SEMPER

Secure Electronic Marketplace for Europe

Gérard Lacoste, Birgit Pfitzmann,
Michael Steiner, Michael Waidner (Editors)

Final report of Project *SEMPER*
(Public Version containing Part I only)
June 19, 2000

SEMPER is part of the European Commission's ACTS Programme (Advanced Communications Technologies and Services). Funding is provided by the partner organisations, the European Union, and the Swiss Federal Department for Education and Science.

Foreword

Some years ago, businesses could choose whether to migrate to electronic commerce, however, today it seems they have no choice. Predictions indicate that companies that do not make the necessary changes will be overrun by competition and ultimately fail. Therefore, we see more and more companies undergoing tremendous transformation in order to adapt to the new business paradigm. At the same time new companies are being established. One thing these companies have in common is the increased dependency on security technology. The invention of electronic commerce has changed the role of security technologies from being merely a protector to being also an enabler of electronic commerce, and it is clear that the development of security technology is a key enabler in the growth and deployment of electronic commerce. This has been recognised at European level [69].

The launch of a comprehensive EU policy in the area of security in open networks is fairly recent with the adoption of a Communication on cryptography in October 1997 [67]. A very important complement and support to the European policy is the European Commission's contribution to overcome technological barriers by giving special importance to R&D (Research and Development) activities.

The *SEMPER* project was launched in September 1995 and was funded partly by the European Community within the *Advanced Communication Technologies and Services* (ACTS) specific research programme part of the Fourth Framework Program (1994-1998). In this book the *SEMPER* project team presents in a coherent, integrated, and readable form the issues addressed, the motivation for the work carried out, and the key results obtained.

SEMPER is an innovative project in several aspects. What really makes it innovative and impressive is the integration of the following components into an overall security framework for electronic commerce:

- *SEMPER* is the first project aiming at securing electronic commerce as a whole by developing a technical security framework realised as a middleware. This brings forward two advantages. First, such a technical security framework supports multiple business scenarios by providing powerful security services to applications implementing the business processes. A novelty compared to other security middleware is the provision of security mechanisms through a more commerce-oriented application programming interface. Second, the development environment shields application designers from the security implementation details and their evolution over time.
- *SEMPER* provides an open security platform which can be configured with relevant modules in order to cope with national regulations.
- A trustworthy user interface, TINGUIN, which ensures that users can securely manage information. TINGUIN provides a single point of interaction between users and their secure platform. Such a single interface is essential to ensure users' consistent perception of their security.
- *SEMPER* has proposed a legal framework for establishing legal predictability of electronic commerce. An interesting result is the adaptation of the technical and legal frameworks to each other by enabling the user's tools to visualize important legal aspects and to manage legal parameters.

The results in this book constitute a major contribution to the development of secure electronic commerce and the work presented has set the scene for future directions in secure electronic commerce. The last chapter of the book highlights some open problems related to the work done by the project. However, many more exist and there is still a lot to be done before the goal of secure electronic commerce will be reached. This has been recognised at European level, and the security-related R&D activities will be intensified under the new 5th Framework Program (1998–2002). Within the 5th Framework Program, the Information Society Technology (IST) Programme addresses the technologies of the online world.

Everyone interested in investigating the state of the art and future directions for secure electronic commerce should find this book extremely valuable and it is without any reservation that I strongly recommend it.

May 2000

Spyros Konidaris
European Commission
Director a.i. – DG XIII-F

Preface

Since the invention of the World-Wide Web (WWW) in 1991, Internet-based electronic commerce has been transformed from a mere idea into reality. Customers browse through catalogues, search for best offers, order goods, and pay for them electronically. Information services can be subscribed online, and many newspapers and scientific journals are readable via the Internet. Most financial institutions have some sort of online presence, allowing their customers to access and manage their accounts, make financial transactions, trade stocks, and so on. Some countries already support filing tax declarations electronically. Electronic mails are exchanged within and between enterprises and often already replace fax copies. Soon there will be no enterprise left without some Internet presence, if only for advertisement reasons. In early 1998 more than 2 million web servers were connected to the Internet, and more than 30 million host computers [180]. Internet business is estimated to have reached $50 to $100 billion in 1998, mostly in business-to-business trade, and continues to increase at a high rate of growth [92].

Thus, doing some electronic business on the Internet is already an easy task. As is cheating and snooping. Several reasons contribute to this insecurity. The Internet does not offer much security per se. Eavesdropping and acting under false identity is simple. Stealing data is undetectable in most cases. Popular PC operating systems offer little or no security against viruses and other malicious software, which means that users cannot even trust the information displayed on their own screens. At the same time, user awareness of security risks is threateningly low.

The only well-accepted security tool for the World-Wide Web is the Secure Sockets Layer protocol (SSL), which provides a secure pipe between web client and web server [79]. It cannot generate signed messages or signed receipts, which naturally makes it unsuitable to tasks like electronic online payments and contract signing. And even for SSL, the problem of *visualizing* security to the user is unsolved: most WWW browsers only distinguish between "secure" and "insecure" connections, but do not tell the users in a simple way with *whom* exactly they are communicating over an established "secure" channel.

Such user interface problems are amplified by the fact that today's electronic-commerce systems offer little support in maintaining consistency in data and security among the different parts of a business process. As in the paper world, users have to fill in the same data again and again, copy data on several forms, accumulate data from different transactions by hand, and so on. More problems are revealed if one looks at the legal aspects: often it is not clear—or not even decided—who bears liability and which country's law is applicable in a specific situation.

In 1994 we, the *SEMPER* consortium, came up with the idea that all these security problems could best be solved by grouping all necessary security technologies under a coherent and open software framework, and a single, consistent user interface. Such a framework should allow automatic linking of parts of a business process. Necessarily, it should support not only secure communication and payments, but also the negotiation of business and security parameters, fair exchange of documents (as in contract signing and certified mail), handling of disputes among the parties, etc. Besides keeping data confidential, it should also grant its users some degree of anonymity and unobservability—like users on physical marketplaces can act anonymously. All this should be based on requirements from the market, and should be consistent with the legal systems.

The idea was turned into a proposal to the European Commission, with the result that we started work on implementing this idea in September 1995. We concluded the project early 1999. Some partners are currently exploiting parts of *SEMPER*, but we also aim at more complete use of the framework in successor projects and products. This book summarizes our main results.

Part I gives an overview of our solutions, i.e., the technical framework and a proposal how to tackle the open legal questions. This part is intended to be readable by everyone, i.e., it does not presuppose a specific technical background except some basic familiarity with the Internet.

Part II covers topics for which fundamentally new scientific or engineering results were obtained, and looking at them in detail would be beneficial to everybody working in the field. See the introduction of Part II for more details.

The results of *SEMPER*, including the full architecture, prototype description, and results from expert surveys and trial evaluations, are documented in a number of public, formal deliverables. These are available online from `http://www.semper.org`, or by writing to: IBM Zurich Research Laboratory, Computer Science Department/Project *SEMPER*, Säumerstrasse 4, CH-8803 Rüschlikon, Switzerland.

May 2000

Gérard Lacoste
Birgit Pfitzmann
Michael Steiner
Michael Waidner

Authors of this report: Part I was written, based on the results of the entire project, by *Birgit Baum-Waidner, Gérard Lacoste, Birgit Pfitzmann, Michael Steiner, Michael Waidner,* and *Arnd Weber.* Chapter 5 was written by *Michael Waidner.* Chapter 6 was written by *N. Asokan, Birgit Baum-Waidner, Torben P. Pedersen, Birgit Pfitzmann, Matthias Schunter, Michael Steiner,* and *Michael Waidner.* Chapter 7 was written by *Dale Whinnett* and *Reinder Wolthuis.* Chapter 8 was written by *Akis Hamamtzoglou, Thomas Hecht, Giannis Papadopoulos,* and *Arnd Weber.* Chapter 9 was written by *Rolf Michelsen, Stig Mjølsnes, Petros Pantis,* and *Kostas Tzelepis.* Chapter 10 was written by *Matthias Schunter.* Chapter 11 was written by *N. Asokan* and *Michael Steiner* and is based on [1, 9, 10]. Chapter 12 was written by *Maria Gatziani, Torben P. Pedersen,* and *Kambiz Zangeneh.* Chapter 13 was written by *Birgit Baum-Waidner.* Chapter 14 was written by *Birgit Baum-Waidner* and *Rita Zihlmann.* Chapter 15 was written by *Michael Waidner.*

People and organizations who contributed to *SEMPER*: *Fabrice Clerc, Philippe Magliulo, Marc Mazoué,* and *Marie-Jo Revillet* from CNET - France Télécom. *Holger Erichsen* and *Bernd Horsch* from the Commerzbank. *Bjarke Dahl Ebert, Maria Gatziani, Peter Landrock, Kim Lueders-Jensen, Timmy G. Madsen, Jesper Drud Nielsen, Thomas Sepstrup Nielsen,* and *Torben P. Pedersen* from Cryptomathic. *Paul Dinnissen, Berry Schoenmakers,* and *Bryce Wilcox* from Digicash. *Akis Hamamtzoglou, John Katakis, Sophia Koutsoukou, Dimitrios Livas,* and *Giannis Papadopoulos* from EUROCOM EXPERTISE. *Christoph Baert, John Schey,* and *John West* from Europay International. *Michael Ehrl, Thomas Hecht,* and *Ralf Kuron* from FOGRA Forschungsgesellschaft Druck e.V.. *Horst Ehmke, Matthias Enzmann, Rüdiger Grimm, Tobias Himstedt, Basawarai Patil, Wolfgang Putz,* and *Kambiz Zangeneh* from the Forschungszentrum Informationstechnik mbH (GMD). *Jean-Marie Blanchère, Sylvain Cornillon, Gerard Lacoste, Philippe Leblanc, Jean-Pierre Le Heiget,* and *Christian Navarro* from IBM France, Centre d'Études et Recherches; *Ulrich Einig, Karsten Riede,* and *Christian Thiel* from IBM Heidelberg; *Jose L. Abad-Peiro, N. Asokan, Andreas Fleuti, Ceki Gülcü, Günter Karjoth, Ferdinando Loiacono, Mehdi Nassehi, Thomas Schweinberger, Michael Steiner, Els van Herreweghen,* and *Michael Waidner* from IBM Research, Zürich. *Petros Pantis, Maria Tsakali,* and *Kostas Tzelepis* from INTRACOM. *Sylvain Cornillon, Sharon Prins, Jako Swanenburg, Matthijs de Vries,* and *Reinder Wolthuis* from KPN Research Netherlands. *D.M.A. Schaap* from MARIS. *Mathias Flenker, Stefan Liesem, Christian Petersen,* and *Ingo Saleck* from Otto Versand. *Mogens Rom Anderson, Birgit Baum-Waidner, Klaus Becker, Felix Jaggi, Thomas Mittelholzer, Armin Müller, Claus Rasmussen, Rainer Rueppel, Bruno Wildhaber,* and *Rita Zihlmann* from Entrust/r3 security engineering ag. *Rolf Michelsen* and *Stig Frode Mjølsnes* from SINTEF Telecom and Informatics. *Peter Bosch, Dick Bulterman, Ray Hirschfeld, Jaap Henk Hoepman, Sjoerd Mullender,* and *Louis Salvail* from the Stichting Mathematisch Centrum (CWI). *Matthias Schunter* from the Universität Dortmund. *Ingo Pippow, Jan Reichert, Arnd Weber,* and *Dale Whinnett* from the Universität Freiburg, Institut für Informatik und Gesellschaft. *Birgit Pfitzmann* and *Matthias Schunter* from the Universität Hildesheim, Institut für Informatik. *Tom Beiler, Jürgen Brauckmann, Lothar Fritsch, Birgit Pfitzmann,* and *Matthias Schunter* from the Universität des Saarlandes, Saarbrücken, Fachbereich Informatik. Sponsoring partners of *SEMPER* were Banksys (Belgium), Banque Générale du Luxembourg (Luxembourg), Enyca (Spain), and Telekurs/Payserv (Switzerland). The project was led by IBM.

The following organisations provided the infrastructure for the *SEMPER* trials: in France: ACRI, Actimedia, Centre d'Études et Recherches IBM France, Centre International de Communications Avancées, France-Télécom, and IDATE; in Germany: Commerzbank, FOGRA Forschungsgesellschaft Druck e.V., Forschungszentrum Informationstechnik mbH (GMD), Gesellschaft für Zahlungssysteme (GZS), Otto Versand, Universität Freiburg, Institut für Informatik und Gesellschaft, and Universität des Saarlandes, Saarbrücken, Fachbereich Informatik; in Greece: EUROCOM EXPERTISE; in The Netherlands: KPN Research, MARIS, and Stichting Mathematisch Centrum (CWI); in Switzerland: IBM Research, Zürich; in the UK: Oil and Gas Product Library Ltd. (OPL); in the USA: GTE; and the members of the MOMENTS consortium.

SEMPER was part of the Advanced Communication Technologies and Services (ACTS) research program established by the European Commission for 1994-1998, and received funding under contract *AC026* from the European Commission DGXIII and the Swiss Bundesamt für Bildung und Wissenschaft.

Acknowledgments: Interesting discussions with several people outside the consortium helped us to

develop and to refine the ideas presented in this book. In particular, we would like to thank *Alain André, Patrick Aubry, Ali Bahreman, Philippe Bardet, Annarosa Baum, Joachim Biskup, Jay Black, Michel Bosco, Peter Büttner, Mario Campolargo, David Chaum, Anne Clarke, Marc Dacier, Michel Dauphin, Michel Fossaert, Renaud di Francesco, Michel Frenkiel, Philippe Garnesson, Mark Greene, Phil Janson, Spiros Konidaris, Jens Kristensen, Philippe Lefevbre, Karine Lieres, Mark Linehan, Jean-Pierre Meyer, Refik Molva, François Montagner, Kostas Papanikolaou, Peter de Rooij, Paul Rottier, Jukka Salo, Boris Saulnier, Tom Scanlan, Julia Sime, Hansrudolf Thomann, Paul Timmers, Gene Tsudik, Joep Van de Veer, Pierre Vannel, Hans-Dieter Zimmermann, and Rosalie Zobel.*

Contents

Part I

The Vision of *SEMPER*

This Part I provides a condensed overview of the objectives, focus and results of the project SEMPER. It is written for a general audience interested in electronic commerce and does not require any particular technical knowledge.

Chapter 1 describes scenarios of electronic commerce and the special security problems that distinguish it from traditional commerce, reviews existing approaches and the most important basic security techniques, and presents the goals and focus of SEMPER.

Chapter 2 summarizes the technical part of the SEMPER approach. We provide a model of electronic commerce, identify the services needed for secure *electronic commerce, and propose an architecture supporting these services.*

Chapter 3 discusses legal aspects of electronic commerce and summarizes our proposal on how to tackle them: the SEMPER Electronic Commerce Agreement (SECA). A particularly important aspect is fair liability distribution for digital signatures given today's technology.

Chapter 4 summarizes the benefits and potential of the SEMPER framework from a commercial point of view, in particular potential product lines.

Necessarily, the presentation in Part I is rather high-level. In Part II we provide details on the most important and innovative issues.

Chapter 1

Secure Electronic Commerce

This chapter sets the stage for the subsequent presentation of our results. We present examples of electronic commerce and the need for security, existing approaches and their shortcomings, and the overall goals and the specific focus of *SEMPER*.

1.1 The Notion of "Electronic Commerce"

The term *electronic commerce* is generally understood to span the whole range of business situations that are at least partially supported by a communication network such as the Internet. This includes the informative parts of commerce—like the provision of business directories, catalogues, and help desks—but also the legally more challenging parts, like signing contracts, transferring funds, executing contracts, issuing credentials, or delivering intangible goods. All the steps must be documented in a legally sound way, e.g., for taxation or as a provision for court disputes.

