To appear in Proc. of 1994 European Symposium on Research in Computer Security (ESORICS' 94), Brighton, England, Nov. 7-9, 1994.

A Calculus for Secure Channel Establishment in Open Networks

Ueli M. Maurer
Inst. for Theoretical Computer Science
ETH Zürich
CH-8092 Zürich, Switzerland

Pierre E. Schmid
Omnisec AG
Trockenloostrasse 91
CH-8105 Regensdorf, Switzerland

Abstract. This paper presents a calculus of channel security properties which allows to analyze and compare protocols for establishing secure channels in an insecure open network at a high level of abstraction. A channel is characterized by its direction, time of availability and its security properties. Cryptographic primitives and trust relations are interpreted as transformations for channel security properties, and cryptographic protocols can be viewed as combinations of such transformations. A protocol thus allows to transform a set of secure channels established during an initial setup phase, together with a set of insecure channels available during operation of the system, into the set of secure channels specified by the security requirements. The necessary and sufficient requirements for establishing a secure channel between two entities are characterized in terms of secure channels to be made available during the initial setup phase and in terms of trust relations between users and/or between users and trusted authorities.

Keywords. Network security, Key management, Cryptography, Security transformations, Formal models.

1. Introduction

The importance of security in large distributed systems has long been identified and addressed by academic and industrial research (e.g., see [1, 2, 4, 6, 7, 16, 17]), and several solutions and products have been proposed [13, 9, 8, 15, 18]. In the coming years, these concepts will most likely be introduced and used on a large scale in government, commercial and academic networks. While the cryptographic technology (both private-key and public-key mechanisms) is available, the key management and in particular the trust management presents non-trivial problems. It remains to be evaluated for which application scenarios approaches based on on-line servers [13, 9], centralized and hierarchical certification authorities [15, 19] or decentralized key certification [18] are best suited. These approaches vary dramatically with respect to the required communication, user responsibility and trust relations, and it is one of the goals of this paper to compare them in a formal framework.

The process of establishing security in a distributed system can be viewed as a two-phase process: During an *initialization phase*, communication channels with security properties (e.g. trusted couriers, personal registration at a trusted center, mutual authentication by speaker identification on a voice channel, etc.) are available for setting up security parameters like shared secret encryption keys and authenticated or certified public keys. During the later *communication phase*, entities (users or applications) can typically communicate only over insecure channels. The purpose of applying cryptographic techniques can be viewed as the tranfer of the security properties of initially available channels to the insecure channels available in the communication phase, thus making the latter secure.

The purpose of this paper is to provide a straight-forward formalism for illustrating and comparing the various approaches to security in distributed open systems. The emphasis of our approach is on the simplicity and expressive power of the model. Unlike many previous formal treatments of security and authentication in distributed systems [2, 5, 14, 17], it is not intended to be applied for the design or the security verification of protocols. On the other hand, it allows to illustrate in a simple manner the minimal requirements for achieving security between two users in a distributed system, the timing constraints on the involved communication channels, the complete duality between authenticity and confidentiality and the distinguishing features between secret-key and public-key cryptography.

While cryptography is sometimes believed to solve all security problems in open systems, our model allows to demonstrate in a simple manner that cryptography cannot "create" security. The design of a cryptographic protocol can rather be seen as the problem of finding an initialization scenario that is practical and realistic in terms of the initially required secure channels and in terms of inherent assumptions such as a person's or autority's trustworthyness, and from which the desired security goal for the communication phase can be derived by cryptographic transformations. The minimal requirements for achieving security between two users in a distributed system are characterized in terms of secure channels to be made available in an initial setup phase and in terms of necessary trust relations between users and/or between users and trusted authorities. Several types of protocols are reviewed within the presented framework, but it is not a goal of this paper to develop new protocols. We do not distinguish in this paper between different types and degrees of trust [17], but our model could be extended in this directions.

The paper is organized as follows. Section 2 describes a classification of channel security properties and Section 3 provides a complete list of cryptographic transformations of such channel security properties. Sections 6 and 4 discuss the necessary and sufficient condition for establishing a secure channel between two users in an open network, with and without exploiting trust relations, respectively, and security transformations based on trust relations are introduced in Section 5. In Section 7, several approaches to bootstrapping security in an open network are discussed and compared.

2. Classification of channel security properties

A communication channel can be viewed as a means for transporting a message from a source (the channel input) to a destination (the channel output). The duality of source and destination of a message or, equivalently, the duality of input and output of a channel, is reflected in the duality of the two fundamental security goals for messages (or channels). A channel provides *confidentiality* if its output is exclusively accessible to a specified receiver and this fact is known to the potential senders on the channel. Similarly, a channel provides *authenticity* if its input is exclusively accessible to a specified sender and this fact is known to the receivers.

