2, pages 42-47, 1999.

[83] Garvey, T. D. & Lunt, T. F.: "Cover stories for database security", Results of the IFIP WG 11.3 Workshop on Database Security V, North-Holland Publishing Co., pages 363-380, 1992.

[84] Gloss, B. & Hauser, C.: "The IP micro mobility approach", Proceedings of the EUNICE 2000 Open European Summer School on Innovative Internet Applications, 2000.

[85] Goel, S.; Robson, M.; Polte, M. & Sirer, E. G.: "Herbivore: A scalable and efficient protocol for anonymous communication", Cornell University, Technical Report Number 2003-1890, February 2003.

[86] Goldschlag, D. M.; Reed, M. G. & Syverson, P. F.: "Hiding routing information", Proceedings of the First International Workshop on Information Hiding, Springer Verlag, pages 137-150, 1996.

[87] Golle, P. & Jakobsson, M.: "Reusable anonymous return channels", WPES '03: Proceedings of the 2003 ACM Workshop on Privacy in the Electronic Society, ACM Press, pages 94-100, 2003.

[88] Gong, L.; Needham, R. & Yahalom, R.: "Reasoning about belief in cryptographic protocols", Cooper, D. & Lunt, T. (ed.) Proceedings 1990 IEEE Symposium on Research in Security and Privacy, IEEE Computer Society, pages 234-248, 1990.

[89] Graft, D.; Pabrai, M. & Pabrai, U.: "Methodology for network security design", IEEE Communications Magazine, Volume 28, Number 11, pages 52-58, 1990.

[90] Gritzalis, S.; Spinellis, D. & Georgiadis, P.: "Security protocols over open networks and distributed systems: Formal methods for their analysis, design, and verification", Computer Communications, Volume 22, Number 8, Elsevier, pages 695-707, May 1999.

[91] Gruteser, M. & Grunwald, D.: "Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis", Mobile Networks and Applications, Kluwer Academic Publishers, Volume 10, Number 3, pages 315-325, 2005.

[92] Gu, L.; Baxter, R.; Vickers, D. & Rainsford, C.: "Record linkage: Current practice and future directions", CMIS Technical. Report No. 03/83, April 2003.

[93] Guan, Y.; Fu, X.; Bettati, R. & Zhao, W.: "An optimal strategy for anonymous communication protocols", Proceedings of the 22nd International Conference on Distributed Computing Systems, 2002, pages 257-266, 2002.

[94] Hafiz, M.: "A collection of privacy design patterns", PLoP 2006: Proceedings of the Conference on Pattern Languages of Programs, 2006.

[95] Hafiz, M. & Johnson, R. E.: "Security patterns and their classification schemes", Technical Report for Microsoft's Patterns and Practices Group, September 2006.

[96] Hale, J.; Shenoi, S.: "Catalytic inference analysis: Detecting inference threats due to knowledge discovery," Proceedings of the IEEE Symposium on Security and Privacy, pages 188-199, May 1997.

[97] Hale, J.; Threet, J. & Shenoi, S.: "A practical formalism for imprecise inference control", IFIP Transactions archive – Computer Science And Technology, Volume 60, pages 139-156, 1994.

[98] Harkins, D. & Carrel, D.: "The Internet Key Exchange (IKE)", IETF, RFC2409, November 1998.

[99] Hauser, C.: "A methodology for evaluating threats to multiple virtual identities", Beitraege zum Essener Workshop zur Netzwerksicherheit (EWNS) 2006, Essen, October 2006.

[100] Hauser, C.: "A new approach for privacy-preserving communication by combining virtual identities with mobility management", ESORICS 2002: European Symposium on Research in Computer Security, Zurich, October 2002.

[101] Hauser, C.: "Mobility management meets privacy - the failure of existing proposals and a new, future-proof approach", Proceedings of the ACM International Workshop on Mobility Management and Wireless Access, pages 122-124, October 2004.

[102] Hauser, C. & Kabatnik, M.: "Towards privacy support in a global location service", Proceedings of the IFIP Workshop on IP and ATM Traffic Management (WATM/ EUNICE 2001), pages 81-89, September 2001.

[103] Hauser, C.; Leonhardi, A. & Kuehn, P.J.: "Sicherheitsaspekte in NEXUS - einer Plattform fuer ortsbezogene Anwendungen", Informationstechnik und Technische Informatik (it+ti), Volume 44, Number 5, pages 268-277, October 2002.