The project *SEMPER* focused on these security-relevant parts of electronic commerce. In an abstract sense, these are all those parts that transfer or generate values, rights or obligations. In order to see what this means more concretely, let us consider two important examples of Internet-based electronic commerce, shopping over the Internet and business-to-business commerce.

1.1.1 Example 1: Shopping over the Internet

Our first example covers the "standard" case of business-to-consumer electronic commerce over the Internet:

Merchants offer their products and services over the Internet, which means they make all the necessary information accessible via a World-Wide Web server (on-line catalogue), allow consumers to place orders electronically by filling in Web-forms or sending electronic mail, and link this new channel to their back-end system which finally fulfills the orders. Payment might be online, e.g., by sending credit-card details to the merchant, or out-of-band (i.e., outside the network), e.g., by invoicing, and it might take place before or after delivery. The merchandise might be CDs, books, computers, cars, nights in a hotel, flights over the Atlantic, stock shares, or Teddy bears.

Typically there is no pre-established business relation between the merchant and the customer. Thus they are unlikely to trust each other a priori: The merchant will require some guarantee of payment before delivering goods, and the customer will require some guarantee of getting the right goods before making a payment. Currently both requirements are satisfied in a very weak sense only:

- The merchant can ask the customer for a credit-card number, and can get an authorization from the credit-card organization for the amount the customer will have to pay, before processing the order. This creates some confidence in the customer, but since the customer does not "sign" a payment order the merchant has no real guarantee that he can capture the amount. The card holder might always revoke the payment subsequently.
  In theory the customer does not take any risk here, as he can always cancel a transaction (this is the regulation for MOTO = mail-order/telephone-order transactions). In practice, however, many people

do not check their credit card statements carefully, and canceling a bogus transaction can be very cumbersome. In Europe it is not even clear that canceling succeeds in real life in spite of the MOTO regulations.

The problem becomes worse by the fact that the Internet does not support identification of business partners, i.e., the customer cannot be sure whether he is talking with a respectable, honest merchant or a criminal, and thus criminals might easily collect lots of credit-card numbers. (SSL, a protocol for secure communication between client and server, has the potential to provide such partner identification. Section 1.3.1 discusses it and its limitations.)

- In contrast to just sending the credit-card number and expiration date to the merchant, real payment systems like SET [126] generate some evidence for a payment. This improves the overall security as it becomes more difficult to forge payments. But it also often means that the customer has to take the larger share of the remaining risk (see Section 1.2.3 for such risks) as he cannot simply revoke a payment any more.[1]
- For catalogue retailing of physical goods, the merchant can ask for payment-on-delivery. Then he will always be paid for delivered goods, but he might still lose money on returned goods due to wasted production, handling and shipping. Usually the customer does not take any risk. Obviously, this approach is unsuitable for intangible goods. It is also unsuitable in cases where production, handling or shipping are expensive, or where the number of "fake" orders is not limited and thus even small expenses can add up to a substantial loss.
- For continuous services, like subscriptions, the merchant can ask the customer to sign a contract in writing, i.e., the relation between merchant and customer is established out-of-band, in the classical, paper-based way. (The same holds for the relation between bank and customer in home banking.) This is pretty safe for the merchant, but unsuitable for spontaneous purchases and in general for low-value transactions. Moreover, many such contracts are not very advantageous for the customer as they assign all liability to him, without ensuring that he is technically equipped for protecting himself accordingly (see Section 1.2.3).

In the following chapters, we show how *SEMPER* improves upon all these approaches by eliminating many risks and allowing reasonable limitations of the remaining risks for all parties.

1.1.2 Example 2: Business-to-Business Commerce

This second example covers the typical business-to-business electronic commerce.

The most basic way for business-to-business electronic commerce, electronic mail, is already well established and often replaces informal business letters, phone calls, and printed product documentation. Typically, payment is performed out-of-band through cheques or funds transfer, and the time of payment is unrelated to that of delivery. For instance, a customer's account is maintained and settled once a month based on the transactions performed during that period. Corporate credit cards are also increasingly used.

As soon as more formal business documents, e.g., orders, are also sent electronically, the question whether they are legally binding is quite similar to Example 1. However, for the near future, partners in business-to-business electronic commerce often have a formal trading partner agreement in place, and in such cases it is simpler to establish a secure communication channel between the partners or even to agree on specific liabilities.

In the long run, more flexible and spontaneous relationships are desired for electronic commerce, e.g., compare the goals of RosettaNet [154]. There are several projects on using EDI (Electronic Data Interchange) over the Internet. Some, such as OBI [134], are already deployed but only automate simple types of procurement, e.g., high-volume low-value goods in a business-to-business scenario. XML/EDI [137] is much

[1] Originally SET was only intended for reducing the overall risk, without placing more of this risk on the customer. The SET specifications ([126], Book 2, Section 2, "Services/Caveat") explicitly state: *"SET does not provide non-repudiation."* But they also *"permit non-repudiation via rules and policies of individual payment card brand implementations,"* and some implementations use this to assign the full risk to the cardholder. For instance the Swiss SET registration form [169] says: *"4. All initiated SET transactions which are based on a system-controlled, true legitimation, using the card number and the matching digital signature, may be charged to the Cardholder without reservation and are legally binding for him/her; this applies even if the SET transaction was not actually initiated by the Cardholder. [...] This legitimation agreement means that the risks arising from the misuse of the password and/or the personal code are in principle incurred by the Cardholder."*

more general, but not yet concrete. Others try to structure the complexity and diversity of the business processes by supporting general supply-chain management and other inter-company workflows over the Internet. So far, none of these projects contains many security considerations. However, studies such as [155] show that lack of security is recognized in industry as one of the main impeding factors for electronic commerce in business scenarios.

1.2 What's Special about Electronic Commerce?

Both types of electronic commerce discussed in Section 1.1 parallel traditional, paper-oriented types, and thus the high-level security requirements are intuitively clear. But *satisfying* these requirements in the electronic world is quite different from satisfying them in the paper world.

1.2.1 Virtuality of Electronic Commerce

Electronic commerce takes place in a virtual world, where everybody is "equal." There is no difference between a message received from the office next door and one from the other end of the world, or between a message sent within the same country and one sent across the border. The web site of a respectable well-known company does not need to look more serious or trustworthy than the web site of a garage business—or a web site set up by some criminal. The Internet also does not ensure a standard semantics for names: `www.XYZ.com` does not necessarily belong to company XYZ.

Hence, even without any technical attacks on the network, establishing *trust* between business partners in electronic commerce requires additional effort. Similar effort is needed for collecting evidence for cases where a dispute occurs. The two problems are related but not identical: Trust can be in a purely "digital" entity, e.g., a certain web server becomes known for delivering useful and bug-free programs or beautiful pictures. Evidence more often needs to point back to a real-world entity, e.g., for paying compensation or other punishment in case of serious misbehavior.

Another aspect of virtuality is that electronic commerce can easily take place across borders. Typically this reduces the legal certainty; e.g., in many cases details such as which country's law is applicable, or who bears what liability under which conditions, are not clear and decrease overall security and trust. This is discussed in more detail in Chapter 3.

1.2.2 The Internet as a Hostile Environment

Typically, messages sent over the Internet are processed by several intermediate computers before they are delivered to the intended recipient. Nothing prevents these intermediate computers from screening traffic for interesting information; there is no confidentiality. Sender addresses are not verifiable: It is trivial to send an electronic mail under a wrong identity. In fact, "multiple personalities" are a built-in feature of many Internet mail clients. Attackers can also effectively change the meaning of addresses, redirecting traffic sent to `www.XYZ.com` to an address of the attacker's choice.[2]

1.2.3 Insecure User Equipment

There is a lot of literature on how to break into PC computer systems, e.g., [54, 55, 131], which is the usual type of device used for electronic commerce over the Internet (at least for consumers and small businesses). Apart from low-tech attacks by people who get physically near the PC, the main problem are attacks with Trojan horses and viri, i.e., malicious code hidden in useful code or data. Traditionally this kind of rogue code was mostly transported in free or pirated software and had to be explicitly loaded by the user. Propagation was slow and user education and virus checkers provided good countermeasures against these distribution media. However, with the advent of the Internet these attacks can propagate much quicker and can be much harder to detect and prevent. Standard browsers, emailers, and also the operating systems themselves have proved to have bugs that allow outsiders to place code on a user's machine while the user is doing his normal

[2]This last problem will be solved by the emerging Secure Domain Name Service [57] and proper use of IPSEC [17].

work and not intending to load code, e.g., while he is only browsing [52, 51] or reading mail.[3] Once the attacker has malicious code on the machine, this code can in particular look for secret data and access codes used by standard electronic-commerce applications (or entered by the user into them) and send those out to the attacker.

If high-value transactions can be made from standard PCs, and become as legally significant and thus valuable as traditional ones, it would be very surprising not to see many attempts to break into such systems. There is no reasonable way for users to protect themselves from such attacks except by giving up all Internet access and thus electronic commerce [177]. Even smartcards do not help very far: Even if they protect secrets, the user's access data and commands to them go through the PC and can still be caught and modified there by malicious software.

Some users feel protected because they only use their PC for low-value transactions, but of course this is an illusion if an attacker can just as well do high-value transactions or a very large number of them with the same set-up. Similarly, some users do not seem to be aware that they are vulnerable to attacks from the Internet not only if they have servers running, but as soon as they use anything like browsers or email. Ignoring these risks makes them even more vulnerable.

1.2.4 New Opportunities to Commit Fraud

Today, fraud in electronic commerce is a relatively minor problem. But most likely the main reason is that the overall volume of electronic commerce has been too small. Attacking traditional commerce has been more cost efficient for serious criminals. But we are convinced that this will change dramatically: the more successful electronic commerce becomes, the more attractive becomes electronic fraud.

An additional argument is that electronic commerce has some properties that favor fraud: Electronic fraud can be highly automated, i.e., a single criminal might be able to simultaneously attack a whole user population, e.g., all the customers of a certain company. Some fraud procedures can be performed at a very high speed, which might make attacks on low-value services attractive that would never represent a risk in the physical world. Plausibility checks like "not too many requests from a single person" do not necessarily work, since criminals might be able to generate an unlimited number of new identities. Cheaters can commit their crime remotely, e.g., from countries with weak criminal laws or no interest to prosecute fraud committed abroad. Finding promising targets is also facilitated by the Internet, as it is usually quite easy to check which companies use which security mechanisms.[4]

1.3 Existing Approaches to Secure Electronic Commerce

Basically two security mechanisms are actually applied in today's electronic commerce: SSL as a means to establish secure channels between business partners, and Trusted Market Providers as a means to implement centralized security policies. Two further mechanisms are widely discussed and used in prototypes and trials, digital signatures with corresponding public-key infrastructures and secure payment systems.

1.3.1 Secure Channels

Most electronic-commerce solutions use the SSL protocol [79] to establish a secure channel between the business partners, e.g., a buyer's World-Wide Web browser and a merchant's server. At least the server, optionally also the client, are identified by means of public-key certificates. "Secure" means that all messages sent through this channel are integrity-protected and encrypted, i.e., any modification of the messages can be detected and outsiders cannot read the messages.

Although SSL is a useful tool, most of the security problems in electronic commerce cannot be solved with it:

[3] See for example `http://www.newscientist.com/cgi-bin/pageserver.cgi?/ns/980425/nwindows.html`.

[4] One should also be aware that unprotected email and browsing can be a good source for conventional burglars and blackmailers to find attractive targets, e.g., by looking for vacation servers, or people who search for and buy expensive tangible goods.

- SSL uses *symmetric* cryptography for the main messages. This means that sender and recipient of a message have the same secrets. Therefore the recipient does not gain any evidence (e.g., for use in court) that the sender actually sent a message (e.g., made a binding offer or ordered expensive goods) because the recipient could just as well have produced the message himself.
- One design goal for SSL was to provide a *transparent* secure channel, i.e., after the set-up phase the application does not need to be aware of SSL. This is a very nice property for simple secure communication. However, for electronic commerce it means that even if asymmetric cryptography were used throughout, SSL messages could not serve as evidence in any easy way, because SSL would automatically sign all sorts of messages without clear relation to what the user actually saw on the screen and authorized.
- SSL is just a protocol (i.e., it defines message formats and other such technicalities). *No legal meaning* is typically associated to the keys used in it (compare Section 1.3.3). Of course, this is fine for its intended use as protection from outsiders, and it is the correct legal reflection of the technical facts in the first two items that prevent it from being used for protection of two communication partners from each other.
- The typical user of SSL clicks on a link to a certain partner in his browser, sees a lock symbol that shows that a secure channel was established, and believes he now has a secure channel to the desired partner. However, as long as he does not go to a particular window to check a certificate, an attacker can still just as easily cheat him as without SSL, because the lock symbol only means that there is a secure channel to *anyone*.

The last issue is not a problem of SSL as a protocol (and not even a limitation of secure channels in general, as the other issues are). But it shows that a secure user-interface design and secure linking between different steps, here between the name clicked on and the name of the partner on the channel, are important for overall security. These are two important topics in *SEMPER*; see below.

1.3.2 Trusted Market Provider

Many electronic-commerce solutions are organized around a central server which manages a marketplace and is responsible to ensure the security of *all* parties on the marketplace. For instance, Open Market [135] and existing auction and broker services use this model. Another example are workflow systems which support the co-operation between pre-defined entities according to pre-defined rules and procedures. Typically a workflow is seen as a single database process which interacts with different entities. Security of a workflow is primarily given by assigning each of the entities the necessary access rights, in the usual database sense. These rights are determined by the entity's role with respect to the workflow and by the current work item.

The disadvantages of this approach are that the players have to trust this central server, usually even without being able to detect fraud, and that this naturally results in closed solutions.

1.3.3 Digital Signatures and Public-Key Infrastructures

Proving that a certain action in the virtual world was performed by a certain entity is technically primarily facilitated by *digital signatures*, first introduced in [56]. A digital signature scheme is a cryptographic mechanism that allows its users to attach certain bit strings, called "digital signatures," to digital documents such that, in principle, everybody can check signatures, but nobody can forge a signature under a document the supposed signer did not actually sign. The information necessary to verify signatures of a certain person is called the *public key*, and the information necessary to generate the signatures is called the *secret key*. For a secure signature scheme, nobody who only knows the public key and some signatures generated by the true holder of the secret key must be able to compute a signature under any additional document.

To link a signed document to a certain *real-world* entity, one has to know whether the public key actually "belongs to" this entity. This is facilitated by *public-key infrastructures* (PKI) (e.g., [75]). The typical solution is to establish one or a few *certification authorities* (CA). Their task is to verify that a certain public key pk belongs to a certain entity X and to sign a digital document, called *certificate,* that confirms this fact to others. The notion of "belong to" is vague, and there have been many discussions on the semantics of certificates. (The better way would have been to make the certificates real statements saying what is meant, instead of essentially pairs of a name and a key.) Naively, one would like a semantics that allows the recipient of a statement with a digital signature to treat it like a statement with a handwritten signature. Even more simplified, this would mean that the certification authority states in the certificate that the entity X is willing to take full responsibility for every statement that carries a correct signature with respect to pk. The phase where X agrees to this responsibility (because such liabilities should not be assigned to a person by third parties, and indeed cannot in most laws) is called *registration* and should involve a handwritten signature by X under a similar statement. The certification authority should verify this signature with respect to an identity document of X to make registration secure.

