

Diss. ETH No. 13156

Dynamic Security in Communication Systems

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY ZÜRICH

for the degree of
Doctor of Technical Sciences

presented by

GERMANO CARONNI

Dipl. Informatik-Ing. ETH
born September 7, 1967
citizen of Mendrisio (TI)

accepted on the recommendation of
Prof. Dr. Bernhard Plattner, examiner
Prof. Dr. Ueli Maurer, co-examiner

1999

Verlagstitelseite

CIP-Eintragung und Copyright (Verlag)

Abstract

The importance of communication security is increasing, because more and more valuable information is being transferred over computer networks. As of now, the provision of security (namely confidentiality, integrity, and authenticity) is an all-or-nothing issue: security is either provided to the maximum extent possible, or not at all. Offering security can be very expensive in terms of ease of use, management requirements, and computing overhead. As a consequence, security is often regarded as bad, not worth the benefits it provides. Currently, there are no mechanisms to fine-tune the strength of offered security and have applications use just the right amount of security to deter attackers.

In this thesis, dynamic aspects of security are explored. This covers Quality of Service (QoS) models and requirements for security, secure multimedia protocols, and peer and component authentication. Each of these issues is examined, and its dynamic properties are shown. The experimental platforms Da CaPo and WaveVideo are used to prototype some of the results. In essence, it is shown that it is possible to provide fine-grained, scalable security to applications and allow them to select at runtime the required amount of processing overhead necessary to achieve sufficient security.

The work first demonstrates the need for dynamic security and defines and discusses the fundamental properties of different aspects of security. These include the available cryptographic mechanisms and their properties, and where to place security functionality in a communication system. In conclusion, a coarse-grained system model is proposed. The then following examination of the state of the art clearly shows that the concept of merging security and QoS is novel. The same is true for protocols that provide a dynamically configurable amount of security, and for dynamically composable peer-authentication protocols with different properties.

In the next part, the focus is placed on QoS and security: the goal is to model dedicated security QoS parameters and to integrate them with

the traditional view of QoS. To this end, the new parameters are quantified by establishing a relationship to the monetary value of the data that are to be protected. Confidentiality and authenticity are then modelled from the user's perspective, and the mapping and translation down to the communication infrastructure is presented. The QoS parameters are used to select protocol and algorithm properties. Additionally, attack costs are evaluated, and the zero-cost attack is presented.

The ability to specify security requirements via QoS parameters allows the extension of multi-media protocols to provide dynamically configurable security functionality. In contrast to traditional solutions, in which security is provided on a multimedia data stream as a whole, dedicated security mechanisms that are embedded in the rest of the multimedia protocol allow very efficient processing. They receive QoS parameters to control their operations, and then provide, besides their more traditional tasks, confidentiality and authenticity for the transferred data. By varying the employed algorithms and the actual data coverage of the algorithm, computing power consumption can be reduced, while still maintaining an adequate degree of security. This property is critical when real-time behaviour is needed in software-only solutions. It is studied in the context of general purpose, audio, and video data. Results derived from a prototype are presented.

To conclude the thesis, the implementation of a framework capable of integrating the new functionalities is presented. For this, the existing QoS-capable middleware Da CaPo is extended. Securing Da CaPo communications is achieved by defining protocols that include encrypting and authenticating modules. Depending on the security requirements that the application specifies, the configuration process can employ these modules. A static key and certificate database allow for the storage and recovery of public keys and related information. The resulting functionality and performance are then evaluated, showing that fine-grained control over security is feasible, and the resulting performance is sufficient for real-world applications.

Zusammenfassung

Da mehr und mehr wertvolle Information über Computernetze übertragen wird, nimmt die Wichtigkeit von sicherer Kommunikation zu. Zur Zeit ist Sicherheit (hier vor allem Vertraulichkeit, Integrität und Authentizität) eine Alles-oder-Nichts Sache: Entweder wird Sicherheit in maximaler Stärke geboten, oder sie fehlt gänzlich. Benutzerfreundlichkeit, Administrierbarkeit und verfügbare Rechenleistung nehmen ab, wenn Sicherheit zur Verfügung gestellt wird. Als eine Folge dieser Nebenwirkungen wird Sicherheit häufig abgelehnt, da sie als unverhältnismässig aufwendig erscheint. Es gibt keine Mechanismen um eine Feinabstimmung der gebotenen Sicherheit zu erreichen. Eine Anwendung kann die Kommunikationsinfrastruktur nicht so einstellen, dass der gerade noch benötigte Overhead getrieben wird der nötig ist um Angreifer abzuschrecken.

In dieser Doktorarbeit werden dynamische Aspekte der Sicherheit erkundet. Dies deckt Dienstgütemodelle und Sicherheitsanforderungen, sichere Multimedia-Protokolle, sowie die Authentisierung von Kommunikationspartnern und Systemkomponenten ab. Jedes dieser Themen wird untersucht und die dynamischen Eigenschaften werden beleuchtet. Die experimentellen Plattformen Da CaPo++ und WaveVideo werden verwendet, um einige der Resultate prototypisch zu implementieren. Im Wesentlichen wird gezeigt, dass es möglich ist, Anwendungen fein abgestufte Sicherheit zugänglich zu machen. Dies erlaubt es ihnen, zur Laufzeit zu bestimmen, welches Mass an Rechenleistung nötig ist, um Daten ausreichend zu schützen.