[104] Helmers, S.: "A brief history of anon.penet.fi - The legendary anonymous remailer", CMC Magazine, 1997.

[105] Henderson, T. R.: "Host mobility for IP networks: a comparison", IEEE Network, Volume 17, 18-26, November-December 2003.

[106] Hernandez Orallo, J.: "Computational measures of information gain and reinforcement in inference processes", University of Valencia, September, 1999.

[107] Hey, J.: "The Data, Information, Knowledge, Wisdom chain: The metaphorical link", Intergovernmental Oceanographic Commission - OceanTeacher: A Training System for Ocean Data and Information Management, December 2004.

[108] Hinden, R. & Deering, S.: "IP Version 6 addressing architecture", IETF, RFC2373, July 1998.

[109] Hinke, T.H.: "Inference aggregation detection in database management systems", Proceedings of the 1998 IEEE Symposium on Security and Privacy, pages 96-106, 18-21 April 1988.

[110] Hinke, T. H.: "Response to research question 3 in research questions list, answers, and revision", in Landwehr, C. E. (Ed.), Database Security, III: Status and Prospects, Results of the IFIP Working Group 11.3 Workshop on Database Security, September 1989.

[111] Hoare, C. A. R.: "Communicating sequential processes", Commununications of the ACM, ACM Press, Volume 21, Number 8, pages 666-677, 1978.

[112] hostip.info—My IP Address Lookup and GeoTargeting Community Geotarget IP Project. What Country, City IP Addresses Map To., www.hostip.info, accessed on 2007/02/07.

[113] Hughes, D. & Shmatikov, V.: "Information hiding, anonymity and privacy: A modular approach", Journal of Computer Security, Volume 12, Number 1, pages 3-36, 2004.

[114] Iachello, G. & Abowd, G. D.: "Privacy and proportionality: adapting legal evaluation techniques to inform design in ubiquitous computing" CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press, pages 91-100, 2005.

[115] IEEE, "Guidelines for 64-bit global identifier (EUI-64) registration authority", http://standards.ieee.org/regauth/oui/tutorials/EUI64.html, July 2007.

[116] IKR Simulation Library, http://www.ikr.uni-stuttgart.de/Content/IKRSimLib/, accessed on 2007/07/19.

[117] Independent Centre for Privacy Protection: "Datenschutz-Guetesiegel", https://www.datenschutzzentrum.de/guetesiegel/index.htm accessed on 2007/08/10.

[118] IP2Location^{TM} - Bringing Geography to the Internet, www.ip2location.com, accessed on 2007/02/07.

[119] IPAddressGuide.com, www.ipaddressguide.com, accessed on 2007/02/07.

[120] International Organization for Standardization (ISO), "Information technology - Security techniques - Information security management systems - Overview and vocabulary", ISO/IEC 27000, http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=41933&scopelist=PROGRAMME, accessed on 2007/08/10.

[121] Irtenkauf, T.: "Entwurf eines Pseudonymisierungsdienstes mit Mobilitaetsverwaltung", Diplomarbeit, Universitaet Stuttgart, 2002.

[122] IST-DAIDALOS, www.ist-daidalos.org, accessed on 2007/07/18.

[123] Jaehnert, J.; Zhou, J.; Aguiar, R. L.; Marques, V.; Wetterwald, M.; Melin, E.; Moreno, J. I.; Cuevas, A.; Liebsch, M.; Schmitz, R.; Pacyna, P.; Melia, T.; Kurtansky, P.; Hasan; Singh, D.; Zander, S.; Einsiedler, H. J. & Stiller, B.: "The 'pure-IP' Moby Dick 4G architecture", Computer Communications, Volume 28, Elsevier, pages 1014-1027, June 2005.

[124] Jajodia, S. & Meadows, C.: "Inference problems in multilevel secure database management systems", Abrams, M. D.; Jajodia, S. & Podell, H. J. (ed.) Information Security: An Integrated Collection of Essays, IEEE Computer Society Press, 1995.

[125] JAP, Anonymity and Privacy, http://anon.inf.tu-dresden.de, accessed on 2007/09/25.

[126] Johnson, D.B.; Perkins, C. & Arkko, J.: "Mobility support in IPv6", IETF, RFC3775, June 2004.

[127] Juerjens, J.: "Secure systems development with UML", Springer Academic Publishers, 2004.