However, people would currently like to carry out electronic commerce with PCs only, and, as explained above, secret keys on PCs can easily be stolen or misused by attackers (and even smartcards do not help all that much). Hence such a strong liability is not desirable because it would expose consumers and small businesses to an incalculable risk. A special case of this liability question was already discussed for SET in Section 1.1.1, and existing solutions typically have the same all-or-nothing approach. *SEMPER* has come up with a much more flexible approach balancing the needs of all parties; see Chapter 3.[5]

At a more abstract level, keys are often seen as "identities" or "pseudonyms" under which the players can act on the electronic marketplace, and PKIs as a means to bind a digital identity to a real-world one. But one should keep in mind that this is just an analogy: Electronic identities are information, basically secret keys. Thus, unlike real identities, they can be stolen. A criminal who gets a secret key can create perfect counterfeits, where no expert witness can find a difference. Moreover, a criminal can create an arbitrary number of new secret keys, thereby possibly multiplying his own power to commit crime.

1.3.4 Payment Systems

There are a number of different proposals for secure electronic payment systems. One main class follows established paper-based systems where security relies on handwritten signatures by replacing those with digital signatures. Here SET [126], favored by large credit-card companies, is best-known; cheques (e.g., [81]) and home banking (e.g., [181]) are other subclasses. The second main class tries to imitate cash. Here the most interesting implemented system is ecash by DigiCash [156], which, in contrast to others, also simulates the privacy aspect of traditional cash. This may become even more important in the virtual world with its additional threats. Payment models are explained in more detail in Chapter 11; for an overview of techniques and proposals see [12]. *SEMPER* did not work on any new payment systems, but on a mechanism that allows users to handle the given variety in a flexible and secure way.

1.4 The Whole Picture of Electronic Commerce

The security mechanisms for electronic commerce considered so far concern individual *steps*. A payment is one such step, and sending a signed document is another. However, such steps do not exist in isolation in electronic commerce: They must be *linked* so that one can safely refer from one step to another. In other words, business consists of *processes*. This is already true for simple catalogue-based shopping over the Internet as described in Section 1.1.1: For example, the price paid should correspond to that in the catalogue. The buyer does not want to pay without receiving the merchandise, nor the merchant send it without being paid. Furthermore, if the buyer does get the merchandise but it is seriously different from the offer in the catalogue he may want to return it, which may result in a dispute in court. Many variants of such processes are possible, in particular in a business-to-business scenario. For example, there may be a longer negotiation phase; there might be different binding offers for the same good or service; and delivery might be in several installments.

In all such processes there are certain rules that a secure electronic-commerce tool should help to ensure: Many steps process certain *forms,* i.e., documents with fields with a fixed semantics, like "price" or "delivery date." Some rules, e.g., from accounting or tax laws, concern a form as such. Other rules, more important for security, concern the relation between the fields of different forms, e.g., that the price of a payment is the same as in the order, or the semantics of the fields, e.g., that something has indeed arrived at the delivery

[5] The German Digital Signature Act [84] and the European Signature Directive [71] do allow limits on the uses of a certificate; here the *SEMPER* solution could be integrated.

date. Each participant in secure electronic commerce wants his own tool to locally check this for him, or at least to aid him in checking and help him to remedy problems, without relying on the partner to run the same tool securely or simply not to cheat.

This means that records of the steps are needed in a meaningful context, i.e., within the processes they belong to. This is not only desired by both private and commercial users for managing their assets, but also to allow the extraction of meaningful evidence in the case of disputes, and to enable the concept of an "electronic auditor."

The process approach is similar to EDI and projects like XML/EDI or OBI mentioned in Section 1.1.2, but none of those focuses enough on the related multi-party security requirements.

How *SEMPER* and such projects could be integrated is mentioned in Chapter 6.

An additional issue with a process-oriented approach is user authorization. In principle, most steps need to be approved by the users, typically a human user in the case of consumers and small businesses. However, the user might not want to be bothered with each step, but want to give the approval once for an entire process. This means that rights to perform certain steps must be delegated to the user's secure electronic commerce tool, for example by means of *authorization and delegation policies.*

1.5 Resulting Goals of *SEMPER*

1.5.1 Security Requirements

Generalizing from concrete scenarios and concrete threats, and from user interviews in *SEMPER* (see Chapters 7 and 8) and outside surveys, one can see the following main security requirements for electronic commerce:

- Fairness: Ensure that situations like payment without delivery, or delivery without payment, do not happen.
- Authenticity: Ensure that partners cannot impersonate others or, where anonymous, ensure that they cannot act without having the proper rights.
- Availability of service: Ensure that all contracts and promises are fulfilled.
- Privacy: Ensure that partners do not collect or use data for unintended purposes. Ensure that outsiders do not get unnecessary information.
- Prevent misuse of goods, like infringement of copyright and illegal resale of information.

All these requirements are not only made on the internal technical system, but relative to how the real users interact with the system, from the user interface design to the legal environment.

1.5.2 The *SEMPER* Focus

As a result of the view on the current situation and future needs as described so far, *SEMPER* has set the following goals for its specific approach:

- *Entire processes*: *SEMPER* should support all the standard steps, linked into complete business processes as described in Section 1.4, such that the security requirements are fulfilled for the entire processes. For extreme cases, disputes must be supported, i.e., the tool must help to find the necessary evidence and must also help arbiters to evaluate it. Privacy must also be supported for entire processes.
- *Multiple scenarios*: *SEMPER* should be usable in a variety of person-to-business, business-to-business, and person-to-person scenarios. In particular for large businesses, it should be suitable for integration into back-end systems, i.e., for use without human intervention. In other cases, only export and import of data with other programs is necessary. The benefits of using one framework for all these scenarios is decreased development cost and increased confidence in its security due to more intensive scrutiny.
- *Openness*: For many of the services (e.g., payments, signatures, encryption) several protocols and products already coexist and will keep coexisting. The *SEMPER* architecture should be able to integrate them easily. It must also support business partners in selecting which protocol and product to use in a given situation.

It should also be possible to integrate less standard building blocks, e.g., anonymous communication (in particular for privacy in web browsing and prepaid low-value purchases) and time-stamping and other notary services.

- *Ensure that users can securely manage information*: All the security technologies need to interact with the human user (except in back-end integration): He has to agree to transfer a right or to give away certain private information to a certain other party, and thus he has to be convinced about this party's current role and rights. This requires a carefully designed user interface: Users must be made aware of security-critical facts, but still "normal" users should not be bothered with so many details that they start clicking "ok" regardless. Important aspects are a uniform look-and-feel for critical steps, and clear indications of "points of no return" where a negotiation or selection phase ends and a legally meaningful or privacy-critical step is taken. We call such an interface a Trustworthy Interactive Graphical User Interface (TINGUIN).
- *Multi-party security*: *SEMPER* must be a distributed system with no a-priori assumptions that everybody will trust particular entities. Where third-party services are needed, everybody should be free to choose them in the most suitable way, and it should be possible to hold such third parties accountable for their actions.
- *Legal framework*: The technical framework alone is insufficient for establishing a predictable security environment because national laws and regulations do not yet include specific and coherent provisions on electronic commerce. A legal framework, acknowledging the liabilities of digital signatures and the value of electronic records taken during business processes, is required. It should in particular provide clarity in cross-border situations (and situations where only the certification authorities are different). A fair and reasonable distribution of rights and responsibilities should be made that takes into account the current risks and vulnerabilities and can adapt smoothly to changed situations (e.g., more secure user devices) in the future.

Apart from developing the technical and legal frameworks, *SEMPER* also worked on particular protocols where no sufficiently secure or efficient technology was available yet. In the following chapters, we will see how *SEMPER* carried out this approach. In particular, Chapter 2 gives an overview of the technical approach and Chapter 3 of the legal framework, while Chapter 4 shows how this basis can be used in important scenarios. Some highlights are presented in more detail in Part II of this book.

Chapter 2

Technical Framework

This chapter gives an overview of the technical framework of *SEMPER* and how this framework fulfills the goals derived in Chapter 1. The framework covers the whole scope of business processes from the standpoint of security and is flexible enough to integrate existing or future security tools.

2.1 The *SEMPER* Model

The technical provisions of the *SEMPER* framework are primarily designed for the users of the marketplace. Typically their roles are buyers or sellers, but *SEMPER* also supports other business processes besides purchases. The framework is also useful for third parties with security-relevant functions like auctioneers, notaries, and arbiters who evaluate disputes on the basis of digital evidence, but then some role-specific additional software will be needed.

The basis for the technical provisions is the model that

- the security-critical parts of electronic commerce are typically sequences of transfers or exchanges of information, and
- users might question the current state of affairs during or after a business process.

The concepts of transfer and exchange address the first point, while the concept of deal addresses the second one. These concepts contribute to the support of consistent business processes while implementing multi-party security.

A *transfer*, in *SEMPER* terms, achieves the transmission of one or more pieces of information like a document or a payment from one party to one or more recipients. The concept of transfer should be understood at an abstract level. Hence, a transfer may need one or several messages to complete. The model distinguishes three main types of items that may be transferred: (a) payments, (b) signed statements like offers, orders, receipts, contracts, and certificates, and (c) information like intangibles to be delivered. Each item or its transfer may be individually associated with security attributes such as confidentiality or non-repudiation of origin or receipt. Items may be grouped together under the concept of container for the purpose of performing a transfer. Security attributes may also be assigned to a container. For example, an order with its corresponding payment can be packaged in a container which demands non-repudiation of receipt.

When several transfers need to be grouped together to represent the semantics of an indivisible operation among two or more parties, the concept of *exchange* is used. Payment against delivery is an example of an exchange. Typically, exchanges are associated with fairness guarantees so that a party's transferring an item will imply the reception of an item in return: payment against delivery implies that payment is received if and only if delivery occurs. Thus, a fair exchange will either complete successfully or everything will happen as if no exchange took place. (The simplest implementation would be to always send both items to a trusted third party, which only forwards them if they fit the expectations.) Figure 2.1 illustrates typical exchanges in electronic commerce.

A third key concept in *SEMPER* is that of a *deal*. A deal corresponds to a sequence of steps (also called transactions later at a technical level). It addresses the need to cope with an entire business process

	Payment	**Signed statement**	**Information**
Information	Purchase of information	Conditional access, certified mail	Information exchange
Signed statement	Payment with receipt	Contract signing	
Payment	Money exchange		

Figure 2.1: Typical exchanges

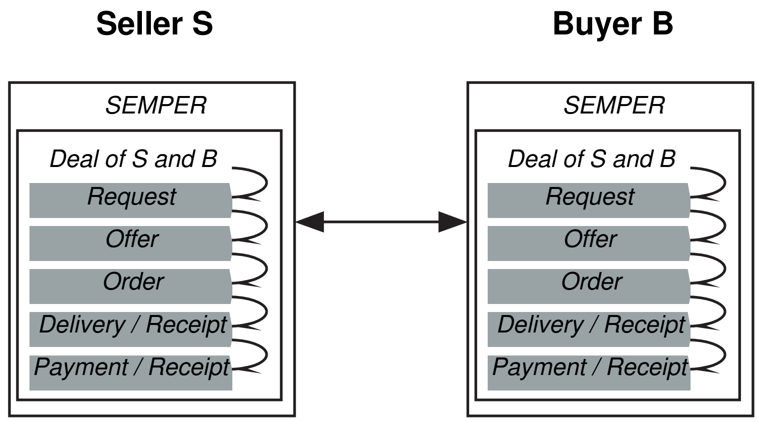

Figure 2.2: A deal: Both parties record the steps and their relations

in a consistent manner, maintaining the link between steps as they occur. A deal is recorded both at the seller's and the buyer's side (or, in other types of business processes, similarly at each participant's side). This is illustrated in Figure 2.2. The record includes the information received and sent during each step, overall information about the process like negotiated security attributes and the digital identity of the parties involved, and the relations among steps.

Based on deal records archived at the buyer's and the seller's, exception handling by the parties themselves and disputes involving an arbiter can take place. By inspection of deal records, securely aided by his own installation of *SEMPER*, an arbiter will evaluate evidence and determine an appropriate resolution.

2.2 Approach

The *SEMPER* model described so far is the basis for fulfilling the goals of considering entire processes and providing multi-party security (see Section 1.5.2). We now consider the *SEMPER* approach to cope with the following three goals: (a) allowing for multiple business scenarios, (b) establishing an open security platform, and (c) ensuring that users can securely manage information.

The first goal, *usability in multiple business scenarios*, is addressed by positioning the main part of *SEMPER* as a piece of middleware, which provides security services to applications implementing business processes.[1] Security services are offered in the style of a security tool box, thereby enabling applications to freely invoke individual services as needed. Hence, an application where one pays after delivery can invoke the information transfer service and the payment service in this order, while other applications could use a fair exchange, or do the payment completely out of band. The division into applications and security services through an application programming interface provides application designers with the flexibility

[1] *SEMPER* has also provided prototypes of standard applications of this middleware part; see Chapter 4.

to cope with as many business scenarios as required. This division also brings them the advantage of shielding their applications from the security-implementation details and their evolution over time. Hence, the approach of *SEMPER* divides the support of business processes into two distinct, but complementary, types of developments: (a) applications, including the integration of back-end systems, and (b) security means. A difference to other security tool boxes is that *SEMPER* provides security means up to a higher, more commerce-oriented level with its transfers, exchanges, and deals, and also with the consequent provisions for meaningful records and dispute resolution.

The second goal, *providing an open security platform*, is addressed through the provision of a set of internal interfaces designed for the integration of existing or future security implementations. The *SEMPER* term for such an exchangeable implementation is a module, e.g., we speak of a crypto module or a payment module. For those implementations that have not been specifically designed for integration in a *SEMPER* platform (in particular implementations of already well-known services like payments), an adapter is required to bridge the implementation's interface with the internal interface provided in the platform. Implementations of newer services, e.g., exchanges, immediately offer a compatible interface for direct integration. In this way, applications can use an open-ended set of security protocols without being aware of their specificity. Users can populate their security tool box with the modules of their choice. The *SEMPER* framework itself performs overall functions like the automatic negotiation!of module of a module that both business partners have in their tool box and that best provides the desired high-level security and quality-of-service attributes.

The introduction of a trustworthy user interface, the TINGUIN, in *SEMPER* addresses the third requirement, *ensuring that users can securely manage information.* It clearly partitions, from the user's standpoint, information into trustworthy and unverified information. The *SEMPER* user interface is designed to be the unique point of interaction between users and their secure platform. A single interface is of paramount importance: Multiple ones, provided by each security implementation integrated in *SEMPER*, would introduce complexity and seriously endanger the user's consistent perception of his security context. This would significantly weaken security. The interface provides the user with means to interact with the other parties involved in electronic commerce, whether a business partner or a third party. It visualizes security-relevant information received such as party authentication, certificates, credentials, signed information, payments, or signed receipts. It empowers users to sign documents like offers, orders, or quotations, to authorize payments or the downloading of a business application, to register with registration authorities, or to request certificates from a certification authority. It also enables users to manage secure information stored locally like past steps of a deal, certificates, preferences, and to securely install new modules on the platform.

2.3 Architecture

The security functions offered by *SEMPER* are structured in three layers and a set of supporting services. The upper layer, named the Commerce Layer, offers security services to the applications and provides the process orientation. It relies on the middle layer for transfers and exchanges. That middle layer, noted the Transfer-and-Exchange Layer, relies itself on the bottom layer, called the Business-Item Layer, for protocols specific to the nature of business items exchanged. Figure 2.3 illustrates the overall architecture of *SEMPER*.