Confidentiality and authenticity are independent and dual security properties. One can be available without the other. Hence channels can be classified into four different types according to whether they provide none, either or both of these security properties.

Channels are denoted by the symbol $\longrightarrow$ and allow to transmit, at a given time, a message of unspecified length from an input to an output. The symbol $\bullet$ attached to the channel symbol $\longrightarrow$ will indicate that the user at the corresponding end of the channel has exclusive access to the channel. The symbols for the four types of channels from an entity A to an entity B, as well as for a bidirectional secure channel, are:

$A \longrightarrow B$	channel that provides no security.
$A \longrightarrow\!\!\bullet\, B$	provides confidentiality but not authenticity.
$A\, \bullet\!\!\longrightarrow B$	provides authenticity but not confidentiality.
$A\, \bullet\!\!\longrightarrow\!\!\bullet\, B$	provides both confidentiality and authenticity.
$A\, \bullet\!\!\longleftrightarrow\!\!\bullet\, B$	bidirectional secure channel between A and B.

An illustrative real-world example of a $\longrightarrow\!\!\bullet$ channel is a mailbox for which only a designated person possesses a key. Someone putting a letter into the mailbox is assured of the message's confidentiality, but the recipient has no direct means for authenticating the sender. A more realistic example will be discussed in Section 8. Examples of $\bullet\!\!\longrightarrow$ channels are a bulletin board that is physically protected by a glass cover and a lock (with the key available only to a designated sender), and an insecure telephone line combined with reliable speaker identification. Examples of a $\bullet\!\!\longrightarrow\!\!\bullet$ channel are a trusted courier, an optical fiber (under certain assumptions) or an insecure channel protected by encryption.

Extending our classification of channels, a parameter above the channel symbol $\longrightarrow$ will indicate the time when such a channel is available. The symbols $\xrightarrow{t}$, $\bullet\!\!\xrightarrow{t}$, $\xrightarrow{t}\!\!\bullet$ and $\bullet\!\!\xrightarrow{t}\!\!\bullet$ will denote channels that are available at time t. For example, the availability of a trusted courier from A to B at time t is denoted as $A\, \bullet\!\!\xrightarrow{t}\!\!\bullet\, B$. If this channel is used to send a secret key which can thereafter be used to encrypt and authenticate messages exchanged between A and B, an insecure channel $A \xrightarrow{t'} B$ from A to B available at some later time t' can thus be converted into a secure channel from A to B available at time t', denoted $A \bullet\!\!\xrightarrow{t'}\!\!\bullet B$.

In many derivations in the paper we will also need to consider channels that allow to send a message at a certain time t_2, but only a message that has been fixed at an earlier time $t_1 < t_2$. For the various types of security properties, such channels will be denoted by $A \xrightarrow{t_2[t_1]} B$, $A \bullet\!\!\xrightarrow{t_2[t_1]} B$, $A \xrightarrow{t_2[t_1]}\!\!\bullet B$ and $A \bullet\!\!\xrightarrow{t_2[t_1]}\!\!\bullet B$, where the notation $t_2[t_1]$ implies that $t_2 > t_1$ and where the bracketed time can be omitted when $t_1 = t_2$. For example, a $A \bullet\!\!\xrightarrow{t_2[t_1]} B$ could result when A gets a certificate from a trusted authority on her public key at time t_1, which she sends to B at time t_2. Assuming that B can validate the certificate, the channel is authenticated, but note that the message (A's public key) had to be known and fixed at time t_1.

We have the following trivial channel transformations: If a $A \xrightarrow{t_2[t_1]} B$ channel is available then so is a $A \xrightarrow{t_2[t_1']} B$ channel for all $t_1' \leq t_1$; this is also true for the other types of channels. Hence we have for instance

$$\left.\begin{array}{c} A \xrightarrow{t_2[t_1]} B \\ t_1' \leq t_1 \end{array}\right\} \quad\Longrightarrow\quad A \xrightarrow{t_2[t_1']} B$$

Furthermore, the symbol $\bullet$ can trivially be dropped when it is not needed in a transformation. For instance,

$$A \bullet\!\!\xrightarrow{t}\!\!\bullet B \quad\Longrightarrow\quad A \bullet\!\!\xrightarrow{t} B \tag{1}$$

If channels are available from A to B and from B to C at some times t_2 and t_4, respectively (where possibly the messages must be fixed at earlier times t_1 and t_3, respectively), then B can relay a message from A to C provided that $t_3 > t_2$. Formally we write:

$$\left.\begin{array}{c} A \xrightarrow{t_2[t_1]} B \\ B \xrightarrow{t_4[t_3]} C \\ t_3 > t_2 \end{array}\right\} \quad\Longrightarrow\quad A \xrightarrow{t_4[t_1]} C \tag{2}$$

Note that the message on the resulting channel from A to C must be fixed by A at time t_1 while it is received by C only at time t_4.