[128] Juerjens, J.: "UMLsec: Extending UML for secure systems developments", UML 2002 - The Unified Modeling Language, pages 412-425, 2002.

[129] Kaschub, M.: "Modellierung eines Anonymisierungsdienstes und Implementierung eines Simulationswerkzeugs zu dessn Leistungsbewertung", Studienarbeit, Universitaet Stuttgart, 2005.

[130] Katti, S.; Cohen, J. & Katabi, D.: "Information slicing: Anonymity using unreliable overlays", Proceedings of the 4th USENIX Symposium on Network Systems Design and Implementation (NSDI), April 2007.

[131] Kemper, A. & Eickler, A.: "Datenbanksysteme - Eine Einfuehrung, 5. Auflage", Oldenbourg, 2004.

[132] Kent, S. & Seo, K.: "Security architecture for the Internet Protocol", IETF, RFC4201, December 2005.

[133] Kesdogan, D.; Egner, J. & Bueschkes, R.: "Stop-and-Go-MIXes: Providing probabilistic anonymity in an open system", Proceedings of the Second International Workshop on Information Hiding, Springer Verlag, pages 83-98, 1998.

[134] Kesdogan, D.; Reichl, P. & Junghaertchen, K.: "Distributed temporary pseudonyms: A new approach for protecting location information in mobile communication networks", ESORICS '98: Proceedings of the 5th European Symposium on Research in Computer Security, Springer Verlag, pages 295-312, 1998.

[135] Kohlweiss, M. & Rannenberg, K.: "Overview of existing assurance methods in the area of privacy and IT security", Public deliverable D5.1.a of the PRIME project, www.prime-project.eu, accessed on 2007/08/10, September 2004.

[136] Kuehn, P.J.: "Lecture notes: Communication Networks I", University of Stuttgart, 2006.

[137] Langheinrich, M.: "Privacy by design - principles of privacy-aware ubiquitous systems", UbiComp '01: Proceedings of the 3rd International Conference on Ubiquitous Computing, Springer Verlag, pages 273-291, September-October 2001.

[138] Laverdiere, M.; Mourad, A.; Hanna, A. & Debbabi, M.: "Security design patterns: Survey and evaluation", CCECE '06: Proceedings of the Canadian Conference on Electrical and Computer Engineering, pages 1605-1608, May 2006.

[139] Law, A.M., Kelton, W.D.: "Simulation Modeling & Analysis", 2nd edition, McGraw-Hill, 1991.

[140] Lawlor, B. & Vu, L.: "A survey of techniques for security architecture analysis", Technical Report DSTO-TR-1438, DSTO Information Sciences Laboratory, Information Networks Division, Edinburgh South Australia 5111, Australia, May 2003.

[141] Leung, Y.Y. & Lee, D.L.: "Logic approaches for deductive databases", IEEE Expert [see also IEEE Intelligent Systems and Their Applications], Volume 3, pages 64-75, Winter 1988.

[142] Levine, B. N. & Shields, C.: "Hordes - A multicast based protocol for anonymity", Journal of Computer Security, Volume 10, Number 3, pages 213-240, 2002.

[143] Lopatik, T.; Eckert, C. & Baumgarten, U.: "MMIP - Mixed Mobile Internet Protocol", CMS'97: Proceedings of the Conference on Communications and Multimedia Security 1997, pages 77-88, September 1997.

[144] Lunt, T.F.: "Aggregation and inference: facts and fallacies", Proceedings of the 1989 IEEE Symposium on Security and Privacy, pages 102-109, 1-3 May 1989.

[145] Machanavajjhala, A.; Kifer, D.; Gehrke, J. & Venkitasubramaniam, M.: "L-diversity: Privacy beyond k-anonymity", ACM Transactions on Knowledge Discovery from Data, ACM Press, Volume 1, Number 1, pages 3, 2007.

[146] Malin, B.: "Trail re-identification and unlinkability in distributed databases", PhD thesis, Institute for Software Research, International School of Computer Science, Carnegie Mellon University, 2006.

[147] Malin, B. & Sweeney, L.: "ENRES: A semantic framework for entity resolution modelling", Technical Report CMU-ISRI-05-134, Data Privacy Laboratory, Institute for Software Research, International School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, November 2005.