Zu Beginn der Arbeit wird der Bedarf für dynamische Sicherheit klargestellt, und die grundlegenden Eigenschaften verschiedener Sicherheitsaspekte werden definiert und diskutiert. Dies schliesst die verfügbaren kryptographischen Mechanismen und ihre Eigenschaften ein, und die Frage wo Sicherheitsfunktionen in einem Kommunikationssystem integriert werden sollen. Als Schlussfolgerung wird ein grobkörniges Systemmodell vorgestellt. Die darauffolgende Untersuchung des

Standes der bisherigen Forschung auf diesem Gebiet zeigt klar auf, dass das Konzept der Verschmelzung von Sicherheit und Dienstgütemodellen neu ist. Das Gleiche gilt für Protokolle mit dynamisch konfigurierbarer Sicherheitsstärke, und für die Möglichkeit, zur Laufzeit dynamisch Authentisierungsprotokolle aus verschiedenen Modulen mit unterschiedlichen Eigenschaften zusammenzusetzen.

Anschliessend konzentriert sich die Arbeit auf die Aspekte der Dienstgüte und Sicherheit. Das Ziel ist hier eine Modellierung dedizierter Sicherheits-Dienstgüteparameter, und ihre Verschmelzung mit der bisherigen Sichtweise von Dienstgüte in Kommunikationssystemen. Um dies zu erreichen werden die Parameter quantisiert, indem ein Bezug zum Geldwert der zu schützenden Informationen geschaffen wird. Dann werden Vertraulichkeit und Authentizität aus Sicht des Benutzers modelliert und ihre Uebersetzung und Abbildung bis hinunter zur Kommunikationsinfrastruktur wird verfolgt. Die Dienstgüteparameter werden verwendet um die Eigenschaften von Protokollen und Algorithmen zu wählen. Zusätzlich werden die Kosten von Angriffen evaluiert, und die Nullkostenattacke wird präsentiert.

Die Möglichkeit, Sicherheitsanforderungen über QoS Parameter zu spezifizieren erlaubt die Erweiterung von Multimedia-Protokollen um dynamisch konfigurierbare Sicherheit. Im Gegensatz zu traditionellen Lösungen, in denen Sicherheit auf einen Datenstrom als Ganzes angewandt wird, ist es dank der Integration dedizierter Sicherheitsmechanismen in das Protokoll möglich, eine sehr effiziente Datenverarbeitung zu gewährleisten. Die Sicherheitsmechanismen erhalten konkretisierte Dienstgüteparameter zur Steuerung ihres Verhaltens, und müssen neben ihren sonstigen Aufgaben Vertraulichkeit und Authentizität übertragener Daten gewährleisten. Durch Veränderung der verwendeten Algorithmen, und durch Variation der tatsächlichen Abdeckung von Daten durch den Algorithmus kann die benötigte Rechenleistung reduziert werden. Dabei wird nach wie vor ein angemessenes Mass an Sicherheit aufrechterhalten. Diese Eigenschaft ist unumgänglich wenn Echtzeitverhalten in reinen Softwarelösungen erreicht werden soll, und sie wird im Zusammenhang mit kontinuierlichen Datenströmen wie Audio und Video näher untersucht. Daraus abgeleitete Resultate werden präsentiert.

Zum Abschluss der Arbeit wird die Implementation einer Systemlösung präsentiert, die diese neuen Funktionen unterstützen kann. Um

dies zu erreichen wird die vorhandene Systemlösung Da CaPo, die von sich aus Dienstgüte unterstützt, erweitert. Die Sicherung von Kommunikationsbeziehungen in Da CaPo wird erzielt indem Protokolle definiert werden die Authentisierungs- und Verschlüsselungsmodule beinhalten. Abhängig von den Sicherheitsanforderungen wie sie durch die Anwendung spezifizierbar sind kann der Konfigurationsprozess diese Module verwenden. Eine statische Schlüssel- und Zertifikatsdatenbank gestatten das anwendungsunabhängige Speichern und Laden öffentlicher Schlüssel und ähnlicher Informationen. Die entstandene Funktionalität und Leistungsfähigkeit wird evaluiert. Die Evaluation zeigt, dass eine feinkörnige Kontrolle von Sicherheitseigenschaften machbar ist, und dass die resultierende Leistungsfähigkeit ausreichend ist, um reale Anwendungen möglich zu machen.

TO EVE
BUTTERFLY WINGS, CATISH GRIN,
AND ETERNAL GUIDING LIGHT.

Table of Contents

Abstract	v
Zusammenfassung	vii
Table of Contents	xi
List of Figures	xvii
List of Tables	xix
1 Introduction	21
1.1 Motivation	23
1.2 Current Weaknesses and New Demands	24
1.3 Sample Scenario	26
1.4 Claims	33
1.5 Cryptography and Politics	35
1.6 Outline	36
2 Foundations	37
2.1 Defining Security	38
2.2 Secure Computing Systems	42
2.3 The Basics of Secure Communication	45
2.3.1 Key Agreement Mechanisms	47
2.3.2 Encryption Mechanisms	48
2.3.3 Integrity Assurance and Authentication	51
2.3.4 Cryptographic Protocols and Attacks	54
2.3.5 Example of a Secured Protocol	56
2.4 Security in a Communication System	58