Of course, for the $A \xrightarrow{t_4[t_1]} C$ channel to be reliable, B must be reliable. However, unlike trust, reliability is not explicitely represented in our model because our goal is to achieve security in an insecure but reasonably reliable open network. If the channels from A to B and from B to C both either provided confidentiality or authenticity or both, then so would the channel from A to C, but only if B can be trusted by A and C. Such transformations based on trust relations will be discussed in Section 5.

A typical security goal for an open network is that every pair (U_i, U_j) of users can communicate securely at any time. In our formalism, a $U_i \bullet\!\!\xleftrightarrow{t}\!\!\bullet U_j$ channel is required for all $i \neq j$ and for all $t > t_0$ where t_0 is some sufficiently early system setup time. In an open system where insecure channels can be assumed to be available at all times, one way for achieving this goal is for two users to agree on a bilateral secret key for use in a symmetric cryptosystem.

3. Basic cryptographic security transformations

The purpose of this section is to present a systematic discussion of the well-known cryptographic primitives (symmetric encryption, public-key cryptosystem and digital signature schemes) by interpreting them as transformations of channel security properties.

3.1. Symmetric encryption and message authentication codes

It is often assumed that a symmetric cryptosystem provides implicit message authentication: the fact that a message is encrypted with a certain key "proves" that the sender knows the key. However, it should be pointed out that this can only be true under the assumption that plaintext is sufficiently redundant and hence that meaningless messages can be distinguished from valid messages. Moreover, certain types of ciphers (e.g. additive stream ciphers) provide no implicit message authentication because single bits in the ciphertext, and hence in the plaintext, can be flipped selectively. This problem can be solved by appending to a given message a cryptographic hash value of the message. In the sequel we therefore assume without loss of generality that a symmetric cipher provides both confidentiality and authenticity.

The basic security transformation provided by a symmetric cipher is to transfer the security of a channel available at some time t_2 to an insecure channel available at some later time t_4. The times t_1 and t_3 are included for the sake of generality and will be used later, but the reader can here and in the sequel just as well assume that $t_2 = t_1$ and $t_4 = t_3$.

$$\left.\begin{array}{c} A \bullet\!\!\xrightarrow{t_2[t_1]}\!\!\bullet B \\ A \xrightarrow{t_4[t_3]} B \\ t_4 \geq t_2 \end{array}\right\} \quad\Longrightarrow\quad A \bullet\!\!\xrightarrow{t_4[t_3]}\!\!\bullet B \tag{3}$$

If the insecure channel is from B to A rather than from A to B, then so is the resulting secure channel:

$$\left.\begin{array}{c} A \bullet\!\!\xrightarrow{t_2[t_1]}\!\!\bullet B \\ A \xleftarrow{t_4[t_3]} B \\ t_3 > t_2 \end{array}\right\} \quad\Longrightarrow\quad A \bullet\!\!\xleftarrow{t_4[t_3]}\!\!\bullet B \tag{4}$$

It is interesting to notice that a symmetric cryptosystem can also be used to transfer confidentiality without authenticity:

$$\left.\begin{array}{c} A \xrightarrow{t_2[t_1]}\!\!\bullet\; B \\ A \xrightarrow{t_4[t_3]} B \\ t_4 \geq t_2 \end{array}\right\} \quad \Longrightarrow \quad A \xrightarrow{t_4[t_3]}\!\!\bullet\; B \tag{5}$$

A can use the confidential $A \xrightarrow{t_2[t_1]}\!\!\bullet\; B$ channel for transferring a (not authenticated) cipher key. Messages encrypted with this key can only be decrypted by B who can check that the message was sent by the same person who previously sent the secret key. However, the $A \xrightarrow{t_2[t_1]}\!\!\bullet\; B$ channel provides no authenticity and hence nor does the $A \xrightarrow{t_4[t_3]}\!\!\bullet\; B$ channel. On the other hand, if the second channel provides authenticity, then so does the resulting channel:

$$\left.\begin{array}{c} A \xrightarrow{t_2[t_1]}\!\!\bullet\; B \\ A \;\bullet\!\!\xrightarrow{t_4[t_3]} B \\ t_4 \geq t_2 \end{array}\right\} \quad \Longrightarrow \quad A \;\bullet\!\!\xrightarrow{t_4[t_3]}\!\!\bullet\; B \tag{6}$$