The *Commerce Layer* maintains the status of a deal and records its progress. It offers the business application, and thus finally the user, services to open new deals, to navigate across or within existing deals, and to retrieve and display previous and current steps. Deals can also be exported, e.g., to an arbiter for inspection and resolution of disputes. The arbiter has the same support for displaying and verifying steps of a deal by the Commerce Layer of his own *SEMPER* installation. Opening a deal includes the negotiation of the overall quality of service, in particular the security attributes. Additional security attributes may be set for individual steps. For example, if confidentiality was not set at deal level, it can be added for the delivery of information. Some attributes only make sense for entire deals, e.g., anonymity. In addition, general application-oriented specializations of lower-layer services are placed on the Commerce Layer.

The *Transfer-and-Exchange Layer* manages exchanges and transfers. In an exchange, each party specifies business items in the form of a description of what will be sent, and what is expected in return. Each party also specifies security attributes for the exchange. The exchange will complete only if two conditions are met: (a) the description of business items to be sent match the expectations, and (b) the security attributes required can be met. For the fairness of the exchange, i.e., the both-or-nothing property, the help of a notary

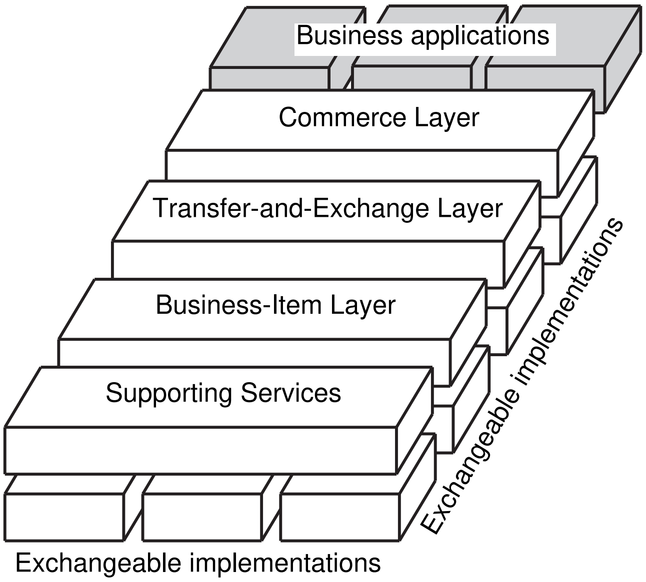

Figure 2.3: *SEMPER* overall architecture

may be requested. To this end, *SEMPER* designed efficient and novel optimistic protocols involving a notary only in case of exceptions.

For transfers, several arbitrary business items can be grouped into containers. Security attributes can be associated to both the container and the information within it. For example, sending a container with non-repudiation of origin will internally lead to a signature of the sending party on aggregated information for the whole container, including items like payments that may involve several rounds of communication.

The *Business-Item Layer* handles transfers according to the nature of the data being sent and received, whether payments, statements, or information. The description of the desired item is passed to the appropriate module for actual transmission and reception through a suitable protocol. For example, a payment to be made on the basis of the SET protocol is passed to a SET module that achieves the necessary three-party communication among the customer, the merchant and the payment gateway. A user can have multiple so-called purses, e.g., for several credit cards and anonymous stored-value payments, and manage and use them consistently with *SEMPER*. In the course of the project, implementations of seven different payment systems were integrated through the interfaces offered by the Business-Item Layer for payments. A similar approach was followed for statements and information.

The *Supporting Services*, including the trustworthy user interface, are available for access by the three main layers. Apart from the user interface, they include various services for cryptography, secure communication, and local bookkeeping. The cryptographic services mainly provide for key generation, encryption, and signatures. There are also provisions for handling certificates both for a public-key infrastructure and for trust management.

The communication services offer communication in a unified way, shielding service use from the specificity of the underlying network. A quality of service parameter enables the selection of the appropriate communication channel. Secure communication is implemented as a natural extension with security attributes as additional quality-of-service parameters. In addition, *SEMPER* provides for anonymous communication in the same framework by an abstract addressing scheme that comprises normal and pseudonymous addresses.

Among the local bookkeeping services, the archiving services manage secure storage and retrieval of certificates, keys, and the information pertaining to deals. The preference services maintain a consistent view of user preferences to modules in *SEMPER*. Access control services protect system and data integrity and confidentiality by verifying the rights of the *SEMPER* components as well as business applications wishing to access to critical resources.

2.4 Protocols and Implementation

The *SEMPER* architecture aims at securing electronic commerce as a whole, and is at present still the only architecture with this ambitious goal. A more technical overall description is given in Chapter 6. There it is also shown how other known projects would fit into our overall architecture. Within the project, a prototype of the architecture was implemented in Java. Trial experiences with this prototype and corresponding business applications are described in Chapters 7 and 8. New protocols and strategies invented in *SEMPER* within this architecture are described in detail in other chapters of Part II. This concerns in particular the payment framework, fair exchanges, trust and liability management and the deal concept.

Chapter 3

Legal Framework

This chapter outlines *SEMPER*'s proposals for establishing legal predictability of electronic transactions. An important novelty is a method for assigning liability to digital signatures in a way that is predictable for all sides without requiring prior investment into highly secure hardware, the Commitment- and Liability-Cover Service. The proposals also cover most other legal aspects of electronic commerce, again with a focus on predictability even in cross-border scenarios, fairness to all parties, and ease of adoption. A model for implementing these proposals is given as the *SEMPER* Electronic-Commerce Agreement, *SECA*.

3.1 Introduction

Electronic commerce promises the ability to perform almost arbitrary transactions over information networks. However, this new support of business has not yet developed in the order of magnitude that was anticipated, partly because legal predictability is still not available. Because their computers are not secure enough, users are still facing many risks, in particular impersonation attacks through stealing private keys or, if the keys are stored on smartcards, through misuse of the smartcard via the insecure computer; see Sections 1.2.3 and 1.3.3. These risks will increase once electronic commerce is commonly used. This means that, technically, digital signatures cannot fully ensure non-repudiation until such attacks are really prevented. Nevertheless, most users wish to use their current PCs for all their actions on the Internet without even bothering to get a smartcard reader and a smartcard, and even less wait a few years for more secure devices.

SEMPER developed a suite of measures that allows all parties, both the holders of signature keys and relying parties (i.e., parties that accept signatures and rely on their validity), to limit their overall risk in a predictable yet flexible way. This represents a market-driven, in contrast to entirely regulation-driven, approach at liability assignment for signatures. It is described in Section 3.2.

Furthermore, while digital signature laws are indeed emerging, different countries are taking different attitudes. For example, the regulations enacted or planned in forty states of the United States have approached digital signatures in about six different ways. (Details about emerging regulations are given in Chapter 14.) Other legal issues like advertising, the applicable law and jurisdiction, consumer-protection laws, privacy and data protection, and copyright and trademark issues, would also need to be harmonized in order to encourage cross-border commerce. *SEMPER* proposes to compensate the current lack of regulations by establishing binding agreements for the immediate use of electronic commerce. Such agreements have a realistic chance to be recognized and accepted by the players and courts if they are clear, fair, comprehensible and practicable. A concrete model agreement is explained in Section 3.3. (Details can be found in Chapter 14.)

3.2 Predictable Liability for Signature Keys

Regulating the liability for digital signatures is the core question in a legal framework for electronic commerce. The concrete objectives in *SEMPER* were:

- To enable a key holder to limit his liability to an acceptable and previously known *overall value* in case the equipment used for signing is compromised.
- To enable relying parties to rely on received signatures regardless of whether the key holder's equipment was compromised.
- Not to require the parties to know each other a priori, or to make paper-based bilateral agreements for this purpose.

Recall that in cases of bilateral relationships that do start with personal contacts, e.g., in home banking or under a trading partner agreement, the first two requirements can be fulfilled quite easily by agreeing on an overall limit on the transaction volume between these partners. With the third requirement, *SEMPER* extends this type of security to the more flexible relationships desired in electronic commerce.

The following main measures achieve these objectives:

- Each party makes an agreement on its concrete liability with its CA (certification authority). This is called an *initial commitment*.
- Online services guarantee to relying parties that the value they rely on is covered by the overall value the key holder is liable for. This is called the *Liability-Cover Service*, short L-Cover Service.

The first measure means that the agreement on liability is made in the one place where an agreement has to be made anyway. Thus there is no significant overhead. In our legal view, the agreement requires a handwritten signature by the person incurring the liability (the owner of the signature key) under the terms of the agreement. For high liabilities, the person's identity and signature should be compared with an identity document. This makes identity stealing, as known with credit cards, as difficult as possible.[1] The CA includes the liability limit in its certificates. Several certificate standards and regulations already allow limits, in particular X.509v3 [102], the German Digital Signature Act [84] and the European Signature Directive [71], but without giving them a clear meaning. (In particular, is the limit per transaction or overall? And in the latter case, how can the relying party know whether it is covered?) These fields can be utilized and made meaningful by the *SEMPER* commitments and liability covers, either by the policies of individual CAs or generally.

3.2.1 Commitments without Online Third Party

For cases where the overhead of the full liability-cover service is not deemed worth while, *SEMPER* proposes two simpler forms of commitment:

- *General liability limit:* The general rule is that a key holder is liable for digital signatures made with this key unless he claims his key was compromised. In that case, the key holder has to satisfy all relying parties that are not covered by higher-class commitments with an overall amount not exceeding the general liability limit he agreed to. Thus the relying parties cannot fully rely on the signatures, but they know that the key holder cannot revoke transactions with impunity.
  Considering that many users do not look at certificate details, the *SEMPER* model agreement *SECA* proposes that this limit could be fixed, depending only on the role of the party (e.g., private or business).
- *Partner-specific liability limit:* For business partners with whom a key holder deals regularly, he can agree to an additional limit on what he would pay this partner if his signature key is compromised. The function is similar to a bilateral agreement between these parties, but the CA mediates the initial agreement, so that the relying party need not verify an identifying document etc.

3.2.2 Liability-Cover Service

If a relying party requires a certain fixed liability per signature and the key holder has not committed to a partner-specific liability limit in advance, the full liability-cover service is needed. The key holder initially commits to a (freely chosen) overall liability limit for this service, e.g., on a certain value per month, to its CA providing this service. For a specific signature, the key holder contacts the CA and requests a commitment

[1] However, in any case, a key certified with an overall liability limit is less dangerous for its supposed owner than one without. Hence the requirements on correct identification can be less strict if anything.

for the amount needed by the relying party. The request is digitally signed. Upon receiving a valid request, the CA verifies that the amount is still within the overall limit and adds this amount to the total so far. The CA then signs a specific liability-cover certificate, which uniquely refers to the transaction to prevent re-use for other transactions. Upon receipt of a signed message and a corresponding liability-cover certificate, the relying party verifies that all signatures are valid and that the committed amount is sufficient for the purpose. Details are presented in Chapter 13.

3.2.3 Security and Market Effectiveness

For security, the important point is that none of the commitment types allows a criminal who has gained access to a signature key to increase the key holder's liability for this key by using the key itself.

The service fulfills multi-party security, i.e., nobody needs to trust the CAs unduly: A key holder is protected by the paper-based commitment procedure to the liability limit. (If criminals can subvert this procedure, they can cheat, but the procedure was required to be equally secure as other commitments to similar amounts in paper-based commerce.) The relying parties obtain clear guarantees in the certificates, and if the CA gave out certificates exceeding the key holder's overall limit, or certificates for which it cannot show the key holder's signed request, the CA should be liable.[2] The remaining, predictable risks may be insured at the discretion of the parties and insurance companies.[3]

The market incentive for using a liability-cover service is that recipients of signatures (typically service providers) will require non-repudiation, i.e., liability of the key holder, in some situations, or offer favorable conditions to key holders giving such guarantees. This will encourage key holders (typically customers) to agree to such liabilities with limits suitable to their financial status, if they can reasonably hope to recover the cost, bother, and risk of a getting a certificate. An organization of potential recipients might also sponsor CAs to make certificates with liability more attractive to customers immediately. Market forces can then also drive the development of more secure devices: When compromising keys with certain liabilities becomes attractive enough for criminals so that it happens often, the incentives that key holders need before agreeing to such liabilities will increase, and at some point buying more secure devices will be cheaper on average.

3.3 The *SEMPER* Electronic-Commerce Agreement

Besides the legal recognition of digital signatures, many other legal issues would need harmonized regulations to make electronic commerce predictable, in particular across borders:

- applicable law and jurisdiction,
- export regulations for cryptographic products,
- consumer protection (in the sense of protecting weaker partners from being taken advantage of),
- privacy and data protection,
- advertising, competition, and spamming (i.e., junk mail),
- illegal content of offers and Internet pages,
- contract law (in the sense of formal requirements),
- copyright and trademark, and
- payment and taxation.

A promising solution to compensate the lack of regulation immediately is to establish model agreements to which parties voluntarily promise to adhere. Agreements have a realistic chance to be accepted by the players and approved in court if they are clear, fair, comprehensible and practicable. *SEMPER* proposes a model agreement for this purpose, called *SECA* (*SEMPER* Electronic-Commerce Agreement). An additional advantage of such an agreement is that it can enhance the awareness of users as a means to reduce risks.

[2] However, it should not be liable for the key holders' financial abilities to fulfill their commitments—this is not the job of a legal framework, but of a credit-rating agency.

[3] In contrast, it is unlikely that insurance companies will insure key compromise without liability limits in the long run, as they cannot verify whether such a claim is true or not (similar to problems with insurances against luggage theft). Excluding key holders whose key is compromised more than, say, twice is not a solution because a criminal might indeed attack the same target several times.

3.3.1 Structure of *SECA*

SECA comprises three parts: the agreement itself that needs to be signed, a code of conduct that the signatories promise to follow, and guidelines that give advice for reducing risks due to failures and attacks and try to establish security and fairness standards for products.

Agreement The first section of the agreement defines the scope of the agreement in terms of areas of participation and applicable transactions, and contains the obligation to follow the code of conduct. The second section regulates contracts concluded by parties having agreed to *SECA* and refers to the default values defined in the code of conduct like the applicable law and jurisdiction. This section also includes provisions for conflicts of *SECA* with other regulations the parties wish to establish. The third section regulates the liability for digital signatures as explained above. The fourth section addresses the case of parties not adhering to the agreement although they agreed to it, including revocation and blacklisting. The fifth section contains general provisions, e.g., for the modification of *SECA* and cases of inapplicability.

Code of conduct The code of conduct is designed to ensure fairness in transactions and to facilitate cross-border commerce. It sets default values of applicable law and jurisdiction according to the relationship between two parties: private-to-private, business-to-business, or business-to-private. The interests of the consumer, as the weaker party, are favored. The code of conduct also includes regulations for fair business concerning advertising and competition, negotiation, offers and acceptances, consumer protection, content of contracts and Internet pages, and privacy and data protection. The few remaining items from the list above are better left to national law.

Guidelines The first section of the guidelines introduces the concept of *SECA*-compliant components such as electronic-commerce software, operating systems, and secure hardware. The criteria for electronic-commerce software emphasizes user friendliness and secure visualization, and creation and handling of evidence. They are a more detailed version of the requirements "entire processes," "ensuring that users can securely manage information" and "multi-party security" listed in Section 1.5.2 as a basis for the technical development in *SEMPER*. Aiming at flexibility with respect to current and future technology, the model agreement does not try to bar other components, but recognizes *SECA* compliance as helpful for risk management.

The second section is dedicated to recommendations to users for protecting themselves. It addresses the use of fair electronic-commerce software and secure hardware and operating systems; the need of backups and secure upgrading; the generation and use of keys; liability and advice in case keys are compromised; housekeeping with respect to records created by electronic-commerce software; and recommendations for commercial parties running a server.