2.4.1	Security and the Layered Model of Communication Systems	60
2.4.2	Properties of Security in the Application Layer	61
2.4.3	Properties of Security in the Communication Layer	62
2.4.4	Properties of Security in the Transport Infrastructure (Hardware)	63
2.5	System Model for the Placement of Security	64
2.6	Summary	66
3	Related Work	67
3.1	Quality of Service (QoS)	70
3.1.1	QoS Standardisation Efforts	71
3.1.2	Communication Systems providing QoS Support	72
3.2	Frameworks for Secure Communication.	76
3.2.1	Relevant Security Standards	77
3.2.2	Projects and Implementations	79
3.3	Dedicated Secure Multimedia Protocols	82
3.3.1	Video Communication— Security and Quality Issues	83
3.3.2	MPEG Audio Encoding.	83
3.3.3	Video Encryption Methods for MPEG	83
3.4	Authentication and Trust Frameworks	84
3.4.1	Expressing Trust in Peers	85
3.4.2	Peer Authentication	85
3.5	Comparison and Evaluation	88
3.5.1	QoS Systems and Secure Communication.	88
3.5.2	Traditional Communication Frameworks	90
3.5.3	Multimedia Protocols	90
3.5.4	Authentication and Trust	90
4	Security as a Quality of Service	93
4.1	Definitions of Security QoS	94
4.2	Required System Properties	94
4.3	The QoS Model	97
4.4	Modelling Security Requirements.	98
4.4.1	The Value of Information	99
4.4.2	Enemy Properties	101
4.4.3	On the Minimal Cost of Attacks.	107
4.4.4	Protocol and Algorithm Properties.	108

4.4.5 Translating Security Requirements — Protocol Driven Model	109
4.4.6 Translating Security Requirements — Value Driven Model	112
4.5 Layering and Mapping of Security QoS	114
4.5.1 Specification of URs	116
4.5.2 Mapping URs to AARs	118
4.5.3 Mapping of AARs to LLRs	119
4.5.4 Evaluation of LLRs	121
4.6 Configuration of Secure Protocols	122
4.7 Monitoring of Secure Protocols	123
4.8 Summary	124
5 Secure Multimedia Protocols	125
5.1 The Idea Behind Dedicated Protocols	125
5.2 Multimedia Data Types	126
5.2.1 General Purpose Data	127
5.2.2 Audio Data	129
5.2.3 Video Data	135
5.3 Properties of a Dedicated Secure Video Protocol	136
5.3.1 Confidentiality	136
5.3.2 Authenticity	138
5.4 An Application: Secure WaveVideo	140
5.4.1 WaveVideo: The Inner Workings	140
5.4.2 Input Parameters	142
5.4.3 Functional Blocks	143
5.4.4 Additional Components	145
5.5 Example Processing and Interfaces	146
5.5.1 Sending Side Crypto API	146
5.5.2 Receiving Side Crypto API	149
5.5.3 WaveVideo API Support	150
5.6 Measurements and Evaluation	151
5.6.1 Reduced Content Quality in WaveVideo	151
5.6.2 Obscuration with Varying Intensity in WaveVideo	152
5.7 Summary	153

6 Da CaPo++: Realisation of a Secure Communication Framework	155
6.1 System Architecture	156
6.1.1 Users and Applications—Setup Phase	157
6.1.2 Communication Layer—Properties	158
6.1.3 Associations and Identities— Assuring Authenticity	158
6.1.4 Attribute Translation—Specifying and Translating Security Requirements	159
6.1.5 Protocol Management— Reconfiguration and Keying	160
6.1.6 Security Assurance	161
6.1.7 Keys and Certificates	161
6.2 Fundamental Assumptions	162
6.3 Framework Components	164
6.3.1 API	165
6.3.2 QoS Parameters	166
6.3.3 Communication-Layer Modules	169
6.3.4 Protocols	170
6.3.5 Key Database	171
6.3.6 Security Manager	172
6.3.7 Run-Time Security Assurance	176
6.4 Specification of Interfaces	176
6.4.1 Application—upper API	176
6.4.2 Security Manager—lower API	177
6.4.3 Security Manager—Connection Manager	179
6.4.4 Security-Related Events	179
6.5 Summary	180
7 System Evaluation	181
7.1 Comprehensive Analysis	181
7.1.1 Security Modules	182
7.1.2 Security Protocols	183
7.1.3 End-to-End Behaviour	185
7.2 Detailed Analysis	186
7.2.1 Sending Side	187
7.2.2 Authentication	190
7.2.3 Encryption	192
7.2.4 Receiving Side	192
7.2.5 Overview of Measurement	193
7.2.6 Connection Manager and Security Manager	196
7.3 Secure Messaging and Secure Audio	196
7.4 Summary	197

8 Conclusions	199
8.1 Further Considerations	200
8.2 Validation of Goals	201
8.3 Further Research Areas	202
Acknowledgements	205
Curriculum Vitae	206
List of References	207

List of Figures

Figure 1: Idealized Framework Representation	27
Figure 2: Foundations and Related Work	37
Figure 3: Mapping TCSEC on ITSEC	44
Figure 4: Algorithm Taxonomy	46
Figure 5: Secure Transmission of Application Data	57
Figure 6: ISO/OSI Layers vs. Place of Integration	58
Figure 7: The 3-Layer Model	60
Figure 8: Possible Components and Relations	65
Figure 9: Organisation of Section 3 — Related Work	68
Figure 10: Layers and Requirements	98
Figure 11: Information Value over Time	99
Figure 12: Information Value and Attack Cost	100
Figure 13: Growth of Enemy Resources over Time	105
Figure 14: Projected Participation in Zero-Cost Efforts	107
Figure 15: Dependency of Properties I	110
Figure 16: Dependency of Properties II	113
Figure 17: End-System Components	116
Figure 18: Feedback and Translation Cycle	117
Figure 19: Audio Protocol Modules	123
Figure 20: Recovery of Sampling Errors in G.721	132
Figure 21: Different Partial Encryption Schemes.	133
Figure 22: WaveVideo Architecture	141
Figure 23: Reduced Content Quality	152
Figure 24: Obscuration by Partial Encryption	152
Figure 25: Architecture Components	157
Figure 26: Comparison of Security Modules	182
Figure 27: Sender Call Graph Overview	188
Figure 28: Sender Initialization	190
Figure 29: Sender Processing	191
Figure 30: Receiver Call Graph Overview	194
Figure 31: Security Module Combination Measurements	197