While a symmetric cryptosystem allows to transfer confidentiality without authenticity (transformation (5)), it is important to note that it does not allow to transfer authenticity without confidentiality. The latter is achieved only by digital signatures and can be seen as a (the) major achievement of public-key cryptography. On the other hand, a symmetric cryptosystem can be used to convert a confidential channel into an authenticated channel. If A sends a secret key to B over the confidential channel, then A can later recognize messages encrypted with this key as authentic from B. However, since B cannot verify that A is indeed the sender of the secret key, the confidentiality of encrypted messages is not guaranteed. If B receives several (not authenticated) secret keys he can authenticate a message for each key separately.

$$\left.\begin{array}{c} A \xrightarrow{t_2[t_1]}\!\!\bullet\; B \\ A \xleftarrow{t_4[t_3]} B \\ t_3 > t_2 \end{array}\right\} \quad \Longrightarrow \quad A \xleftarrow{t_4[t_3]}\!\!\bullet\; B \tag{7}$$

If the second channel provides confidentiality, then so does the resulting channel:

$$\left.\begin{array}{c} A \xrightarrow{t_2[t_1]}\!\!\bullet\; B \\ A \;\bullet\!\!\xleftarrow{t_4[t_3]} B \\ t_3 > t_2 \end{array}\right\} \quad \Longrightarrow \quad A \;\bullet\!\!\xleftarrow{t_4[t_3]}\!\!\bullet\; B \tag{8}$$

Note that the timing constraint in (4), (7) and (8) is different from that in (3), (5) and (6) because B can send the reply only after receiving the message from A. Transformations (7) and (8) can also be achieved by using a message authentication code (MAC) which provides explicit symmetric authentication of messages that need not be confidential.

3.2. Public-key cryptosystems

The basic transformation provided by an (asymmetric) public-key cryptosystem is to transform the authenticity of a channel into confidentiality of a channel available at some later time:

$$\left.\begin{array}{c} A \bullet\!\!\xrightarrow{t_2[t_1]} B \\ A \xleftarrow{t_4[t_3]} B \\ t_3 > t_2 \end{array}\right\} \quad \Longrightarrow \quad A \bullet\!\!\xleftarrow{t_4[t_3]} B \tag{9}$$

If the second channel provides authenticity then so does the resulting channel:

$$\left.\begin{array}{c} A \bullet\!\!\xrightarrow{t_2[t_1]} B \\ A \xleftarrow{t_4[t_3]}\!\!\bullet B \\ t_3 > t_2 \end{array}\right\} \quad \Longrightarrow \quad A \bullet\!\!\xleftarrow{t_4[t_3]}\!\!\bullet B \tag{10}$$

It should be pointed out that a public-key distribution system as defined by Diffie and Hellman [3], if combined with a symmetric cryptosystem, is equivalent to a public-key cryptosystem in the sense that it provides exactly the same transformations (9) and (10).

A comparison of transformations (7) and (9) suggests that a public-key cryptosystem is in some sense the dual of a symmetric message authentication code (MAC).

3.3. Digital signature schemes

The set of transformations considered so far is not complete. The missing one, namely to transfer the authenticity of a channel to an insecure channel available at some later time, is provided by a digital signature scheme:

$$\left.\begin{array}{c} A \bullet\!\!\xrightarrow{t_2[t_1]} B \\ A \xrightarrow{t_4[t_3]} B \\ t_4 > t_2 \end{array}\right\} \quad \Longrightarrow \quad A \bullet\!\!\xrightarrow{t_4[t_3]} B \tag{11}$$

If the second channel is confidential, then so is the resulting channel:

$$\left.\begin{array}{c} A \bullet\!\!\xrightarrow{t_2[t_1]} B \\ A \xrightarrow{t_4[t_3]}\!\!\bullet B \\ t_4 > t_2 \end{array}\right\} \quad \Longrightarrow \quad A \bullet\!\!\xrightarrow{t_4[t_3]}\!\!\bullet B \tag{12}$$

Of course the $A \bullet\!\!\xrightarrow{t_2[t_1]} B$ channel is not "consumed" by the transformation. Thus for instance if $t_2 > t_4$ one could use the $A \bullet\!\!\xrightarrow{t_2[t_1]} B$ channel directly without applying digital signatures. Notice the different timing constraints when compared to transformations (9) and (10). A comparison of transformations (5) and (11) demonstrates that a digital signature scheme is in some sense the dual of a symmetric cryptosystem.