Certificates An agreement like *SECA* contains a number of parameters characterizing the signatory. Hence each CA should offer several different certificates for the same key, each with a different subset of these parameters. The signatory can then choose how many parameters to show to each partner. This achieves flexible privacy of personal data or business parameters. Of course, care must be taken such that essential parameters like liability limits cannot be omitted or have a suitable default (here zero).

The full details of *SECA* can be found in Chapter 14.

3.3.2 Introducing Electronic-Commerce Agreements

A model agreement like *SECA* can easily be introduced in practice because it is only an agreement between a party and a CA. In fact, it consists of two successive unilateral statements: First the party (a person or a representative of a legal entity) declares its agreement in a traditional way to the CA; then the CA declares this fact in a certificate. *SECA* contains sufficient provision for the case where such a party does business with parties who did not sign a similar agreement. Hence it can be useful right from the start and offer a business advantage over competitors who did not sign such a clarifying agreement.

Nevertheless, an agreement is the more useful the more widely it is accepted. An obvious reason is that one also wants one's trading partner to have agreed to certain fairness rules. In addition, wide-spread use of the same agreement means that people no longer need to read the small print each time. Hence groups of CAs or Chambers of Commerce would be well advised to provide *SECA* or a similar agreement as a default. This would imply tasks like maintaining *SECA* templates, disseminating information on the legal situation in different countries, publishing key certificates of CAs supporting the agreement, and providing black lists of parties violating *SECA* rules they had agreed to.

Finally, it is useful to adapt a technical framework and a legal framework to each other as in *SEMPER* because the user's electronic-commerce tool can then securely visualize important legal aspects. For instance, the *SEMPER* TINGUIN shows at the beginning of a deal whether the partner also signed *SECA* and can zoom to certain parameters, e.g., the partner's liability limits.

3.4 Conclusions

No solution—technical, legal or organizational—can currently remove all risks in electronic commerce. Nevertheless, voluntary agreements that each party individually makes towards its CA, hopefully based on a small number of model agreements, can greatly increase legal predictability at once. We developed the model agreement *SECA* for this purpose. A particularly important aspect is the regulation of liability for digital signatures. Here, *SEMPER*'s liability limits and online Liability-Cover Service provide a good compromise for users willing to accept a certain liability. This may be a critical step to significantly increase electronic-commerce activities, including those crossing borders, while international legal frameworks are being progressed.

Chapter 4

Vision of Future Products

In this chapter, we describe how the *SEMPER* framework can be used in various types of commercial products. Besides that, we believe that free distribution of a basic client version would be a useful investment in infrastructure, similar to browsers, and we describe a prototype of such a tool, the Fair Internet Trader (FIT).

4.1 Four Facets of *SEMPER* as a Product

In the course of the project, we have implemented a prototype of the *SEMPER* framework and several business applications, used them in several trials (see Chapters 7 and 8), and demonstrated it at commercial trade fairs and scientific conferences. Building upon this experience, we developed a vision of future products for secure electronic commerce based on the *SEMPER* framework. This does not mean that we propose to use precisely our prototype as a commercial product—product-quality code according to industrial standards was out of reach for us as a research project. But we do propose to implement new products within our technical and legal frameworks. Our vision of *SEMPER*-based products has four facets:

New business applications Concrete business applications can easily and securely be implemented on top of the *SEMPER* framework. This minimizes time-to-market for new services, as well as security-related costs and technical difficulties.

Naturally all business applications at one site should use the same *SEMPER* installation. But as a strategy for introducing the framework, or in order to differentiate specific applications on the market, the framework can be bundled with the business application. For instance, we did this with the Fair Internet Trader presented in Section 4.2.2.

Facilitating the development and deployment of secure business applications has been the main motivation for *SEMPER* and will also be the main motivation for any *SEMPER*-related product. We therefore discuss this facet in more detail in Section 4.2.

New security implementations New implementations of certain services, e.g., new payment instruments or new contract-signing tools, can easily be integrated into the framework, making them instantly accessible for all existing business applications (recall Section 2.2 and Figure 2.3). Moreover, developing new implementations within the framework is likely to be much easier and cheaper than developing them outside because one can build upon existing services, in particular those of the lower layers of the architecture.

The framework itself The previous two facets presuppose that the framework itself is on the market. There are several economic ways for supporting the framework as a product:

- *Component:* The *SEMPER* framework can be offered as a new component to be integrated into existing, general merchant-server products. This allows for a gradual deployment of the *SEMPER* approach; for instance, the IBM Payment Server is based on the payment framework only (see Chapter 11).

- *Middleware:* The *SEMPER* framework can be provided as typical middleware like CORBA. As with most middleware products, the main motivation for producing it would be a general cost reduction in industry, which requires the framework to become an industry standard. The best strategy to achieve this goal would be to put a product-quality implementation into the public domain. A direct source of revenue would be the provision of technical support services.
- *Preinstalled software:* The *SEMPER* framework can be seen as an extension to a bundle of standard software initially delivered with a device. A particularly interesting market segment is mobile devices like personal digital assistants (PDAs): First, this is the most likely form factor for more secure user devices in the long run. Secondly, porting *SEMPER* to mobile devices opens up many functional options: The same security framework with a similar look-and-feel (not exactly the same because of the smaller displays) can then also be used, e.g., at Internet kiosks, at counters to buy airline tickets, at stock exchanges to take over a company, and for accessing one's private or business resources remotely when traveling.

Specific parts alone Specific aspects of *SEMPER* can be turned into products or commercial services independently of the overall framework. The following two are the most obvious choices:

- *Exchanges: SEMPER* developed new solutions for classical problems such as fair contract signing or certified mail that are more efficient than all previously known solutions (see Chapter 10 for more details). These problems are at the core of many business processes, and anybody who designs such systems could benefit from our solutions.
- *Legal framework and certificates: SEMPER* developed new approaches to the legal problems of electronic commerce and technical solutions support these approaches by making the financial risks predictable (see Chapter 3). First, one can sell these solutions as products. Secondly, one can offer the third-party services commercially, e.g., the certificates with liability limits and the liability-cover service, Seeing what prices certification authorities currently try to ask from clients even for certificates where the client has unlimited liability and the authority none, *SEMPER*-type certificates should be quite valuable. Moreover, they should not increase the costs or the risk for existing certification authorities if those already operate in a secure way.

4.2 *SEMPER*-based Business Applications

The rest of this chapter discusses the first, and in our opinion most important facet of our vision of future products: the development of new and attractive business applications on top of *SEMPER*. Therefore, assume for the moment that the technical and legal *SEMPER* framework were available commercially.

In the course of the project we prototyped about 10 different business applications. They fell roughly into two classes, discussed in the following two sections. We strongly believe that any similar business application and many others can be implemented with the same benefits on top of the *SEMPER* framework.

4.2.1 Secure Internet Shopping

This is the standard scenario of Internet-based electronic commerce as described in Section 1.1.1: A human buyer, sitting in front of his or her PC, browses the World-Wide Web and wishes to buy something from a remote merchant. "Something" can mean tangible goods such as books or computers, or intangible goods such as information from databases or the right to attend a certain course over the Internet.

Usually, the seller's side is fully automated and does not require any human intervention in order to perform a sale. All decisions, e.g., whether a requested good is on stock, are taken by querying the seller's back-end system.

The buyer's side is manually operated. The information-gathering phase of electronic commerce, like searching for suitable sellers and browsing catalogues, is done via the browser only. Only legally relevant actions are controlled and performed via the *SEMPER* framework. For instance, as soon as the buyer requests a binding offer, the *SEMPER* user interface is started. This happens simply by clicking on something like "yes, send me a binding offer" on the merchant's page, which links to the *SEMPER* application on the buyer's computer. All the subsequent security-relevant user interactions are done via this interface. All the necessary services are already available in the framework:

- Business documents can be signed and used later on, e.g., to justify after-sales services or as a basis for disputes at court (digital signature, non-repudiation of origin, dispute handling).
- Receipts for documents and payments are ensured: even dishonest, malicious business partners cannot get hold of documents or money without acknowledging their receipt (fair exchange, certified mail, contract signing).
- The framework does not prescribe specific implementations of these services: this can be negotiated at business time or configured per business application, partner or user. In particular, business partners can dynamically negotiate which payment instruments they want to use. New implementations can be integrated efficiently. In particular there is no need to redo the back-end integration on the merchant's side once it has been done.
- Users can act on the marketplace in confidence: documents can be encrypted so that only the intended recipient can read them. Where desired, users can even act anonymously by using pseudonyms, anonymous payment instruments, and means for anonymous communication.
- The dedicated user interface encourages users to carefully watch what is displayed and entered there. This raises the user's attention in case he or she is asked to digitally sign something, which is an advantage for all parties as users cannot claim not to have understood the action.
- The user can influence the risk he or she is taking in several ways, e.g., by referring to the *SEMPER* Electronic Commerce Agreement and setting liability limits for digital signatures.
- The overall legal context can also be determined through application of the *SEMPER* Electronic Commerce Agreement (see Chapter 3).

As this kind of business is largely standardized, it would be very simple to write business applications that can be integrated in existing merchant-server products, respectively to provide implementations of standards like OTP [136] and OBI [134] with *enhanced security.*

4.2.2 Person-to-Person Scenario: The Fair Internet Trader

The Fair Internet Trader (FIT) is *SEMPER*'s prototype for scenarios that require human interaction at both ends. We believe that this is an important and still neglected type of electronic commerce, in particular for small businesses offering or needing specialized services. Other areas of applicability range from transactions between private persons, like the occasional sale of used furniture or stamps, to high-value transactions like public procurement. In addition, the FIT could be used as a standardized client-side tool for typical Internet shopping as explained above, while providing more flexibility than, e.g., OBI.

The parties involved might have a pre-established business relation, like in procurement, or might meet spontaneously on the Internet just for one transaction. However, to enable certain applications, users of the FIT need to have registered with a certification authority, as explained in Chapter 3.

A typical example of a FIT application could involve a professional translator and her customer, an author of scientific books. Author and translator negotiate a commission to translate a certain book within a certain period of time. The original book is part of the contract, which is digitally signed by both parties. Thus the subject of the commission is indisputable. The translation will be fairly exchanged for a receipt, and time-stamped. This avoids disputes about whether the task was completed in time or not. The author will then want to take a look at the translation before paying. Of course, the payment or the delivery of the translation could also be done by traditional means. For their information, or in case of a dispute, the parties can review all transactions, and if necessary extract and show evidence in court. In particular, if the author does not pay, the translator can show the contract and the receipt to an arbiter. The receipt can even uniquely characterize the received information (in contrast to classical certified mail) in case there is a dispute about what translation was received.

Any FIT interaction follows a pre-defined flow with certain variants, i.e., the FIT guides both parties through a deal. Most of the interactions take place via filling in pre-defined forms on-screen and exchanging them over the Internet, *securely* and *fairly* via the *SEMPER* framework. Thus, nobody can cheat the other party without leaving a trail that can be used at court. Using pre-defined forms has several advantages:

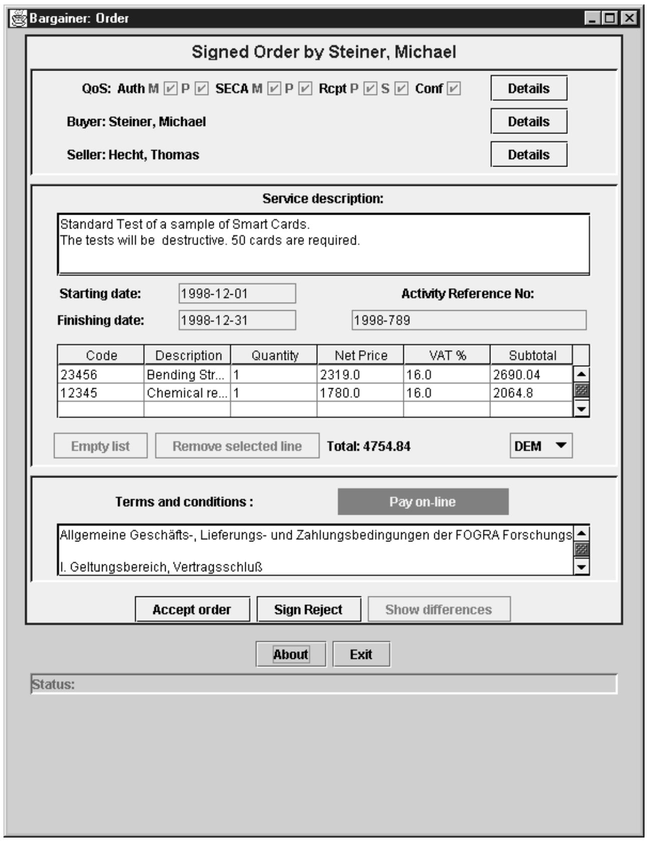

Figure 4.1: A signed order in a business process in the FIT prototype

- Forms are a familiar tool for structuring business processes and workflows in the physical world. Most users have an a-priori understanding of which fields should or must be on a form. As soon as a user knows a form like an "order," it becomes easy to verify the essential fields, like applicable law, reference to terms and conditions, delivery address, signatories, and amounts. Moreover, fields with a well-defined business semantic prevent some disputes a priori that could arise from omissions, imprecision and ambiguities which can occur with digital signatures on unstructured text such as email. Figure 4.1 shows a signed order as visualized in a business process in the prototype.
- Fields can be named and described in several languages, making National Language Support for all business partners fairly easy.
- Data of corresponding fields are automatically copied from one form to another (e.g., from "offer" to "order") as the default. Any party can then change those data that it really wants to change. The other party can have all changes highlighted by its local FIT. This makes negotiation significantly simpler and more secure.
- An "electronic auditor" can verify that the forms are filled in according to basic legal or accounting rules, e.g., that an "offer" contains a sender, a date, a description of contents, an amount, a currency, and a signature.
- Forms with fixed semantics can be imported from and exported to other programs, like accounting, fulfillment or banking programs. All transactions can be recorded, and records include signatures and receipts. Thus, arbiters can base their decisions on the evidence of these records.

Using predefined forms does not preclude fields on forms that are user-defined. The delivery of intangible goods can be done by exchanging files through the *SEMPER* framework as a kind of attachments to the forms; these files can be encrypted, signed and fairly exchanged as well.

Obviously, such a forms-based Fair Internet Trader is much more powerful than, e.g., using PGP or

S/MIME to sign and encrypt standard electronic mail. On the other hand, it is more flexible and comprehensive than other proposed standards like OBI and OTP, and more worked-out than yet others like XML/EDI are so far. Nevertheless, synergy with several other standards is possible, in particular with EDI-based solutions for the standard fields on a large number of forms.

The FIT prototype provides a realistic vision of a *SEMPER*-based product for person-to-person electronic commerce. It also supports the use of *SECA* (see Section 3) and of liability limits. For more details see Chapter 8.

4.3 Outlook

We have shown that the *SEMPER* framework provides a good basis for efficient development and deployment of applications for secure electronic commerce. It offers a lot of market opportunities for the providers of both security technology and business software. Once being deployed, it will help to bring about the envisioned economic benefits of increased electronic commerce. Last but not least it has the social value of allowing anyone to make use of the benefits of the information society in a fair, secure and flexible way.

Part II

Project Achievements

Part I gave an overview of the concepts and results developed by SEMPER. Part II now goes into details. As it is impossible to present all results of a project like SEMPER in detail within one book, we had to be selective. A more complete description can be found in the public project deliverables (see **http://www.semper.org***).*

Chapter 5 briefly describes the organizational structure of the project and general lessons we learned.

Chapter 6 gives a more detailed presentation of the technical framework and architecture than Chapter 2.

Chapters 7 and 8 report on practical experiences with the SEMPER prototype. Chapter 7 describes trials with special SEMPER-based business applications for different sellers in business-to-consumer and business-to-business scenarios. Chapter 8 describes the generic Fair Internet Trader.