List of Tables

Table 1:	Comparison of Systems	89
Table 2:	Enemy Computing Power	104
Table 3:	Possible Enemy Resources	106
Table 4:	Example Database of Properties.	111
Table 5:	Example For User Requirements	118
Table 6:	Audio Compression Costs	130
Table 7:	Partial Audio Encryption.	131
Table 8:	Module Parameters	169
Table 9:	Achieved Throughput of Security Protocols	183
Table 10:	Algorithm Costs and Throughput.	184
Table 11:	Security End-to-End Measurements	185
Table 12:	Sender Initialization Measurements	187
Table 13:	MD5 Measurements	193
Table 14:	MD4 Measurements	193
Table 15:	DES Measurements.	195
Table 16:	IDEA Measurements.	195
Table 17:	RC5-12-16 Measurements.	195

“Would you tell me, please, which way I ought to go from here?”

“That depends a good deal on where you want to get to,” said the Cat.

– Lewis Carroll, *Alice’s Adventures in Wonderland*

1 Introduction

Until recently, security in communication systems has been considered an ugly duckling by the broader user community. No justification for the design and implementation costs of security software and hardware existed, and the loss in user comfort by far outweighed security gains. Providers of network services were trusted parties, and abuse of transferred data was very unlikely. Over the last five years, the use of today’s most important open network—the Internet—has changed radically. While in the past it was mainly used by universities and other research facilities, it is now of great interest to user communities with commercial focus. This poses new demands for the network designers, such as the taxability of provided services, higher and reservable throughput, reliability, and especially security. Users of open network infrastructures incur, for example, the risk that access to transferred data is possible from intermediate nodes—and they cannot always trust their service providers not to abuse the transferred information. The cost of security is no longer perceived as prohibitive, but as a necessary investment and a requirement for the commercial use of the Internet.

While *dependable systems* as a whole have been well studied and have found a multitude of applications in the banking environment, in airplanes and air control, and wherever reliable and fail-safe systems were required, the task of providing secure communication in a large

heterogeneous environment (other than for military purposes) is just now being undertaken. Reliability and security complement each other, at least in real-world environments. Secure systems, as defined by [ITS90] or [TCS85] offer, depending on the required strength of security, a broad spectrum of capabilities. Section 2.2 contains a brief overview of such systems. To insure efficient, authentic and private communications for the common user, only a few of these capabilities need to be considered.

Confidentiality, authenticity, and authority (see Section 2.1 for definitions) of participants and data are aspects of security that are important in the current developments of services that are offered over public data networks, especially the Internet. Although a multitude of cryptographic protocols and mechanisms have been designed in the past 20 years (*e.g.* [Kruys89], [DES77], [Lampson91], [Kerberos88], to name a few), most of these security-related functions are available only in special purpose, dedicated communication infrastructures, in proprietary and limited parts of networks and are usually costly to deploy and maintain. These mechanisms have to be included into today's communication technology on a broad scale, without hindering future development of communication services and interaction of different communicating peers.

Security components have to be embedded into current and new communication systems and still allow optimal exploitation of the available resources. Security has to become a standard service component, easily used by anybody needing security. This is the goal I strive for, and its need will be demonstrated in the sections to follow. Achieving this goal requires the integration of security in a general Quality-of-Service (QoS) based framework and finding the means to express the costs and the benefits of security. Once such a definition is found, the user or application programmer can tailor the degree of security offered by the application according to his needs. The system is then also able to inform him about the additional resource consumption that he will incur. A *selection* or even *configuration* of appropriate communication protocols, depending on the demands of an application, is the concluding step for providing dynamic security on this level.

1.1 Motivation

Many parties have by now observed the requirement for confidentiality and authenticity in an infrastructure that is supposed to offer the means for electronic commerce, to allow for the existence of closed user groups, *e.g.* in *Virtual Private Networks* (VPNs), to support confidentiality requirements of the common user, and to fulfil many other sensitive tasks, such as e-commerce or e-voting. Security mechanisms that are firmly established in the communication infrastructure will allow assigning responsibility and accountability to its commercial users, which, when jurisdictional systems catch up with today's technical possibilities, may even be stronger than the contractual system that is in force now. For this to work out, and not be subject to abuse by its participants or by third-party organisations, a complete and well-founded architectural framework is required—very similar to a traditional secure system, but with more flexibility.

A principal requirement in this framework is flexibility. Assuming that multiple user communities with vastly different needs employ it, and keeping in mind that today's cryptographic mechanisms evolve nearly as fast as available computing power and other resources, the framework has to address different needs that may change rapidly. By defining the different security aspects in a communication system as Quality of Service (QoS) parameters, and by expressing these parameters independently from an algorithm or platform, a first step towards such a system can be taken. Understanding and using these abstract parameters expressing different degrees of security aspects leads to a much more homogenous integration of security in a communication framework, as in the scenario shown in Section 1.3.