[148] Marias, G. F.; Delakouridis, C.; Kazatzopoulos, L. & Georgiadis, P.: "Location privacy through secret sharing techniques", WOWMOM '05: Proceedings of the First International IEEE WoWMoM Workshop on Trust, Security and Privacy for Ubiquitous Computing, IEEE Computer Society, pages 614-620, 2005.

[149] Alfredo Matos, J. G.; Sargento, S. & Aguiar, R. L.: "Preserving privacy in mobile environments with virtual network stacks", Globecom 2007: 50th Annual IEEE Global Telecommunications Conference, November 2007.

[150] Matos, A.; Santos, J.; Sargento, S.; Aguiar, R.; Girao, J. & Liebsch, M.: "HIP location privacy framework", First ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture, December 2006.

[151] Maurer, U. M.: "Modelling a public-key infrastructure", ESORICS '96: Proceedings of the 4th European Symposium on Research in Computer Security, Springer-Verlag, pages 325-350, 1996.

[152] Meadows, C.: "Formal verification of cryptographic protocols: A survey", ASIACRYPT '94: Proceedings of the 4th International Conference on the Theory and Applications of Cryptology, Springer-Verlag, pages 135-150, 1995.

[153] Millett, L. & Holden, S.: "Authentication and its privacy effects", IEEE Internet Computing, Volume 7, Number 6, pages 54-58, November-December 2003.

[154] Milner, R.: "A calculus of communicating systems", Springer Verlag, Lecture Notes in Computer Science, Volume 92, 1980.

[155] Millner, R.: "Communicating and mobile systems: the Pi-Calculus", Cambridge University Press, 1999.

[156] Mislove, A.; Oberoi, G.; Post, A.; Reis, C.; Druschel, P. & Wallach, D. S.: "AP3: Cooperative, decentralized anonymous dommunication", EW11: Proceedings of the 11th Workshop on ACM SIGOPS European Workshop: Beyond the PC, ACM Press, pages 30, 2004.

[157] Mixmaster, http://mixmaster.sourceforge.net, accessed on 2007/09/10.

[158] Morgenstern, M.: "Controlling logical inference in multilevel database systems", Proceedings of the 1988 IEEE Symposium on Security and Privacy, pages 245-255, April 1988.

[159] Morgenstern, M.: "Security and inference in multilevel database and knowledge-base systems", SIGMOD '87: Proceedings of the 1987 ACM SIGMOD International Conference on Management of Data, ACM Press, pages 357-373, May 1987.

[160] Moskowitz, R. & Nikander, P.: "Host Identity Protocol (HIP) architecture", IETF, RFC4423, May 2006.

[161] Naessens, V.: "A methodology for anonymity control in electronic services using credentials", PhD thesis, Katholieke Universiteit Leuven, Faculteit Ingenieurswetenschappen, Departement Computerwetenschappen, 2006.

[162] Naessens, V. & De Decker, B.: "A methodology for designing controlled anonymous applications", Proceedings of the IFIP SEC 2006, May 2006.

[163] Narten, T.; Nordmark, E. & Simpson, W.: "Neighbor discovery for IP version 6 (IPv6)", IETF, RFC2461, December 1998.

[164] Narten, T. & Draves, R.: "Privacy extensions for stateless address autoconfiguration in IPv6", IETF, RFC3041, January 2001.

[165] Nelson, B.; Choi, R.; Iacobucci, M.; Mitchell, M. & Gagnon, G.: "Cyberterror - prospects and implications", White Paper, Defense Intelligence Agency Office for Counterterrorism Analysis (TWC-1), October 1999.

[166] Network-based Localized Mobility Management (netlmm), http://www.ietf.org/html.charters/netlmm-charter.html, accessed on 2007/09/10.

[167] Neubauer, M.: "Untersuchung und Bewertung von Architekturen zur Verhinderung der Verkettung virtueller Identitaeten durch mobile Kommunikation, Diplomarbeit, Universitaet Stuttgart, 2003.

[168] Nickols, F.W.: "The knowledge in knowledge management", Cortada, J. & Woods, J. (ed.), The knowledge management yearbook 2000-2001, 2000.

[169] Nikander, P.; Arkko, J. & Ohlman, B.: "Host Identity Indirection Infrastructure (Hi3)", The Second Swedish National Computer Networking Workshop, November 2004.

[170] Older, S. & Chin, S.: "Formal methods for assuring security of protocols", The Computer Journal, Volume 45, Number 1, Oxford University Press, pages 46-54, 2002.