4. The necessary and sufficient condition for a secure channel $A \bullet\!\!\longleftrightarrow\!\!\bullet B$ between two users

The transformations discussed in the previous section can be interpreted as methods for moving or replacing channel symbols ($\longrightarrow$) while keeping the security symbols $\bullet$ in place and attached to the corresponding users. For example, transformation (3) can be interpreted as replacing the channel $\overset{t_2[t_1]}{\longrightarrow}$ in $A \bullet\!\!\overset{t_2[t_1]}{\longrightarrow}\!\!\bullet B$ with the channel $\overset{t_4[t_3]}{\longrightarrow}$ while keeping the $\bullet$'s in place. This allows to transfer the security from an initially available secure channel to a later available insecure channel.

It is important to notice that a security symbol $\bullet$ is attached to the corresponding user rather than to the corresponding channel. Channels can be replaced by cryptography, as mentioned above, but it is obvious that $\bullet$'s cannot be "created" or moved from one user to another by cryptographic transformations. In other words, the fact that a user is exclusive in a certain sense cannot be transferred to another user. Hence security symbols $\bullet$ must be created by non-cryptographic means such as authentication based on a passport or on speaker identification. This observation appears to be impossible to prove and is therefore stated as an axiom.

Axiom. *There exists no cryptographic transformation allowing to "create" a $\bullet$ or to move a $\bullet$ from one user to another.*

A typical security goal for an open network is that every pair of users (e.g., A and B), can communicate securely, i.e., over a $A \bullet\!\!\longleftrightarrow\!\!\bullet B$ cannel, at any time. Of course, a necessary condition is that they be able to communicate at all, i.e., that there exists a channel $A \overset{t}{\longleftrightarrow} B$ at any time t later than some initial system setup time. For the remainder of this section we focus our attention on security transformations rather than the availability of communication channels and therefore make the following assumption, which can in some sense be interpreted as a characterization of a reliable open networks. The assumption will be dropped again in Section 7.

Assumption 1. *Insecure channels ($\longleftrightarrow$) between every pair of users are always available.*

Theorem 1. *Under Assumption 1 it is a sufficient condition for achieving a secure channel between A and B from time t_0 on ($A \bullet\!\!\overset{t}{\longleftrightarrow}\!\!\bullet B$ for $t \geq t_0$) that one of the following four preconditions is satisfied for some $t_2 < t_0$ and $t_4 < t_0$:*

$$
\begin{array}{ll}
 & \{A \bullet\!\!\overset{t_2[t_1]}{\longrightarrow} B \text{ and } A \overset{t_4[t_3]}{\longrightarrow}\!\!\bullet B\} \\
\text{or} & \{A \bullet\!\!\overset{t_2[t_1]}{\longrightarrow} B \text{ and } A \overset{t_4[t_3]}{\longleftarrow}\!\!\bullet B\} \\
\text{or} & \{A \bullet\!\!\overset{t_2[t_1]}{\longleftarrow} B \text{ and } A \overset{t_4[t_3]}{\longrightarrow}\!\!\bullet B\} \\
\text{or} & \{A \bullet\!\!\overset{t_2[t_1]}{\longleftarrow} B \text{ and } A \overset{t_4[t_3]}{\longleftarrow}\!\!\bullet B\}
\end{array}
$$

Assuming the above axiom, this condition is also necessary.

Proof sketch. It is easy to verify that for every precondition there exists a transformation or a sequence of transformations for creating a secure channel. For instance, every confidential channel can be transformed into an authenticated channel by application of transformation (7) and the obtained scenario consisting of two complementary authenticated channels can be transformed into a secure channel by transformation (10). Now transformations (3) and (4) can be applied to complete the proof.

Remarks. It need not be assumed that $t_4 > t_2$. The precondition $A \bullet\!\!\xrightarrow{t}\!\!\bullet B$ (e.g. a trusted courier) implies the first precondition with $t_2 = t_4 = t$. It is interesting to note that conventional symmetric cryptography allows to achieve the security transformation of Theorem 1 if and only if the first (in time) of the two available channels is a confidential one. If the first of the two available channels is authentic but not confidential, then public-key cryptography is required. This observation demonstrates the significance of the discovery of public-key cryptography by Diffie and Hellman [3], especially in view of the fact that in many practical scenarios there exist authenticated channels (e.g., partner identification on telephone channels) that are not confidential.