While the existence of an end-to-end reliable security framework is imperative for the successful deployment of security in a commercially oriented environment, this work restricts its view to the technical communication issues. Data storage, policy-driven access control, auditing, and most upper-layer management issues are not covered. The reason for this limitation is that I perceive security in the communication subsystem as one of the more important core elements of a complete security framework. The only aspects of upper-layer management that are briefly covered are selected areas concerning authentication and key

and trust management, because without them, the underlying framework cannot function.

While studying security mechanisms and ways to employ them, one has to keep in mind that the security ultimately serves to achieve human needs, even if the humans may be hidden behind many layers of delegation when data reach the untrusted infrastructure. This directly results in requirements concerning the usability of the system, and transparency of employed mechanisms. If the system is too complex and holds too many hidden layers, it may lose its trustworthiness, and therefore, user acceptance.

1.2 Current Weaknesses and Emerging New Demands

In the previous Section, it was observed that the public poses new demands on the communication infrastructure. Two organisations having such demands are explored now as an example. Additionally, examples of incidents will be given, where the weaknesses of the current infrastructure become plainly visible.

One of the most common scenarios is the one of working groups that are distributed over a city or the continents. Several persons, all working for the same organisation, either in one of its offices, at home, or from abroad, have to gain access to corporate information data bases, exchange information, and may, from time to time, need to converse in real-time with each other. These persons may be managers of the corporation, consultants, or technical personnel. To stay competitive, the distributed enterprise will demand that its data communications remain private, and that information is authentic. Other scenarios include competing organisations (such as Chrysler and Ford) that want to share a certain amount of data and infrastructure to exploit synergies and to allow for joint efforts, but still keep most of their internal knowledge (contracts, technical innovations, *etc.*) strictly private.

At the same time, for LAN or MAN multimedia applications, such as video on-demand or audio and video conferences, the bottleneck in the current infrastructure has shifted from the actual network to the protocol implementation. This leads to the fact that end-system CPU load is

increased and is now a scarce resource for multimedia real-time applications. Trends in gigabit and terabit switching technologies indicate that the gap between network speeds and CPU speeds will grow even more, making the application of costly algorithms, as they are found in cryptographic protocols, ever more counterproductive. Thus, cryptographic solutions for above demands must take into account the costs they are causing, and a mechanism for balancing minimum cryptographic necessity against its cost must be found. Solutions based on hardware (*e.g.* link-layer encryption) may not be sufficient because they lack flexibility, are proprietary, and are costly to employ.

Weaknesses of the current infrastructure are regularly discussed in the newspapers. For example, an August 1996 resolution of the Swiss government has stated that “*Internet voting would be too vulnerable to manipulation* [SwissNC96]”. Fraud involving the spoofing of credit-card numbers that were transmitted over insecure links constitutes a substantial source of loss to credit card issuers [Slotter97]. Starting in 1995, major alerts concerning network wiretaps to recover passwords for remote logins were issued [Howard97]. On numerous occasions during the last few years, the existing infrastructure that is being used for critical and valuable tasks, has been abused, broken into, and compromised [CERT97].

Several studies of forms of attack exist (*e.g.* [Cheswick94], [Curry92], [Purser93]); therefore, only a very short classification is given here. The first type of attack targets the individuals operating and using secured systems and must be countered by organisational measures. The second type of attack targets the surrounding infrastructure of a cryptographic system, such as key certification and distribution procedures, and for example the operating systems of machines running the actual secure protocols. The third attack finally works on the cryptographic algorithms and protocols themselves, trying to reveal the content of encrypted messages and recovering keys that were employed. Future systems must resist these types of attacks as well as possible and must be flexible enough to change algorithms rapidly—on the order of days—when the currently used are proven to be weak.

Typical use of cryptographic methods in the future will also need to integrate mechanisms that allow the assertion of access rights, such as what is needed for the process of logging into a computer, the transport of secured data of an arbitrary type (such as e-mail or video) over an

arbitrary medium, the broadcasting of information that may only be received and understood by a group of limited size (such as charged TV transmissions, or interbank clearing), secure voting applications and shared control (such as multi-party signing of legal documents).¹

As shown in Section 3 on related work, currently existing systems lack the important properties of flexibility and configurability, which strongly discourages their use in open and global networks. When application specific security demands must be met, the result is an approach where security is configurable. While traditional Quality of Service (QoS) mechanisms are ideal to provide configurability, no QoS definitions for security (other than nothing or all) currently exist. Any portable and application independent solution to provide security to communication systems, such that any emerging new demand can be satisfied will rely on QoS concepts.

1.3 Sample Scenario

This section examines how a configurable and adaptive communication infrastructure offering security might behave in the view of its users. In the scenario, the following assumptions are made:

- A surrounding security framework offers its standard services, this includes, at least, a unified authentication scheme.
- The end system in question is itself trustworthy for the user.²
- The end user is used to the concept of applications that offer a variable amount of security on request, depending on which functions he wants to perform. A user interface helps him to choose the degree of security he deems necessary for a certain task.
- Administrators may enforce system- or site-wide minimum security constraints, control access, and perform other administrative functions. While the end user may be aware of these constraints, he must be unable to circumvent them.

1. These applications lead to the necessity of unification of secure systems and secure communications.
2. If the end user cannot trust his working environment, additional (trusted) hardware becomes necessary.