[171] Orava, P.; Haverinen, H.; Honkanen, J. & Edney, J.: "Temporary MAC addresses for anonymity", submission to IEEE P802.11, http://grouper.ieee.org/groups/802/11/Documents/D2T251-300.html, accessed on 2007/08/15, May 2002.

[172] Perkins, C. E.: "Mobile IP", IEEE Communications Magazine, Volume 35, Number 5, May 1997.

[173] Pfitzmann, A.: "A switched/broadcast ISDN to decrease user Oobservability", Proceedings of the 1984 IEEE International Zurich Seminar on Digital Communications, pages 183-190, 1984.

[174] Pfitzmann, A.: "Security in IT networks: Multilateral security in distributed and by distributed systems", Lecture Notes, Technische Universitaet Dresden, October 2006.

[175] Pfitzmann, A. & Hansen, M.: "Anonymity, unlinkablity, unobservability, pseudonymity, and identity management - A consolidated proposal for terminology", May 2006.

[176] Pfitzmann, A. & Waidner, M.: "Networks without user observability", Computers and Security, Elsevier Advanced Technology Publications, Volume 6, Number 2, pages 158-166, 1987.

[177] Plummer, D.C.: "An Ethernet address resolution protocol", IETF, RFC826, November 1982.

[178] Pothamsetty, V. & Balinsky, A.: "A structured and practical methodology for security evaluation of an IP based stack", Cisco, Critical Infrastructure Assurance Group (CIAG), http://www.cisco.com/web/about/security/security_services/ciag/documents/stack-howto.pdf, accessed on 2007/08/14, June 2003.

[179] Preneel, B.; Rompay, B. V.; Quiquater, J. & Massias, H.: "Evaluation methodology for security primitives", Technical report, TIMESEC Project (Federal Government Project, Belgium), 1997.

[180] PRIME - Privacy and Identity Management for Europe, https://www.prime-project.eu, accessed on 2007/08/22.

[181] Privacy Preserving Data Mining Bibliography, http://www.cs.umbc.edu/~kunliu1/research/privacy_review.html, accessed on 2007/07/24.

[182] Qian, X.; Stickel, M. E.; Karp, P. D.; Lunt, T. F. & Carvey, T. D.: "Detection and elimination of inference channels in multilevel relational database systems", Proceedings of the 1993 IEEE Symposium on Security and Privacy, IEEE Computer Society, 1993.

[183] Ramamohanarao, K. & Harland, J.: "An introduction to deductive database languages and systems", The VLDB Journal, Springer Verlag New York, Inc., Volume 3, pages 107-122, 1994.

[184] Reiter, M. K. & Rubin, A. D.: "Crowds: anonymity for web transactions", ACM Transactions on Information and System Security, Volume 1, Number 1, pages 66-92, 1998.

[185] Rennhard, M. & Plattner, B.: "Introducing MorphMix: Peer-to-peer based anonymous internet usage with collusion detection", WPES 2002: Proceedings of the Workshop on Privacy in the Electronic Society, November 2002.

[186] Romanosky, S.: "Enterprise security patterns", Information Systems Security Association Journal, March 2003.

[187] Romanosky, S.; Acquisti, A.; Hong, J.; Cranor, L. F. & Friedman, B.: "Privacy patterns for online interactions", PLoP 2006: Proccedings of the Conference on Pattern Languages of Programs, 2006.

[188] Sadicoff, M.; Larrondo-Petrie, M. M. & Fernandez, E. B.: "Privacy-aware network client pattern", PLoP 2005: Proceedings of the Conference on Pattern Languages of Programs, 2005.

[189] Saha, D.; Mukherjee, A.; Misra, I.; Chakraborty, M. & Subhash, N.: "Mobility support in IP: A survey of related protocols", Network, IEEE, Volume 18, pages 34-40, 2004.

[190] Sampigethaya, K. & Poovendran, R.: "A survey on mix networks and their secure applications", Proceedings of the IEEE, Volume 94, Number 12, December 2006.

[191] Schneider, S. & Sidiropoulos, A.: "CSP and anonymity", ESORICS '96: Proceedings of the 4th European Symposium on Research in Computer Security, Springer Verlag, pages 198-218, 1996.

[192] Schneier, B.: "Applied Cryptography", John Wiley & Sons, 1996.

[193] Schulzrinne, H. & Wedlund, E.: "Application-layer mobility using SIP", Mobile Computer Communication Review, Volume 4, pages 47-57, 2000.