Chapters 9–12 elaborate on particular aspects of the technical architecture, starting at the top. The two higher layers, the Commerce Layer and the Transfer-and-Exchange Layer, are described in Chapters 9 and 10, respectively. Chapter 11 describes how SEMPER integrates different payment systems. It also serves as an example of the internal structure of a SEMPER part. Chapter 12 describes public-key infrastructure and why local trust management for certificates is necessary in electronic commerce.

Chapters 13 and 14 describe SEMPER's proposal for fair liability distribution for digital signatures and SEMPER's legal framework in general.

Chapter 15 concludes Part II with an outlook on future directions in research, development, and legal issues in secure electronic commerce.

The chapters in Part II are not contained in this public version of the final report. They are only available in the book which will be published as Volume 1854 in the Lecture Notes in Computer Science (LNCS) Series by Springer-Verlag, Heidelberg in 2000.

Bibliography

[1] Jose L. Abad-Peiro, N. Asokan, Michael Steiner, and Michael Waidner. Designing a generic payment service. *IBM Systems Journal*, 37(1):72–88, 1998.

[2] Masayuki Abe. Universally verifiable mix-net with verification work independent of the number of mix-servers. In *Advances in Cryptology — Eurocrypt '98* [96], pages 437–447.

[3] Hal Abelson, Ross Anderson, Steven M. Bellovin, Josh Benaloh, Matt Blaze, Whitfield Diffie, John Gilmore, Peter G. Neumann, Ronald L. Rivest, Jeffery I. Schiller, and Bruce Schneier. The risk of key recovery, key escrow, and trusted third-party encryption. *The World Wide Web Journal*, 2(3):241–257, 1997.

[4] Carlisle Adams, Patrick Cain, Denis Pinkas, and Robert Zuccherato. Internet X.509 public key infrastructure time stamp protocol (TSP). Internet Draft, June 1999. draft-ietf-pkix-time-stamp-02.txt.

[5] Carlisle Adams and Robert Zuccherato. Notary protocols. Internet Draft, February 1997. draft-adams-notary-01.txt.

[6] William Aiello, Sachin Lodha, and Rafail Ostrovsky. Fast digital identity revocation. In Crypto98 [95], pages 137–152.

[7] Ross Anderson and Markus Kuhn. Tamper resistance — A cautionary note. In *Second USENIX Workshop on Electronic Commerce* [173], pages 1–12.

[8] ANSI Accredited Standards Committee X.12 (ASC X.12/DISA). X12 standard, version 3. American National Standards (ANS), 1992.

[9] N. Asokan. *Fairness in Electronic Commerce.* PhD thesis, University of Waterloo, May 1998.

[10] N. Asokan, Els Van Herreweghen, and Michael Steiner. Towards a framework for handling disputes in payment systems. In *Third USENIX Workshop on Electronic Commerce* [174], pages 187–202.

[11] N. Asokan, Phil Janson, Michael Steiner, and Michael Waidner. State of the art in electronic payment systems. *IEEE Computer*, 30(9):28–35, September 1997.

[12] N. Asokan, Phil Janson, Michael Steiner, and Michael Waidner. State of the art in electronic payment systems. In Marvin V. Zelkowitz, editor, *Advances in Computers*, volume 53. Academic Press, March 2000. This is an extended and revised version of [11].

[13] N. Asokan, Matthias Schunter, and Michael Waidner. Optimistic protocols for multi-party fair exchange. Technical Report RZ 2892, IBM Zürich Research Laboratory, November 1996.

[14] N. Asokan, Matthias Schunter, and Michael Waidner. Optimistic protocols for fair exchange. In *4th ACM Conference on Computer and Communications Security*, pages 6–17, Zürich, 1997. ACM Press.

[15] N. Asokan, Victor Shoup, and Michael Waidner. Asynchronous protocols for optimistic fair exchange. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 86–99, Oakland, CA, 1998. IEEE TC S&P, IEEE Computer Society Press.

[16] N. Asokan, Victor Shoup, and Michael Waidner. Optimistic fair exchange of digital signatures. In *Advances in Cryptology — Eurocrypt '98* [96], pages 591–606.

[17] Randall Atkinson. Security architecture for the Internet Protocol. Internet Request for Comment (RFC) 1825, IETF, 1995.

[18] Vijayalakshmi Atluri and Wei-Kuang Huang. An authorization model for workflows. In E. Bertino, H. Kurth, G. Martella, and E. Montolivo, editors, *Proceedings of the Fourth European Symposium on Research in Computer Security (ESORICS)*, number 1146 in Lecture Notes in Computer Science, pages 44–64, Rome, Italy, 1996. Springer-Verlag, Berlin Germany.

[19] Alireza Bahreman. Generic electronic payment services: Framework and functional specification. In *Second USENIX Workshop on Electronic Commerce* [173], pages 87–103.

[20] Alireza Bahreman and Rajkumar Narayanaswamy. Payment method negotiation service. In *Second USENIX Workshop on Electronic Commerce* [173], pages 299–314.

[21] Alireza Bahreman and Doug Tygar. Certified electronic mail. In *1994 Symposium on Network and Distributed Systems Security (NDSS 94)*. Internet Society, IEEE Press, 1994.

[22] Yannis Bakos. The emerging role of electronic marketplaces on the Internet. *Communications of the ACM*, 41(8):35–42, 1998.

[23] Dave Bayer, Stuart Haber, and W. Scott Stornetta. Improving the efficiency and reliability of digital time-stamping. In *Sequences II: Methods in Communication, Security, and Computer Science*, pages 329–334, Berlin, 1993. Springer-Verlag.

[24] Mihir Bellare, Juan Garay, Ralf Hauser, Amir Herzberg, Hugo Krawczyk, Michael Steiner, Gene Tsudik, Els Van Herreweghen, and Michael Waidner. Design, implementation and deployment of the iKP secure electronic payment system. *IEEE Journal on Selected Areas in Communications*, 18(4):611–627, April 2000.

[25] Michael Ben-Or, Oded Goldreich, Silvio Micali, and Ron L. Rivest. A fair protocol for signing contracts. *IEEE Transactions on Information Theory*, 36(1):40–46, 1990.

[26] Tim Berners-Lee, Roy T. Fielding, Henrik Frystyk Nielsen, Jim Gettys, and Jeff Mogul. Hypertext transfer protocol — HTTP/1.1. Internet Request for Comment (RFC) 2068, IETF, 1997.

[27] Andreas Bertsch. On sustainable digital signatures. In Günter Müller and Kai Rannenberg, editors, *Multilateral Security in Communications, Vol. 3*, pages 269–282. Addison-Wesley, München, Germany, 1999.

[28] C. Bradford Biddle. Comment: Misplaced priorities: The Utah digital signature act and liability allocation in a public key infrastructure. *San Diego Law Review*, 33:1143–1193, November 1996. Available from `http://www.acusd.edu/~biddle/`.

[29] Joachim Biskup and Christian Eckert. About the enforcement of state dependent security specifications. In *Database Security VII: Status and Prospects*, pages 3–17. North Holland, 1994.

[30] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of the IEEE Symposium on Research in Security and Privacy* [98], pages 164–173.

[31] Matt Blaze, Joan Feigenbaum, and Angelos D. Leromytis. Keynote: Trust management for public-key infrastructures. In Bruce Christianson, Bruno Crispo, William S. Harbison, and Michael Roe, editors, *Security Protocols—6th International Workshop*, volume 1550 of *Lecture Notes in Computer Science*, pages 59–66, Cambridge, United Kingdom, 1999. Springer-Verlag, Berlin Germany.

[32] Daniel Bleichenbacher, Eran Gabber, Phillip B. Gibbons, Yossi Matias, and Alain Mayer. On secure and pseudonymous client-relationships with multiple servers. In *Third USENIX Workshop on Electronic Commerce* [174], pages 99–108.

[33] Manuel Blum. Three applications of the oblivious transfer. Technical Report Version 2, University of California at Berkeley, 1981.

[34] Jean-Paul Boly, Antoon Bosselaers, Ronald Cramer, Rolf Michelsen, Stig Mjølsnes, Frank Muller, Torben Pedersen, Birgit Pfitzmann, Peter de Rooij, Berry Schoenmakers, Matthias Schunter, Luc Vallée, and Michael Waidner. The ESPRIT project CAFE — High security digital payment systems. In Dieter Gollmann, editor, *Proceedings of the Third European Symposium on Research in Computer Security (ESORICS)*, number 875 in Lecture Notes in Computer Science, pages 217–230, Brighton, UK, 1994. Springer-Verlag, Berlin Germany.

[35] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults. In *Advances in Cryptology — Eurocrypt '97*, number 1233 in Lecture Notes in Computer Science, pages 37–51. IACR, Springer-Verlag, Berlin Germany, 1997.

[36] Ahto Buldas, Peeter Laud, Helger Lipmaa, and Jan Villemson. Time-stamping with binary linking schemes. In Crypto98 [95], pages 486–501.

[37] Holger Bürk and Andreas Pfitzmann. Digital payment systems enabling security and unobservability. *Computers & Security*, 8(5):399–416, August 1989.

[38] Holger Bürk and Andreas Pfitzmann. Value exchange systems enabling security and unobservability. *Computers & Security*, 9(8):715–721, 1990.

[39] Suresh Chari, Charanjit Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Crypto99 [97], pages 398–412.

[40] David Chaum. The dining cryptographers problem: Unconditional sender and recipient untraceability. *Journal of Cryptology*, I(1):65–75, 1998.

[41] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–88, February 1981.

[42] David L. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[43] David D. Clark and David R. Wilson. A comparison of commercial and military computer security policies. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 184–194, Oakland, CA, 1987. IEEE TC S&P, IEEE Computer Society Press.

[44] Tom Coffey and Puneet Saidha. Non-repudiation with mandatory proof of receipt. *Computer Communication Review*, 26(1):6–17, 1996.

[45] Common Criteria Project Sponsoring Organisations. *Common Criteria for Information Technology Security Evaluation*, August 1999. Version 2.1, adopted by ISO/IEC as ISO/IEC International Standard (IS) 15408 1-3. Available from `http://csrc.ncsl.nist.gov/cc/ccv20/ccv2list.htm`.

[46] Benjamin Cox, Doug Tygar, and Marvin Sirbu. Netbill security and transaction protocol. In *First USENIX Workshop on Electronic Commerce* [172], pages 77–88.

[47] Ronald Cramer and Ivan Damgård. New generation of secure and practical rsa-based signatures. In *Advances in Cryptology — Crypto '96* [94], pages 173–185.

[48] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Crypto98 [95], pages 13–25.

[49] Lorrie Faith Cranor and Brian A. LaMacchia. Spam! *Communications of the ACM*, 41(8):74–83, 1998.

[50] Neil Daswani, Dan Boneh, Hector Garcia-Molina, Steven Ketchpel, and Andreas Paepcke. A generalized digital wallet architecture. Technical report, Stanford University, Computer Science Department, 1998.

[51] Drew Dean, Edward W. Felten, and Dan S. Wallach. Java security: From HotJava to Netscape and beyond. In *Proceedings of the IEEE Symposium on Research in Security and Privacy* [98], pages 190–200.

[52] Drew Dean, Edward W. Felten, Dan S. Wallach, and Dirk Balfanz. Java security: Web browsers and beyond. In Denning and Denning [54], pages 241–269.

[53] Robert H. Deng, Li Gong, Aurel A. Lazar, and Weiguo Wang. Practical protocols for certified electronic mail. *Journal of Network and Systems Management*, 4(3):279–297, 1996.

[54] Dorothy E. Denning and Peter J. Denning, editors. *Internet Besieged: Countering Cyberspace Scofflaws.* ACM Press / Addison-Wesley, New York, 1998.

[55] Peter J. Denning. *Computers under Attack - Intruders, Worms and Viruses.* ACM Press, New York, 1990.

[56] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[57] Donald E. Eastlake. Domain name system security extensions. Internet Request for Comment (RFC) 2535, IETF, 1999.

[58] Donald E. Eastlake, Brian Boesch, Steve Crocker, and Magdalena Yesil. CyberCash credit card protocol version 0.8. Internet Draft, July 1995. draft-eastlake-cybercash-v08-00.txt.

[59] Donald E. Eastlake, Stephan D. Crocker, and Jeffrey I. Schiller. Randomness requirements for security. Internet Request for Comment (RFC) 1750, IETF, 1994.

[60] European Union. The OECD guidelines for the security of information systems, 1992.

[61] European Union. Directive 93/13/EEC of 5 April 1993 on unfair terms in consumer contracts, April 1993.

[62] European Union. Council regulation (EC) no 3381/94, December 1994. Setting up a community regime for the control of exports of dual-use goods.

[63] European Union. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, October 1995.

[64] European Union. Communication on illegal and harmful content on the Internet. COM(96) 0487 - C4-0592/96, 1996.

[65] European Union. Directive 97/55/EC of European Parliament and of the Council of 6 October 1997 amending directive 84/450/EEC concerning misleading advertising so as to include comparative advertising, October 1997.

[66] European Union. Directive 97/7/EC of the European Parliament and of the Council of 20 May 1997 on the protection of consumers in respect of distance contracts, May 1997.

[67] European Union. Ensuring security and trust in electronic communication: Towards a european framework for digital signatures and encryption. Communication to the European Parliament, the Council, the Economic and Social Committee and the Committee of the Regions, COM(97)503, November 8 1997.

[68] European Union. EU directives 91/250, 96/9 and 92/100 on copyright and related rights in the information society, December 1997.

[69] European Union. A european initiative in electronic commerce. Communication to the European Parliament, the Council, the Economic and Social Committee and the Committee of the Regions, COM(97) 157, April 15 1997.

[70] European Union. The OECD cryptography policy guidelines (1997), 1997.

[71] European Union. Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a community framework for electronic signatures. *Official Journal of the European Communities*, L 13:12–20, January 19 2000. Available from `http://www.qlinks.net/comdocs/elsig/en.pdf`.

[72] Shimon Even. A protocol for signing contracts. *ACM SIGACT News*, 15(1):34–39, 1983.

[73] Shimon Even and Yacov Yacobi. Relations among public key signature systems. Technical Report 175, Computer Science Department, Technion, Haifa, Israel, March 1980.

[74] Joan Feigenbaum. Towards an infrastructure for authorization (position paper). In *Third USENIX Workshop on Electronic Commerce* [174]. (not in printed version of proceedings; available from `http://www.research.att.com/~jf/pubs/usenix-ecommerce98.ps`).

[75] Warwick Ford and Michael S. Baum. *Secure Electronic Commerce*. Prentice Hall, Upper Saddle River, New Jersey, USA, 1997.

[76] Martin Fowler and Kendall Scott. *UML Distilled : Applying the standard object modeling language*. Addison-Welsey, Reading MA, 1997.

[77] Dirk Fox. Zu einem prinzipiellen Problem digitaler Signaturen. *Datenschutz und Datensicherheit DuD*, 22(7):386–388, 1998.

[78] Matthew K. Franklin and Michael K. Reiter. The design and implementation of a secure auction service. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, pages 2–14, Oakland, CA, 1995. IEEE TC S&P, IEEE Computer Society Press.

[79] Alan O. Freier, Philip Karlton, and Paul C. Kocher. The SSL protocol version 3.0. Technical report, IETF Transport Layer Security Working Group, November 1996.

[80] Lothar Fritsch. Sichere anonyme Kommunikation für den elektronischen Marktplatz *SEMPER* — Design und Implementierung in Java. Diplomarbeit, Fachbereich Informatik, Universität des Saarlandes, August 1998.