From a user's perspective, security should be invisible. Only in very rare instances should he be confronted with security issues. If the user unconditionally trusts his working environment, he simply needs to *identify* (and authenticate) himself once for each session. From then on, all security relevant operations may be performed transparently, unless an exception occurs, *e.g.* keying material expires or an ongoing attack is detected. This can be done because the user in fact has delegated his identity to his working environment, which can now act on his behalf. Figure 1 shows a coarse-grained representation of the elements (mostly located on a single workstation) in such a system.

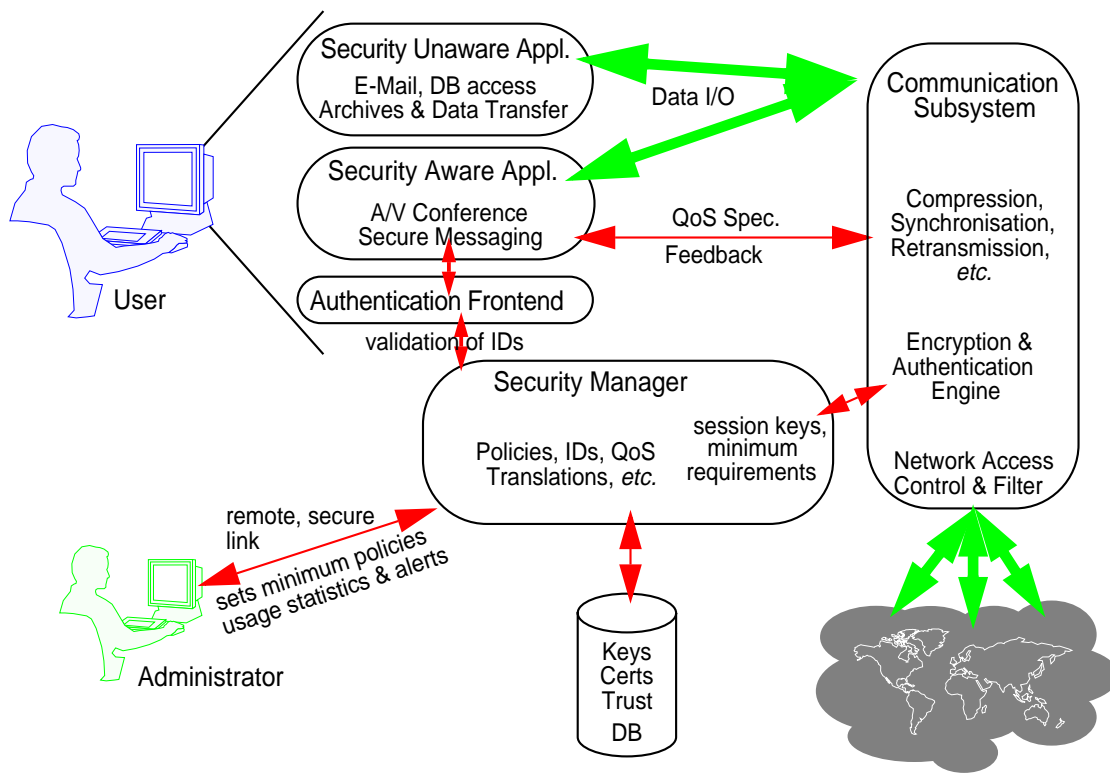


Figure 1: Idealized Framework Representation

The user of the system interacts directly with the applications running on it, while the administrator controls the policies that govern the security manager. The administrator can do this directly or remotely via a secure communication link. The *authentication frontend* is responsible for acquiring and validating the identity of the user by appropriate means. Any *security-aware application* may then communicate with the authentication frontend (via the security manager) and collect this information or related keying material. *Security-aware* here means that

application knows about security, and *e.g.* offers the user a choice on how much security he wants to employ. A *security-unaware application* would not be aware of the fact that its communications are being secured by the underlying infrastructure. Applications that use the communication subsystem specify their requirements as QoS parameters, and they may receive feedback from the communication subsystem when events (such as communication peers joining, or the detection of a likely attack) need to be reported. The security engine can perform firewall functions (network traffic filtering) as required by the security manager, and offers cryptographic algorithms and protocols to the applications. These algorithms may be closely integrated in the communication flow, *e.g.* to secure outgoing traffic.

In the following scenario, the main actor is Adam, a member of a globally distributed working group. He works together with Bob and Eve, who have not yet met each other in person. When Adam starts his working day, he logs into the secure system, and communicates with remote peers. The following tasks are explored in more detail:

- System login, sending mail
- Authenticated join of a public video conference
- Transfer moderately sensitive multimedia data
- Transfer highly sensitive multimedia and conventional data
- System logout

During each task, a short description of the actual system behaviour, and the effects the user sees are given. As long as costs and performance allow, the system should work in an unconditionally secure mode. Only if, *e.g.* multimedia processing costs, raise so high that the QoS, as required by the application, cannot be fulfilled, security (or multimedia quality) may be reduced to a user- or application-defined minimum.

System login: As a first step, Adam has to log in. The process of logging in mainly consists of specifying one's identity and authenticating it by means of a password, a hardware token, other means such as biometric features, or combinations of them.

Once Adam has proven his identity to the workstation, he can start working using his environment. If Adam trusts his workstation, he

can delegate his identity, meaning that now the system (or an application on the system) will be able to act on his behalf at least locally. Depending on the design of the authentication mechanism, the application might even act on remote systems as well. In the further steps, local—but not global—delegation of the user identity, and the implied trust, is assumed.