5. Security transformations based on trust

The necessary condition of Theorem 1 is pessimistic because it states that in order to establish security in an open system, some secure channel(s) must exist between every pair of users at some time. The only solution to this quadratic (in the number of users) growth of the key distribution problem is to involve a trusted user or authority which can serve as a relay for authenticated or confidential messages. Trust is a fundamental ingredient for secure communications in open networks. Various types and degrees of trust can be distinguished (e.g., see [17]), but for the sake of simplicity of the model, such a distinction will not be made in this paper, although our model could be extended in this direction.

If a user B trusts another user or authority T to send only authenticated information (i.e., T is trusted to properly authenticate its sources of information as well as not to fraudulently distribute inaccurate information), T can connect two authenticated channels $A \bullet\!\!\xrightarrow{t_2[t_1]} T$ and $T \bullet\!\!\xrightarrow{t_4[t_3]} B$ to result in an authenticated channel from A to B, provided that $t_3 > t_2$, i.e., provided that the message on the $T \bullet\!\!\xrightarrow{t_4[t_3]} B$ channel need not be fixed before the first message is received by T on the $A \bullet\!\!\xrightarrow{t_2[t_1]} T$ channel:

$$\left.\begin{array}{c} A \bullet\!\!\xrightarrow{t_2[t_1]} T \\ T \bullet\!\!\xrightarrow{t_4[t_3]} B \\ t_3 > t_2 \\ B \text{ trusts } T \end{array}\right\} \quad\Longrightarrow\quad A \bullet\!\!\xrightarrow{t_4[t_1]} B \tag{13}$$

This transformation is a generalization of transformation (2). Note that A need not trust T.

If a user A trusts another user or authority T to treat secret information confidentially and to send it only to entities approved by A, then T can connect two confidential channels, provided that $t_3 > t_2$:

$$\left.\begin{array}{c} A \xrightarrow{t_2[t_1]}\!\bullet\, T \\ T \xrightarrow{t_4[t_3]}\!\bullet\, B \\ t_3 > t_2 \\ A \text{ trusts } T \end{array}\right\} \quad \Longrightarrow \quad T \xrightarrow{t_4[t_1]}\!\bullet\, B \tag{14}$$

The following transformation, which corresponds to the classical secret key distribution by a trusted authority, cannot be derived by combining previously described transformations. It requires additionally that A and B trust T to generate a random session key.

$$\left.\begin{array}{c} T \bullet\!\xrightarrow{t_2[t_1]}\!\bullet\, A \\ T \bullet\!\xrightarrow{t_4[t_3]}\!\bullet\, B \\ A \xrightarrow{t_5} B \\ t_5 > t_2, t_5 \geq t_4 \\ A \text{ and } B \text{ trust } T \end{array}\right\} \quad \Longrightarrow \quad A \bullet\!\xrightarrow{t_5}\!\bullet\, B \tag{15}$$

A crucial application of digital signatures (transformation (11)) is for achieving transformation (13) even when $t_3 < t_2$. Of course, no communication can take place from A to B in this case. However, if we assume the existence of an insecure channel $T \xrightarrow{t} B$ from T to B at some time $t > t_2$, then we can use transformation (11) to obtain a channel $T \bullet\!\xrightarrow{t} B$ and hence by application of (13) a channel $A \bullet\!\xrightarrow{t[t_1]} B$. The drawback of this approach is that T must participate actively in the communication from A to B.

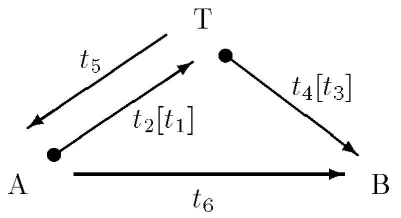

Figure 1. Connecting two authenticated channels by a trusted party T.

Therefore, a more realistic scenario, which corresponds to the well-known certification of public keys by a trusted user or authority, is shown in Figure 1, where we assume the existence of insecure channels $T \xrightarrow{t_5} A$ and $A \xrightarrow{t_6} B$. Here the interaction between A and T is independent of the message sent by T over the $T \bullet\!\xrightarrow{t_4[t_3]} B$ channel to B.