[194] Schuemmer, T.: "The public privacy - patterns for filtering personal information in Collaborative Systems", CHI2004: Proceedings of the Conference on Human Factors in Computing Systems, 2004.

[195] Schumacher, M.: Goos, G.; Hartmanis, J. & van Leeuwen, J. (ed.) "Security engineering with patterns - origins, theoretical model, and new applications", Springer Verlag, LNCS 2754, 2003.

[196] Schumacher, M.: "Security patterns and security standards - with selected security patterns for anonymity and privacy", EuroPLoP 2002: Proceedings of the 7th European Conference on Pattern Languages of Programs, 2002.

[197] Schumacher, M.; Fernandez-Buglioni, E.; Hybertson, D.; Buschmann, F. & Sommerlad, P.: "Security patterns - integrating security and systems engineering", John Wiley and Sons, 2005.

[198] Schumacher, M. & Roedig, U.: "Security engineering with patterns", PLoP 2001: Proceedings of the Conference on Pattern Languages of Programs, 2001.

[199] Schweigel, M.: "Beitraege zur Berechnung bewegungsabhaengiger Kenngroessen von Mobilfunknetzen", Dissertation, Technische Universitaet Dresden, 2005.

[200] Serjantov, A. & Danezis, G.: "Towards an information theoretic metric for anonymity", PET2002: Proceedings of the Workshop on Privacy Enhancing Technologies 2002, Springer Verlag, LNCS 2482, April 2002.

[201] SFB 627: Nexus, Umgebungsmodelle fuer Mobile Kontextbezogene Systeme, www.nexus.uni-stuttgart.de, accessed on 2007/08/22.

[202] Sharma, N.: "The Origin of the "Data Information Knowledge Wisdom" Hierarchy", December, 2005.

[203] Sherwood, R.; Bhattacharjee, B. & Srinivasan, A.: "P5: A protocol for scalable anonymous communication", Proceedings of the 2002 IEEE Symposium on Security and Privacy, pages 58-70, 2002.

[204] Shunichi, T.: "CS381 discrete structures web course material", http://www.cs.odu.edu/~toida/nerzic/content/web_course.html, accessed in November 2006.

[205] Smith, G.: "Modeling security-relevant data semantics", IEEE Transactions on Software Engineering, Volume 17, Number 11, pages 1195-1203, November 1991.

[206] Smith, G.: "Modeling security-relevant data semantics", Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy, pages 384-391, May 1990.

[207] Soliman, H.; Castelluccia, C.; Malki, K. E. & Bellier, L.: "Hierarchical Mobile IPv6 mobility management (HMIPv6)", RFC4140, August 2005.

[208] Stallings, W.: "Cryptography and network security", Prentice Hall, 2006.

[209] Stallings, W.: "Network security essentials: Applications and standards", Prentice Hall, 2007.

[210] Steel, C.; Nagappan, R. & Lai, R.: "Core security patterns", Prentice Hall, 2006.

[211] Steinbrecher, S. & Koepsell, S.: "Modelling unlinkability", PET2003: Proceedings of the Workshop on Privacy Enhancing Technologies 2003, Springer Verlag, LNCS 2760, 2003.

[212] Stoica, I.; Adkins, D.; Zhuang, S.; Shenker, S. & Surana, S.: "Internet Indirection Infrastructure", IEEE/ACM Transactions on Networking, Volume 12, Number 2, pages 205-218, April 2004.

[213] Stoneburner, G.; Hayden, C. & Feringa, A.: "Engineering principles for information technology security (a baseline for achieving security)", NIST Special Publication 800-27, National Institute of Standards and Technology, June 2001.

[214] Su, T. A. & Ozsoyoglu, G.: "Controlling FD and MVD inferences in multilevel relational database systems", IEEE Transactions on Knowledge and Data Engineering, IEEE Educational Activities Department, Volume 3, Number 4, pages 474-485, 1991.

[215] Sweeney, L.: "k-Anonymity: A model for protecting privacy", International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems, Volume 10, Number 5, pages 557-570, 2002.

[216] Syverson, P. F. & Stubblebine, S. G.: "Group principals and the formalization of anonymity", World Congress on Formal Methods (1), pages 814-833, 1999.

[217] The Liberty Alliance, http://www.projectliberty.org accessed on 2007/08/17.