If Adam reads e-mail that is currently stored on his equally trusted mail server, he does not need to prove his identity to the server. The local application gaining access on the mail server receives a request for identification, and, after asking the user for confirmation for this request from the remote mail server, performs the needed proof of identity. The same behaviour is expected for remote logins, file transfers, accessing encrypted data, *etc.*¹

From the view of a communication system, this implies the existence of a unified representation of authentication requests and responses (as in systems such as [Samar95], [Kerberos94], or [CORBA96]) for distributed tasks, and the corresponding protocols.

Authenticated join: Authenticated join illustrates two concepts.

Firstly, joining participants must prove their identities to each other, and secondly, exchanged data must be authenticated, *i.e.*, bound to a participant's identity.

For the proof of identity, either a hierarchical authentication framework or more a generic web of trust [Maurer96] can be employed. When Adam takes part in a video conference with Bob and Eve, all three participants want to be assured of their respective identities. The conference is not private, meaning that anyone can join in and exchanged data are not being encrypted. The participants want to know who the other participants are and that the audio and video streams really are originating from them. After Adam, Bob, and Eve retrieve each others public keys from a large, distributed key database, they check the validity of the keys by analysing attached signatures. If a key is signed by an *introducer* (a third entity that

1. This behaviour is only achievable if the user can trust the local application not to divulge or abuse the capability of proving his identity. The problem can be mitigated by introducing trusted hardware that stays in possession of the user at all times.

they have earlier decided to trust as an introducer of public keys unknown to them, and whose public key they already hold), or by a chain of those, they can, to a certain degree, trust the key they just fetched from the database. So Bob and Eve, not having prior knowledge about their respective public keys, can still choose to trust the identity bound to those keys, if they trust Adam or another introducer.

Internally, the application that verified the authenticity of keying material received the relevant parts of a web of trust (or, in the hierarchical case, a chain of certificates) that was extracted from the global database. The end-to-end trust level was evaluated at runtime by the security manager. Throughout the session, the communication subsystem then verifies the authenticity of the different audio and video data streams and presents this information to the users via their user interfaces.

At the level of the communication system, transmitted audio and video data must be authenticated. Because it is expensive to strongly authenticate high-volume data streams (such as video), the hierarchical encoding of video compression algorithms, such as *e.g.* wavelet compression can be employed, by only authenticating certain components of the compressed video stream. Computation power is saved by using partial authentication methods (see Section 5.1). This selective application of authentication allows the participants to achieve a near-maximal video and audio performance while being convinced that they are in fact talking to the right people. If the participants trust each other group member, they might even switch to *group-wise authentication*, using one shared key to protect the traffic between all the members. This saves still more communication bandwidth, avoids separate authentication and encryption overhead, and allows to employ multicast transmissions. See [Caronni98].

Moderately secured data: Data are secured by applying encryption and authentication mechanisms to it. Users are able to select the amount of security they require, and the communication system possesses different means to satisfy those requirements.

In our example, after a while Adam, Bob, and Eve switch their topic of discussion to a more sensitive matter. They discuss the

contents of a patent application they are going to submit, and know that several other people are working in similar areas, so they want to be careful. The patent application under discussion will be filed within a month, thus they do not need long-lasting confidentiality. One of the three participants, say Eve, initiates a switch of the conference to “medium security” mode. Bob and Adam accept this, and the communication protocol now additionally encrypts data with an algorithm of appropriate strength. Purely practical considerations prevent them from using the strongest possible confidentiality. Encryption protocols offering this drastically reduce throughput and introduce unwanted communication delay. See Section 4.4 for considerations of strengths of algorithms, and how to express them on the interface from the application to the communication system. After the level of security has been increased, it can only be reduced with consent of everyone involved in the communication.

When the communication system receives a request for a certain degree of security, it will examine the properties of the involved communication links and the available resources on the end systems. It will then select and configure a protocol that is the least expensive and still offers the required degree of confidentiality. Here too, if group-wise trust exists, multicasting infrastructure can be used to reduce communication load, and the data need only be encrypted once.¹

Strongly secured data: Later, Adam and Eve may engage in a separate video conference and discuss matters of mutual interest. As they consider the exchanged information of very confidential nature, they demand a maximum of confidentiality to be provided. This means that the strongest possible encryption mechanisms must be selected, and then balanced against the minimum quality requirements of the video conference applications. If both sets of requirements can be met, a (probably low-quality) connection is established, otherwise the participants are informed of the conflict, and advised to reconsider their requirements. They may now

1. If supporting hardware is available on both sides, this security can be quite strong, and if underlying links are tagged as being “more secure” no confidentiality mechanisms at all will be engaged.

reduce the minimal quality requirements for their video conference, or their security requirements.

Having finished their private dialogue, they lessen the security of the link with mutual consent, and join back to the discussion with Bob and maybe other participants.

System logout: Finally, Adam will leave his trusted system. When he logs out, the machine will not longer be able to perform an action on his behalf, because all secret information was destroyed when he logged out. In the case of an untrusted system, this, naturally, needs special consideration. One common solution in that case is to add a trusted hardware component to the system, such as a chip-card that is owned by the user, or to have delegated rights implicitly expire after a certain period. Both methods leave the system in a state in which it is unable to perform actions on behalf of a user.