[81] FSTC. Electronic check proposal. Technical report, Financial Services Technology Consortium, 1995.

[82] Eran Gabber, Phillip B. Gibbons, Yossi Matias, and Alain Mayer. How to make personalized web browsing simple, secure, and anonymous. In *Proceedings of the First Conference on Financial Cryptography (FC '97)* [100], pages 17–32.

[83] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns - Elements of Object-Oriented Software*. Addison-Wesley-Longman, Reading, 1995.

[84] German Government. Digital Signature Act (Signaturgesetz — SigG), June 1997. Available from `http://www.iid.de/rahmen`, also in an English translation.

[85] Theodore Goldstein. The gateway security model in the Java electronic commerce framework. In *Proceedings of the First Conference on Financial Cryptography (FC '97)* [100], pages 340–354.

[86] James Gosling, Bill Joy, and Guy Steele. *The Java^{TM}Language Specification*. Sun Microsystems, 1.0 edition, August 1996. Appeared also as book with same title in Addison-Wesleys 'The Java Series'.

[87] Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111, 1991.

[88] Michael A. Harrison, Walter L. Ruzzo, and Jeffry D. Ullman. Protection in operating systems. *Communications of the ACM*, 19(8):461–471, 1976.

[89] Ralf Hauser and Michael Steiner. Generic extensions of WWW browsers. In *First USENIX Workshop on Electronic Commerce* [172], pages 147–154.

[90] Ralf Hauser, Michael Steiner, and Michael Waidner. Micro-payments based on iKP. In *14th Worldwide Congress on Computer and Communications Security Protection* [159], pages 67–82.

[91] Ralf Hauser and Gene Tsudik. On shopping incognito. In *Second USENIX Workshop on Electronic Commerce* [173], pages 251–258.

[92] David Henry, Sandra Cooke, Patricia Buckley, Jess Dumagan, Gurmukh Gill, Dennis Pastore, and Susan LaPorte. The emerging digital economy II. Technical report, U.S. Department of Commerce, Economics and Statistics Administration, Office of Policy Development, Washington, June 1999. Available from `http://www.ecommerce.gov/ede/report.html`.

[93] Amir Herzberg, Yosi Mass, Yoris Mihaeli, Dalit Naor, and Yiftach Ravid. Access control meets public key infrastructure, or: Assigning roles to strangers. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, CA, 2000. IEEE TC S&P, IEEE Computer Society Press.

[94] IACR. *Advances in Cryptology — Crypto '96*, number 1109 in Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 1996.

[95] IACR. *Advances in Cryptology — Crypto '98*, number 1462 in Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 1998.

[96] IACR. *Advances in Cryptology — Eurocrypt '98*, number 1403 in Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 1998.

[97] IACR. *Advances in Cryptology — Crypto '99*, number 1666 in Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 1999.

[98] IEEE TC S&P. *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Oakland, CA, 1996. IEEE Computer Society Press.

[99] IETF Working Group. Public-key infrastructure (X.509) (PKIX). `http://www.ietf.org/html.charters/pkix-charter.html`.

[100] IFCA. *Proceedings of the First Conference on Financial Cryptography (FC '97)*, number 1318 in Lecture Notes in Computer Science, Anguilla, British West Indies, 1997. Springer-Verlag, Berlin Germany.

[101] ISO/IEC. Information technology — Open Systems Interconnection — The directory: Authentication framework. ISO International Standard 9594-8, 1990. Same as ITU-T Rec X.509.

[102] ISO/IEC. Information technology — Open Systems Interconnection — The directory: Authentication framework. ISO International Standard 9594-8:1995, 1995. Same as ITU-T Rec X.509v3.

[103] ISO/IEC JTC 1/SC27, N 1105. Information technology — Security techniques — Non repudiation — Part 1: General model. ISO International Standard 13888-1, 1995.

[104] ISO/IEC JTC 1/SC27, N 1106. Information technology — Security techniques — Non repudiation — Part 1: Using symmetric encipherment algorithms. ISO International Standard 13888-2, 1995.

[105] ISO/IEC JTC 1/SC27, N 1107. Information technology — Security techniques — Non repudiation — Part 1: Using asymetric techniques. ISO International Standard 13888-3, 1996.

[106] ISO/IEC TC154. Electronic data interchange for administration, commerce and transport (EDIFACT) – Application level syntax rules – Part 5: Security rules for batch edi (authenticity, integrity and non-repudiation of origin). ISO Draft International Standard 9735-5:1999, 1999. Syntax version number 4.

[107] Markus Jakobsson. A practical mix. In *Advances in Cryptology — Eurocrypt '98* [96], pages 448–461.

[108] Markus Jakobsson, David M'Raihi, Yiannis Tsiounis, and Moti Yung. Electronic payments: where do we go from here. In R. Baumgart, editor, *Secure Networking - CQRE [Secure] '99*, number 1740 in Lecture Notes in Computer Science, pages 43–63. Springer-Verlag, Berlin Germany, 1999. Invited talk.

[109] Markus Jakobsson, Elizabeth Shriver, Bruce K. Hillyer, and Ari Juels. A practical secure physical random bit generator. In *5th ACM Conference on Computer and Communications Security*, pages 103–111, San Francisco, 1998. ACM Press.

[110] JavaSoft. Java native interface specification, November 1996. Release 1.1.

[111] Anja Jerichow, Jan Müller, Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner. Real-time mixes: A bandwidth-efficient anonymity protocol. *IEEE Journal on Selected Areas in Communications*, 16(4):495–509, 1998.

[112] Brian Kahin. The strategic environment for protecting multimedia. In *Technological Strategies for Protecting Intellectual Property in the Networked Multimedia Environment*, The Journal of the Interactive Multimedia Association Intellectual Property Project, Coalition for Networked Information, pages 1–8, MIT, Program on Digital Open High-Resolution Systems, 1994. Interactive Multimedia Association, John F. Kennedy School of Government.

[113] M. Ethan Katsch. Dispute resolution in cyberspace. In *Connecticut Law Review Symposium: Legal Regulation of the Internet*, number 953 in 28, 1996. Available from `http://www.umass.edu/legal/articles/uconn.html`.

[114] Steven P. Ketchpel, Hector Garcia-Molina, Andreas Paepcke, Scott Hassan, and Steve Cousins. U-PAI: A universal payment application interface. In *Second USENIX Workshop on Electronic Commerce* [173], pages 105–121.

[115] John Knott and Arend Stemerding, editors. *A Guide to Financial EDI.* EBES/EEG4, 1996.

[116] Paul Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In *Advances in Cryptology — Crypto '96* [94], pages 104–113.

[117] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Crypto99 [97], pages 399–397.

[118] KRISIS Consortium. Key recovery in secure information systems. Final report of EU project KRISIS, May 1998. Available from `http://www.cordis.lu/infosec/src/study9.htm`.

[119] Arjen K. Lenstra and Eric V. Verheul. Selecting cryptographic key sizes. 3rd Workshop on Elliptic Curve Cryptography (ECC '99), November 1999. Revised November 1999, `http://www.cryptosavvy.com`.

[120] Peter F. Linington. Fundamentals of the layer service definitions and protocol specifications. *Proceedings of the IEEE*, 71(12):1341–1345, December 1983.

[121] John Linn. Generic security service application program interface, version 2. Internet Request for Comment (RFC) 2078, IETF, 1997.

[122] Mark Lomas, editor. *Security Protocols—International Workshop*, volume 1189 of *Lecture Notes in Computer Science*, Cambridge, United Kingdom, 1997. Springer-Verlag, Berlin Germany.

[123] Sthepen H. Low, Nicholas F. Maxemchuk, and Sanjoy Paul. Anonymous credit cards. In Jacques Stern, editor, *2nd ACM Conference on Computer and Communications Security*, pages 108–117, Fairfax, Virginia, 1994. ACM Press.

[124] MANDATE II Consortium. MANDATE final report. Draft version 2.0, February 1998. Available from `http://www.cryptomathic.dk/mandate/`.

[125] Francisco Fernandes Masaguer. Security in electronic trading over open networks: a detailed analysis and comparison. In *14th Worldwide Congress on Computer and Communications Security Protection* [159], pages 39–66.

[126] MasterCard and Visa. *SET Secure Electronic Transactions Protocol*, Version 1.0 edition, May 1997. Book One: Business Specifications, Book Two: Technical Specification, Book Three: Formal Protocol Definition. Available from `http://www.setco.org/set_specifications.html`.

[127] Gennady Medvinsky and B. Clifford Neuman. NetCash: A design for practical electronic currency on the Internet. In Victoria Ashby, editor, *1st ACM Conference on Computer and Communications Security*, pages 102–106, Fairfax, Virginia, 1993. ACM Press.

[128] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press series on discrete mathematics and its applications. CRC Press, Boca Raton, 1997.

[129] Silvio Micali. Certified e-mail with invisible post offices — or — a low-cost, low-congestion, and low-liability certified e-mail system. Presented at RSA 97, 1997.

[130] Clifford Neuman and Gennady Medvinsky. Requirements for network payment: The NetCheque Prespective. In *Proceedings of IEEE Compcon '95*, San Francisco, 1995.

[131] Peter G. Neumann. *Computer Related Risks*. Addison Wesley – ACM Press, Reading Massachusetts, 1995.

[132] Hans Nilsson and Denis Pinkas. Validation of electronic signatures. White paper, iD2 Technolgies and Bull, January 1999. `http://www.id2.se/whitepapers/ES_validation.pdf`.

[133] Hans Nilsson, Patrick Van Eecke, Manuel Medina, Denis Pinkas, and Nick Pope. Final report of the EESSI expert team. Technical report, European Electronic Signature Standardization Initiative (EESSI), Brussels, July 1999.

[134] OBI Consortium. Open buying on the Internet (OBI) — Technical specifications, release v1.1. Technical report, The OBI Consortium, June 1998. `http://www.openbuy.org`.

[135] Open Market. Internet commerce: The Open Market Transact solution. Technical white paper, Open Market, Inc., July 1998. Available from `www.openmarket.com`.

[136] OTP Consortium. Internet open trading protocol (version: 0.9.9). Technical report, The OTP Consortium, August 1998. `http://www.otp.org`.

[137] Bruce Peat and David Webber. Introducing XML/EDI — The e-business framework. Technical report, The XML/EDI Group, August 1997. `http://www.geocities.com/WallStreet/Floor/5815/guide.htm`.

[138] Torben P. Pedersen. Electronic payments of small amounts. In Lomas [122], pages 59–68.

[139] Andreas Pfitzmann, Birgit Pfitzmann, Matthias Schunter, and Michael Waidner. Trusting mobile user devices and security modules. *IEEE Computer*, 20(2):61–68, 1997.

[140] Andreas Pfitzmann, Birgit Pfitzmann, and Michael Waidner. ISDN-mixes — untraceable communication with very small bandwidth overhead. In *7th IFIP International Conference on Information Security (IFIP/Sec '91)*, pages 245–258. Elsevier, 1991.

[141] Andreas Pfitzmann and Michael Waidner. Networks without user observability. *Computers & Security*, 6(2):158–166, 1987.

[142] Birgit Pfitzmann. *Digital Signature Schemes - General Framework and Fail-Stop Signatures*. Number 1100 in Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1996.

[143] Birgit Pfitzmann, Matthias Schunter, and Michael Waidner. Optimal efficiency of optimistic contract signing. In *17th Symposium on Principles of Distributed Computing (PODC)*, New York, 1998.

[144] Birgit Pfitzmann and Michael Waidner. How to break and repair a "provably secure" untraceable payment system. In *Advances in Cryptology — Crypto '91*, number 576 in Lecture Notes in Computer Science, pages 338–350. IACR, Springer-Verlag, Berlin Germany, 1992.

[145] Birgit Pfitzmann and Michael Waidner. Properties of payment systems — General definition sketch and classification. Research Report RZ 2823 (#90126), IBM Research Division, May 1996. Submitted for Publication.

[146] Birgit Pfitzmann, Michael Waidner, and Andreas Pfitzmann. Rechtssicherheit trotz Anonymität in offenen digitalen Systemen. *Computer und Recht*, 3(10,11,12):712–717, 796–803, 898–904, 1987. Extended revision in Datenschutz und Datensicherung DuD 14/5–6 (1990) 243–253, 305–315. (English translation available from authors.).

[147] David Pointcheval. Strengthened security for blind signatures. In *Advances in Cryptology — Eurocrypt '98* [96], pages 391–405.

[148] Michael O. Rabin. Transaction protection by beacons. *Journal of Computer and System Sciences*, 27:256–267, 1983.

[149] Michael G. Reed, Paul F. Syverson, and David M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, 1998.

[150] Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security*, 1(1):66–92, 1998.

[151] Paul Resnick and James Miller. PICS: Internet access controls without censorship. *Communications of the ACM*, 39(10):87–93, 1996.

[152] Ron L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Journal of the ACM*, 21(2):120–126, February 1978. US Patent 4,405,829: Cryptographic Communications System and Method, Public Key Partners PKP.

[153] Ronald L. Rivest and Adi Shamir. PayWord and MicroMint: Two simple micropayment schemes. In Lomas [122], pages 69–88.

[154] RosettaNet. RosettaNet - an overview, 1998. `http://www.rosettanet.org/general/overview.html`.

[155] Detlef Schoder, Ralf E. Strauss, and Peter Welchering. *Electronic Commerce Enquête 1997/98, Survey on the business uses of electronic commerce for companies in the German speaking area.* Konradin Verlag, Stuttgart, 1998. Executive Research Report.

[156] B. Schoenmakers. Basic security of the ecash™payment system. In Bart Preneel and Vincent Rijmen, editors, *State of the art in applied cryptography*, number 1528 in Lecture Notes in Computer Science, pages 338–352. Springer-Verlag, Berlin Germany, 1998.

[157] Berry Schoenmakers. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In Crypto99 [97], pages 148–164.

[158] Matthias Schunter. *Optimistic Fair Exchange.* PhD thesis, Universität des Saarlandes, Saarbrücken, 2000.

[159] SECURICOM. *14th Worldwide Congress on Computer and Communications Security Protection*, C.N.I.T Paris-La Defense, France, 1996.

[160] SEMPER Consortium. Basic services: Architecture and design. Deliverable D03 of ACTS project AC026, public specification, September 1996. Available from `http://www.semper.org`.

[161] SEMPER Consortium. Architecture, services and protocols. Deliverable D10 of ACTS project AC026, public specification, January 1999. Available from `http://www.semper.org`.

[162] SEMPER Consortium. Evaluation of the enhanced trial. Deliverable D12 of ACTS project AC026, April 1999. Available from `http://www.semper.org`.

[163] State of Utah. Utah digital signature act. Title 46 — Chapter 03 of the Utah Code, 1996. Available from `http://www.commerce.state.ut.us/web/commerce/digsig/act.htm`.

[164] Christian Stüble. Development of a prototype for a security platform for mobile devices. Diploma thesis, Fachbereich Informatik, University of Dortmund, May 2000.

[165] Sun Microsystems. *The Java Wallet^{TM}Architecture White Paper*, 1998. Available from `http://java.sun.com/products/commerce/docs/`.

[166] Efraim Turban. Auctions and bidding on the Internet: An assessment. *EM — Electronic Markets*, 7(4):7–11, 1997. `http://www.electronicmarkets.org`.

[167] U. S. National Institute of Standards and Technology NIST. The digital signature standard. Federal Information Processing Standards Publication 186 (FIPS PUB 186), May 1994.

[168] U. S. National Institute of Standards and Technology NIST. Security requirements for cryptographic modules, January 1994. Federal Information Processing Standards Publication 140 (FIPS PUB 140).