If $t_6 > t_5$ we can apply transformation (2) to obtain

$$\left.\begin{array}{c} T \xrightarrow{t_5} A \\ A \xrightarrow{t_6} B \\ t_6 > t_5 \end{array}\right\} \quad \stackrel{(2)}{\Longrightarrow} \quad T \xrightarrow{t_6[t_5]} B$$

Now transformation (11) for digital signatures yields

$$\left.\begin{array}{c} T \bullet\!\!\xrightarrow{t_4[t_3]} B \\ T \xrightarrow{t_6[t_5]} B \\ t_6 > t_4 \end{array}\right\} \quad \stackrel{(11)}{\Longrightarrow} \quad T \bullet\!\!\xrightarrow{t_6[t_5]} B$$

If B trusts T and if $t_5 > t_2$ we can now apply transformation (13) to obtain

$$\left.\begin{array}{c} A \bullet\!\!\xrightarrow{t_2[t_1]} T \\ T \bullet\!\!\xrightarrow{t_6[t_5]} B \\ t_5 > t_2 \\ B \text{ trusts } T \end{array}\right\} \quad \stackrel{(13)}{\Longrightarrow} \quad A \bullet\!\!\xrightarrow{t_6[t_1]} B$$

which together with transformation (11) for digital signatures gives the desired result, an authenticated channel from A to B:

$$\left.\begin{array}{c} A \bullet\!\!\xrightarrow{t_6[t_1]} B \\ A \xrightarrow{t_6} B \end{array}\right\} \quad \stackrel{(11)}{\Longrightarrow} \quad A \bullet\!\!\xrightarrow{t_6} B$$

Since we assumed that $t_3 > t_2$, the applications of transformations (2) and (13) are unavoidable. Hence the above derivation illustrates that an authenticated channel $A \bullet\!\!\xrightarrow{t_6} B$ can be achieved if and only if $t_2 < t_5 < t_6$ and only if B trusts T. Two applications of a digital signature scheme are required, first by T for certifying A's public key and secondly by A to authenticate actual messages. In this model of public-key certification, user A serves as a relay from T to B for his own public key certificate. When a $A \xleftarrow{t_7} B$ channel is available for some $t_7 > t_6$ and the goal of the transformations is to achieve a confidential $A \bullet\!\!\xleftarrow{t_7} B$ channel, this could be achieved by replacing the last transformation by

$$\left.\begin{array}{c} A \bullet\!\!\xrightarrow{t_6[t_1]} B \\ A \xleftarrow{t_7} B \\ t_7 > t_6 \end{array}\right\} \quad \stackrel{(9)}{\Longrightarrow} \quad A \bullet\!\!\xleftarrow{t_7} B$$

6. The necessary and sufficient condition for security in an open network

The following theorem follows from the above sequence of transformations, from Theorem 1 and from the fact that confidential channels can be transformed into authenticated channels by transformation (7).

Theorem 2. *Under Assumption 1 it is a sufficient condition for achieving an authenticated channel from A to B from time t_0 on ($A \bullet\!\!\xrightarrow{t} B$ for $t \geq t_0$) that there exists a connected path of channels from A to B such that*

(1) every channel in the path is available at some time earlier than t_0 and has a $\bullet$ attached to that end of the channel which is closer to A and
(2) user B trusts every intermediate user on the path.

Assuming the above axiom the condition is also necessary.

Corollary 3. *Under Assumption 1 it is a sufficient condition for achieving a secure channel between A and B from time t_0 on ($A \bullet\!\!\xleftrightarrow{t}\!\!\bullet B$ for $t \geq t_0$) that there exist two paths of channels according to Theorem 2, one from A to B and one from B to A. Assuming the above axiom the condition is also necessary.*

Example: Consider the somewhat artificial scenario of Figure 2: Assume that T_1, T_2 and T_3 are trusted by A and B, that the channels are available at the indicated times and that insecure channels are freely available. In order to determine the earliest time after which A and B can communicate securely, one has to find two paths as required by Corollary 3 with the smallest possible maximal path time on the paths. In this example there exist two paths from A to B, namely $A - T_3 - B$ and $A - T_1 - T_3 - B$, and one path from B to A, namely $B - T_2 - T_1 - T_3 - A$. Hence the earliest time for secure communication between A and B is $\max(t_2, t_3, t_4, t_6, t_7, \min(t_2, \max(t_1, t_3)))$.

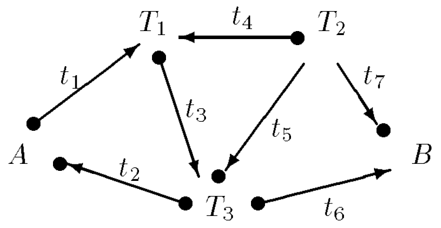

Figure 2. A security bootstrapping scenario.