In above description of user and communication system behaviour, several distinctive advantageous properties of the proposed communication framework can be observed. *Single login* facilitates the system operation of all users, not requiring multiple authentications. Allowing security to be configured dynamically into a system, and the ability to specify a generic degree of security by the users or applications results in a powerful and easy-to-use instrument. Adam, Bob, and Eve are able to engage in an authenticated and private conversation, and drop back to a public discussion afterwards dynamically, without nothing more than a few mouse clicks. The quality of the service offered to them balanced their security requirements against independent protocol properties such as compression quality, frame rate, or image resolution and size, and adapted these, or suggested an effective adaption of parameters. Finally, by thriving on the implicit trust that was derived from the globally available public key database, communication setup and authentication of the peers took place in a very easy fashion.

1.4 Claims

Some elements of the behaviour described in the scenario of Section 1.3 are of fundamental interest to future communication systems, and offer a distinct extension to today's secure systems. As is shown in Section 3.2, Related Work, the research in this area has barely begun. In this work, the achievement of the following results is to be expected:

Security as a Quality of Service: QoS parameters are gaining importance in communication systems. Security must become a part of those parameters. Thus a method has to be developed that allows the specification of *e.g.* confidentiality and authenticity requirements as QoS parameters. A definition of security parameters supporting a wide range of applications must be developed, and should formally specify security requirements. This implies the following achievements:

- Security Requirements that can be expressed as numerical QoS parameters, allowing for a level of abstraction over the use of concrete algorithm information, and for a qualitative or even quantitative expression of the different security properties of a distinct service.
- Existing protocols and security mechanisms are mapped onto abovementioned QoS parameters. This implies the need to assess their relative and absolute strength.
- Algorithms that allow for the selecting of the best fitting protocols, *i.e.* the ones that offer the required degree of security having minimal cost. They imply the ability to negotiate for an appropriate degree of security versus cost in a distributed manner between multiple communicating peers.

Trust and Authenticity: Security mechanisms in the communication system must be supplemented by means to express trust and authenticity of users, communication peers and system components. Additionally, keying material must be offered. Without such mechanisms, communication security is worthless in practical environments. The following steps thus will be pursued:

- Realisation of a simple public-key architecture involving the principle of the web-of-trust.
- Provision of a method to prove correct configuration for a protocol offering security. More precise, for a given correct description of authenticated protocol building blocks, and when reliably knowing how these building blocks are interconnected, the assurance whether this protocol conforms to the specified security requirements can be given efficiently at run-time.
- A new user-dependant way of confirming the authenticity of applications (and the user they claim to represent) is given by validating the integrity of the application at run-time, and receiving confirmation of the authenticity of requests via an independent channel to the end user.

Practical Proof of Concept: To show the usability and efficiency of above mentioned mechanisms and algorithms, they are integrated into an innovative communication system. This allows the evaluation of the concepts in a working testbed with prototypical applications:

- Relevant security protocol elements are shown to be dynamically configurable.
- Resulting from this, a set of real communication protocols is designed for applications transporting audio and video, thus offering validation of performance. Dedicated protocols for encryption and authentication of multimedia data under certain constraints are used.
- To make the system usable, key-generation and -management functions and dedicated data exchange functions are made available.
- It is shown that the dynamic selection of security protocols gives a better price *vs.* performance ratio. Expected benefits are more (and adequate) confidentiality and authenticity, well balanced with the acceptable amount of performance degradation to be expected in secured communication protocols, and the increase of processing cost.

1.5 Cryptography and Politics

In our world, information is related to control, and ultimately to power. Possession and distribution of information is a highly sensitive process, which all factions holding power, or warring for it, try to influence to their utmost advantage. Cryptography is the science of securing information, *e.g.* protecting its origin and transfer. As a direct consequence, the powers-that-be try to influence the deployment of cryptography in the world. Until very recently, the United States of America treated cryptographic mechanisms like weapons technology, and some countries, *e.g.* France, even prohibited domestic encrypted communications unless the keys to the communication were available to specific authorities within the state. Civil liberty groups oppose these influences of governments, requesting free and protected speech for everybody¹, leading to heated discussions and political battles.

The two well-declared goals of states are to be able to read domestic and foreign traffic for criminal pursuit, national security, and protection or furthering of commercial interests. To this end, the strength of products implementing privacy is sought to be reduced by regulation, and key recovery features are imposed on the industry. A full, albeit strongly opinionated, treatment can be found in [Abelson98].

In today's business processes, key recovery of business-related stored data (in contrast to communicated data) is a undisputed necessity. Even the present work, which focuses on the technical view and does try not to address risk management issues, has to be concerned with the issue of key recovery. For the scope of this work, I decided not to consider key recovery and the issues of cryptography that are related to politics. All communication protocols are designed in such a way as to present the strongest possible protection, hindering as much as possible third-party key recovery processes that can be abused to gain illegitimate access to private data.

1. See for example www.cdt.org, or www.privacy.org.

1.6 Outline

The foundations of secure communication frameworks are security QoS, secure protocols and finally authenticity and trust considerations of peers and components. In Section 2, fundamental issues related to security and secure communications are discussed, building a first view of a possible system architecture, and the base for Section 3, where related work and standardisation efforts are explored in the focus of above mentioned fundamental views.

The definition of security QoS, mapping processes and configuration issues are the topic of Section 4, without any direct connection to current implementations. Section 5 elaborates upon this under the aspect of partially securing multimedia data streams, and at the same time examines the applicability to a video codec. Here, some measurements performed with an implementation based on WaveVideo are also performed.

Some aspects of the work were implemented in the context of Da CaPo++. Section 6 outlines the relevant system architecture, assumptions made, and the implementation issues. The system behaviour and performance of the incorporated security mechanisms are evaluated in Section 7, leading to Section 8 where the final conclusions of the work are drawn.