[169] UBS. Registration for SET (Secure Electronic Transaction) — Application for the supplementary function SET for the UBS Visa Card. Document SETRUBSE - 3.9.98, UBS AG, VISA Center, Flughofstrasse 35, CH-8152 Glattbrugg, Switzerland, 1998.

[170] United States. United States' uniform commercial code article 2 — Sales, part 2: Form, formation and readjustment of contracts, § 2-201. Formal requirements; statute of frauds.

[171] University of Leuven. Study for DG XV, European Commission with respect to the legal aspects of digital signatures, 1998. Interdisciplinary centre for Law and Information Technology.

[172] USENIX. *First USENIX Workshop on Electronic Commerce*, New York, 1995.

[173] USENIX. *Second USENIX Workshop on Electronic Commerce*, Oakland, California, 1996.

[174] USENIX. *Third USENIX Workshop on Electronic Commerce*, Boston, Mass., 1998.

[175] The Wassenaar arrangement on export controls for conventional arms and dual-use goods and technologies, 1996.

[176] Arnd Weber. See what you sign. secure implementation of digital signatures. In *Intelligence in Services and Networks: Technology for Ubiquitous Telecom Services (IS&N'98)*, number 1430 in Lecture Notes in Computer Science, pages 509–520, Berlin, 1998. Springer-Verlag.

[177] Arnd Weber. Full bindingness and confidentiality. Requirements for secure computers, and design options. In *8th European Conference on Information Systems ECIS 2000: A Cyberspace Odyssey.* Vienna University of Economics and Business Administration, IEEE, 2000.

[178] Ursula Widmer and Konrad Bähler. *Rechtsfragen im Internet - Sichere Geschäftstransaktionen im Internet.* Orell Füssli Verlag, Zürich, 1997.

[179] WIPO. Berne convention for the protection of literary and artistic works (1886); WIPO copyright treaty (WCT) (1996); WIPO performances and phonograms treaty (WPPT) (1996), 1996. `http://www.wipo.org/eng/general/index5.htm`.

[180] Robert Hobbes Zakon. Hobbes' Internet timeline v3.3. Internet Request for Comment (RFC) 2235, IETF, 1998. See `http://www.isoc.org/guest/zakon/Internet/History/HIT.html`.

[181] Zentraler Kreditausschuss (ZKA). HBCI — Homebanking computer interface, March 1999. Version 2.1.

[182] Jianying Zhou and Dieter Gollmann. A fair non-repudiation protocol. In *Proceedings of the IEEE Symposium on Research in Security and Privacy* [98], pages 55–61.

[183] Jianying Zhou and Dieter Gollmann. An efficient non-repudiation protocol. In *10th Computer Security Foundations Workshop*, pages 126–132. IEEE Computer Society Press, Los Alamitos, 1997.

Glossary

Acquirer The bank used by the recipient of electronic money.

Audit trail Logging information which is the basis for an audit of the use of a service or system. The audit trail may contain non-repudiation tokens to resolve disputes. *SEMPER* also provides higher-level transaction records and a transaction browser.

Authentication A proof of the origin (and integrity) of a message or the identity of a person; in the second case typically combined with establishing a secure channel to that person. Also said for producing or verifying such a proof. Authentication does not imply non-repudiation. See also *Digital signature* and *Message authentication code*.

Business application Business applications implement specific business processes. In the *SEMPER* context they are built on top of the *SEMPER* services. As there are no restrictions on the implementation of Business Applications by third parties, they are a priori untrusted and not allowed to perform security-critical actions without user authorization.

Business-Item Layer The Business-Item Layer handles the business items of various nature, such as payments, statements, or information. In particular, it deals with the simple transfer of such items and provides, where necessary, management functionality.

Business process A business process consists of several linked step, e.g., a purchase process contains an offer, and order, delivery and payment. Many security properties must be ensured for the process as a whole. In *SEMPER*, business processes are represented by *deals*.

Certificate A certificate is a digitally signed statement about a person or key. *SEMPER* distinguishes three types of certificates: Key certificates link a public key and a person, while attribute certificates link a person or public key and an attribute, typically a property or right. Hybrid certificates combine this, i.e., they link a person, a public key, and an attribute.

Certification authority (CA) A third party that signs digital certificates.

Certified mail A fair exchange of a message for a receipt.

Channel A medium shared by peer entities for communication.

Commerce Layer The Commerce Layer offers security services to business applications in *SEMPER*, in particular those that offer process orientation. Its main parts are the deal support services and the commerce transaction service. It also maintains the user's security policy.

Commerce transaction Commerce transactions represent service primitives in the Commerce Layer. A commerce transaction encapsulates a protocol to be executed between the participants of a deal. It is typically a value-added version of a lower-layer transaction, e.g., the simple delivery of a message or a complex payment protocol. Commerce transactions must always exist in the context of a deal.

Conditional access Access to a service, which is restricted to entities having certain properties or rights.

Confidentiality Protecting a message against eavesdropping such that it is only meaningful to the intended set of recipients. More generally, any protection of the secrecy of an action.

Container A data structure used in *SEMPER* to transfer or exchange information and payments. It is structured in the form of a tree. The leaves specify or containing information, statements, or payments. Internal nodes contain security attributes applying to subtrees.

Context Intuitively, a context means all the prior information relevant for a business transaction. Technically, in *SEMPER* a context is the state of a session, in particular (on the Commerce Layer) of a deal. It evolves from the preferences set by the user, the results of negotiating parameters with business partners, and prior transactions in the same session.

Contract signing In general, a non-repudiable proof of agreement on a certain text. A fair exchange of two signatures under a contract is a special case.

Credential A statement about properties or rights of a person, or of the issuers relation (e.g., trust) to this person. A physical credential might be a passport present at time of registration with a RA. For the electronic world, we use it as a synonym of attribute certificate.

Customer The role of a user with respect to a service provider, typically a buyer with respect to a merchant.

Deal The Commerce Layer introduces deals as a representation of business processes, i.e., several transactions operating on the same context. A deal contains the representation of an association between the participants, the history of commerce transactions between the participants and private data stored by the participants.

Device A physical object, such as a personal computer. Usually, a device belongs to the player (or set of players) who relies on it. We also say user equipment.

Digital signature A digital signature is an electronic analog of handwritten signatures. It is verified against a public key, and if the signature system is secure, this serves as a proof that the signed message originates from someone knowing the corresponding secret key. This should be only one person; whether it is, depends on the security of user equipment and certification.

Directory authority (DA) A third party that maintains a register of public information about users including their certificates.

Dispute handling See *Exception handling*.

Electronic money Electronic money is the electronic analog of conventional money. We typically speak of electronic payments, because the notion of "money" has additional connotations in economics.

Entity This word has no specific definition in *SEMPER*. We often use it as a vague version of "person", comprising legal entities and persons acting under pseudonyms. We also use it to designate an arbitrary active element in the *SEMPER* architecture (e.g., "the peer entity"); thus an entity could be a manager, a module, a transaction etc.

Exception handling Exception handling means resolving an exception, i.e., an undesired state noticed by an entity. In a first approach the exception handling bases on the assumption that all parties are honest. If this fails, the parties are in a dispute and a more pessimistic approach must be used, usually involving arbiters and finally courts. Exception handling requires all parties to keep sufficient audit trails. For real disputes, the audit trail has to contain evidence, e.g., non-repudiation tokens.

Exchange An exchange is a protocol whereby a number of parties can exchange business items (payment, information, vouchers, etc.). In the simplest case these are two transfers. See also *Fair exchange*.

External interface The interface of a block of the *SEMPER* architecture to callers from outside the block. This is also often called an Application Programming Interface (API).

Fair exchange A fair exchange is an exchange where it is guaranteed that either all parties obtain the desired items or none. For this, they input their expectations at the start of the transactions. Examples are *Certified mail*, *Fair purchase*, and *Contract signing*.

Fair Internet Trader The Fair Internet Trader (FIT) is an electronic assistant for interactive person-to-person trade. Such trade ranges from low-value transactions like occasional sales of used items between private persons over medium-value transactions between small businesses offering specialized services, to high-value transactions with strong formal requirements like public procurement. Technically, the FIT is a business application in *SEMPER*.

Fair purchase A fair exchange of information for a payment.

Generic Payment Service Framework A *SEMPER* framework enabling business applications to use a variety of payment systems in a transparent manner. In *SEMPER*, it is also called the Payment Block.

Hash function A function (or family of functions) used to compute a fixed-length digest of any bit string. Unless stated otherwise, a hash function is assumed to be one-way and collision-resistant, meaning that it is infeasible in practice to find the input string given the digest or to find two different bit string with the same digest.

Identification A process whereby an entity proves its identity.

Integrity protection The protection of information against unauthorized modification.

Internal interface The interface that a module (e.g., a specific payment instrument) must offer to fit into *SEMPER*. This is also often called a Service Provider Interface (SPI).

Issuer In the context of an electronic payment systems it means the bank of the payer. In the context of certificates it means the certification authority.

Layer A (virtual) collection of services, and of entities in all devices offering these services, that have a similar degree of abstraction.

Liability-cover service A *SEMPER* service that allows to limit the risk of both key owners and relying parties from compromise of signature keys, e.g., due to *Trojan horses*.

Manager A manager is a fixed entity in a *SEMPER* block, which provides functions for negotiation and selection of an appropriate module.

Message authentication code (MAC) A MAC assures the recipient of a message of its origin. However, this is symmetric cryptography, i.e., the sender and the recipient share the key of the MAC function. Hence a MAC does not offer non-repudiation, i.e., it cannot be used to convince a third party of the origin of the message. See also *Digital signature*.

Module In *SEMPER*, each type of services may be provided by different modules (e.g., different payment systems of crypto libraries). It is expected that modules will come from different providers. A module must support the internal interface of a *SEMPER* block in order to be used by the manager.

Multi-party security Security without a-priori assumptions that everybody will trust particular entities.

Non-repudiation Non-repudiation of an event means that one can convince a third party that the event took place. In particular, non-repudiation of origin means that one can prove that a message originated from a certain entity, and non-repudiation of delivery means that one can prove that a message has been delivered.

Non-repudiation token A non-repudiation token is the information needed for non-repudiation, i.e., the proof needed to convince a third party. These tokens are often necessary for dispute handling.

Optimistic protocol A protocol where correctness is guaranteed by a third party which is only involved in case of faults.

Payment instrument An instance of one player's component of a payment system.

Payment manager Overall controller of an instance of the Generic Payment Service.

Payment system A collective name for one "way" of making a payment. It consists of protocols, contractual agreements, and data structures.

Payment system provider A party (such as a financial institution) that makes a payment system available.

Peer entity Peer entities are two entities of the same block but located on different devices. Usually, we only speak of peer entities if the entities work together to provide a certain service, e.g., two peer managers or two peer entities in a transaction.

Player The word "player" has no specific meaning in *SEMPER*. It is typically used for is a real-world person or body participating in the electronic marketplace.

Preferences Settings entered by the user that personalize a program according to his wishes. In *SEMPER*, in particular the user's default choices of security attributes.

Protocol A description of how a service is provided by means of interactions of peer entities, often using lower layer services.

Public-key infrastructure (PKI) All entities involved in handling public-key certificates. See *Certification authority, Directory authority* and *Registration authority.*

Purse Representation of a payment instrument within the Generic Payment Service Framework.

Purse-management application User application to create and manage purses in the Generic Payment Service.

Registration Here it usually means the registration of the public key of a signature system. Optimally, an entity (usually a real person) appears in person at a *Registration authority*, proves his or her identity, and signs that he or she wants to accept a certain liability for this key. In return, the participant gets a certificate from an associated *Certification authority.*

Registration authority (RA) An entity that carries out registration.

Revocation The process of withdrawing something; typically a public key or a certificate. The reason can be that the key is compromised or lost or was registered by an impostor, or that the key remains valid but some information in the certificate changes (e.g., if it contains an address). An attribute certificate may be revoked if the rights change.

Role The function of an entity in a certain transaction. All protocols are specified for roles (e.g., "a buyer and a merchant"), but performed by specific entities "playing" the corresponding roles. In a related sense, the word "role" is used in access control.

SECA certificate A certificate stating that an entity is using a key and accepting liability for it according the specific provisions of the *SEMPER Electronic Commerce Agreement.*

Secure channel A channel between two entities fulfilling certain security requirement, e.g., authenticity, integrity and/or confidentiality. Note that non-repudiation can typicallly not be fulfilled by a secure channels.

Security attribute Security attributes are a high-level way to select the security level of a certain service. Examples are confidentiality, non-repudiation and anonymity. In *SEMPER*, security attributes of a transaction are derived from preferences, the surrounding session if there is one, inputs from the caller if it is trusted, and in rare cases from direct user input.

SEMPER Electronic Commerce Agreement (SECA) The SECA is a model agreement that can be used to regulate (among its subscribers) liability questions and other legal issues which are not or only unsatisfactorily covered by current regulation.

Service Service generally means the joint functionality of a system or component as offered to its environment. layer to its upper interface. For instance, we speak of the services offered by a *SEMPER* block, or the entire *SEMPER* services.

Session A session provides a common context to several transactions. Compared with a transaction, a session is longer-lived and atomicity is not required. Sessions can be nested: a session on a higher layer may start several sessions on lower layers (or use existing sessions) and exchange parts of its state with them, e.g., security attributes).

SME Small and medium size enterprises.

Third party A third party is an entity supporting a business transaction without being directly involved as a business partner. Typical third-party services are notary services.

TINGUIN (Trustworthy Interactive Graphical User Interface) Security requires a carefully designed user interface. Important aspects are a uniform look-and-feel for critical steps, and clear indications of “points of no return”. In *SEMPER*, this is called TINGUIN. If one had a *Trusted computing base*, a trusted path to the TINGUIN would be implemented, so that the user could not be tricked into taking other windows for the TINGUIN.

Transaction Intuitively, a transaction in *SEMPER* is one step in a business process. More technically, it should be a transaction in the usual computer-science sense, i.e., atomic. This means that it either finishes successfully or has no effect at all. It should also mean a transition from one consistent state into another, at least for the block providing the transaction. (For instance, a payment transaction keeps the purse states and the states of the bank consistent, but on the higher layers a delivery may have to precede or follow.)

Transaction browser User application to examine transaction records, e.g., from payment transactions.

Transfer The process of transferring business items from a sender to a recipient, usually in a secure way. Transferable business items are for example payments, signed statements, and information.

Transfer-and-Exchange Layer The Transfer-and-Exchange Layer coordinates the secure exchanges and transfers of business items and containers. In particular it handles fairness. See also *Business-Item Layer, Container* and *Fair exchange.*

Trojan horse A program that contains malicious parts apart from its usual functionality. Viruses are the best-known case. Together with bugs in well-known programs like browsers and emailers that allow outsiders to get code executed on a machine whose user is not intending to load any code, Trojan horses are the main reason why one cannot place unlimited trust in digital signatures at present. See *Liability-cover service, Trusted computing base.*

Trust management Trust management enables the user to specify policies for using his own certificates and accepting certificates from other parties in business processes, and provides services for selecting the appropriate certificates that satisfy all requirements for a business application.

Trusted computing base (TCB) The part of the system which enforces the security policies. The TCB should be nontamperable and noncircumventable, i.e., it has to protect itself and its users from malicious applications, in particular *Trojan horses.* As the TCB has to be trusted by definition it’s important that the design and the implementation of the TCB is also trustworthy, i.e., it is minimal and verifiable.

User A user is anyone using the marketplace (e.g., buyer or seller). We also say “user” for the owner of a device, e.g., in *User authorization.*

User authorization The approval of an action (of a program) by the user, typically a human user in the case of consumers and small businesses. Mandatory for security-critical operations such as signing a contract or payments.

Index