[218] Thomson, S. and Narten, T.: "IPv6 address autoconfiguration", IETF, RFC2462, December 1998.

[219] Thuraisingham, B.: "A primer for understanding and applying data mining", IT Professional, Volume 2, pages 28-31, January-February 2000.

[220] Thuraisingham, B. M.: "Current status of R&D in trusted database management systems.", SIGMOD Record, Volume 21, Number 3, pages 44-50, 1992.

[221] Toth, G.; Hornak, Z. & Vajda, F.: Liimatainen, S. & Virtanen, T. (ed.) "Measuring anonymity revisited", Proceedings of the Ninth Nordic Workshop on Secure IT Systems, Espoo, Finland, pages 85-90, November 2004.

[222] USA Department of Defense, "Trusted computer system evaluation criteria", DOD, 5200.28-STD, December 1985.

[223] van Herreweghen, E.: "Unidentifiability and accountability in electronic transactions", PhD thesis, Katholieke Universiteit Leuven, Faculteit Toegepaste Wetenschappen, Departement Computerwetenschappen, 2004.

[224] Varney, C.; Cole, P.; Duserick, W.; Lesser, J.; Podorowsky, G.; Sibieta, P. & Thornby, C.: Liberty Alliance: Privacy and Security Best Practices November 2003.

[225] Verykios, V.; Bertino, E.; Fovino, I.; Provenza, L.; Saygin, Y. & Theodoridis, Y.: "State-of-the-art in privacy preserving data mining", ACM SIGMOD Record, Volume 3, Number 1, pages 50-57, March 2004.

[226] Wang, G.; Xu, J. & Chen, H.: "Using social contextual information to match criminal identities", HICSS '06: Proceedings of the 39th Annual Hawaii International Conference on System Sciences 2006, Volume 4, January 2006.

[227] Waters, B. R.; Felten, E. W. & Sahai, A.: "Receiver anonymity via incomparable public keys", CCS '03: Proceedings of the 10th ACM Conference on Computer and Communications Security, ACM Press, pages 112-121, 2003.

[228] Winkler, W. E.: "Overview of record linkage and current research directions", Statistical Research Division, U.S. Census Bureau, Number Statistics 2006-2, February 2006.

[229] Wright, J.; Stepney, S.; Clark, J. A. & Jacob, J.: "Formalizing anonymity: A review", Department of Computer Science, University of York, Technical Report YCS-2005-389, June 2005.

[230] Xu, H.; Fu, X.; Zhu, Y.; Bettati, R.; Chen, J. & Zhao, W.: "SAS: A scalar anonymous communication system", ICCNMC 2005: Proceedings of the 3rd International Conference of Networking and Mobile Computing, pages 452-461, 2005.

[231] Yip, R. & Levitt, E.: "Data level inference detection in database systems", Proceedings of the 11th IEEE Computer Security Foundations Workshop, pages 179-189, June 1998.

[232] Yoder, J. & Barcalow, J.: "Architectural patterns for enabling application security", PLoP 1997: Proceedings of the Conference on Pattern Languages of Programs, 1997.

[233] Yu, C.H.: "Abduction? Deduction? Induction? Is there a logic of exploratory data analysis?", Annual Meeting of American Educational Research Association, New Orleans, Louisiana, April 1994.

[234] Zhuang, L.; Zhou, F.; Zhao, B. Y. & Rowstron, A.: "Cashmere: Resilient anonymous routing", NSDI2005: Procedings of the Symposium on Networked Systems Design and Implementation, 2005.

[235] Zhuang, S.; Lai, K.; Stoica, I.; Katz, R. & Shenker, S.: "Host mobility using an internet indirection infrastructure", MobiSys'03: Proceedings of the First International Conference on Mobile Systems, Application, and Services, May 2003.

[236] Zugenmaier, A.: "Anonymity for users of mobile devices through location addressing", Dissertation, Fakultaet fuer Angewandte Wissenschaften der Albert-Ludwigs-Universitaet Freiburg im Breisgau, October 2002.

[237] Zugenmaier, A.: "FLASCHE - A mechanism providing anonymity for mobile users", PET2004: Proceedings of the Workshop on Privacy Enhancing Technologies 2004, Springer Verlag, LNCS 3424, May 2004.

[238] Zugenmaier, A.: "The Freiburg privacy diamond", Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE, Volume 3, pages 1501-1505, volume 3, 1-5 December 2003