7. Protocols for bootstrapping security in open networks

In the previous section we have demonstrated the theoretical limitations to establishing security in an open network. This section is devoted to reviewing within our framework some previously proposed protocols for establishing a secure channels between two users.

7.1. Protocols based on symmetric cryptography

Assume that every user shares a secret key with a trusted party T. In other words, there exist channels $T \bullet\!\!\xrightarrow{t_1}\!\!\bullet A$ and $T \bullet\!\!\xrightarrow{t_2}\!\!\bullet B$ for some time instances t_1 and

t_2. From the point of view of the required (insecure) communication channels, the simplest protocol exploits transformation (15), requiring only communication between A and B, not involving T. While this approach is used in military and diplomatic applications, it is of course completely impractical in large networks because the communication on the two channels $T \bullet\!\xrightarrow{t_1}\!\bullet A$ and $T \bullet\!\xrightarrow{t_2}\!\bullet B$ is necessarily correlated. In other words, T must generate a secret key for every pair of users and send each user the approriate secret keys (for communication with the other users) over the secure channel (e.g. by a trusted courier).

7.1.1. Message or session key relaying by a trusted server

The correlation between the secure channels available during the setup phase can be avoided at the expense of requiring T to be *on-line*. This also allows the encryption key generation to take place only when needed (session keys). The most simple such protocol, involving transformations (8), (13) and (14), is when T serves as a relay for messages. If A wants to send a message to B, he or she encrypts it with the secret key shared with T and sends it to T using a channel $A \longrightarrow T$, who decrypts and reencrypts it with the secret key shared with B, and sends the result to B using a $T \longrightarrow B$ channel.

A more reasonable protocol additionally requiring direct interaction between A and B is the so-called wide-mouthed-frog protocol proposed by Burrows [2]. Here, T merely relays a session key generated by A for communication between A and B, and all subsequent encrypted communication between A and B happens over a $A \longleftrightarrow B$ channel.

7.1.2. Session key distribution by a trusted server

The following type of protocol is used in Kerberos [13] and in KryptoKnight [9] and has the advantage that it does not require users to be capable of generating "good" encryption keys. Otway and Rees [11] have proposed a similar protocol with a different sequence of interactions with the trusted server T. Figure 3 illustrates the scenario in which T agrees on a bilateral secret key with every user during an initialization phase (in our example at time t_1 with A and at time t_2 with B).

When A wants to communicate with B she asks T to provide a session key. T sends the encrypted session key to A, together with the same session key encrypted for B (i.e., with the key shared by T and B). A can then initiate a communication with B by sending the encrypted session key.

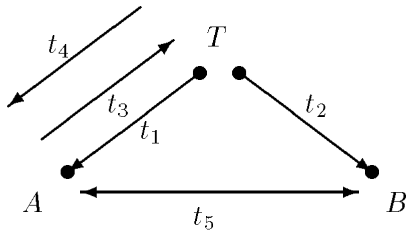

Figure 3. Session key distribution by a trusted server T.

The sequence of transformations used for obtaining a secure channel $A \bullet\!\!\xrightarrow{t_5}\!\!\bullet B$ is as follows: If $t_5 > t_4$ then the channels $T \xrightarrow{t_4} A$ and $A \xrightarrow{t_5} B$ can be connected by transformation (2):

$$\left.\begin{array}{c} T \xrightarrow{t_4} A \\ A \xrightarrow{t_5} B \\ t_5 > t_4 \end{array}\right\} \quad \overset{(2)}{\Longrightarrow} \quad T \xrightarrow{t_5[t_4]} B$$

Now transformation (3) applied to channels $T \xrightarrow{t_5[t_4]} B$ and $T \bullet\!\!\xrightarrow{t_2}\!\!\bullet B$ gives

$$\left.\begin{array}{c} T \bullet\!\!\xrightarrow{t_2}\!\!\bullet B \\ T \xrightarrow{t_5[t_4]} B \\ t_5 \geq t_2 \end{array}\right\} \quad \overset{(3)}{\Longrightarrow} \quad T \bullet\!\!\xrightarrow{t_5[t_4]}\!\!\bullet B$$

Channels $T \bullet\!\!\xrightarrow{t_1}\!\!\bullet A$ and $T \xrightarrow{t_4} A$ are combined by another application of (3):

$$\left.\begin{array}{c} T \bullet\!\!\xrightarrow{t_1}\!\!\bullet A \\ T \xrightarrow{t_4} A \\ t_4 > t_1 \end{array}\right\} \quad \overset{(3)}{\Longrightarrow} \quad T \bullet\!\!\xrightarrow{t_4}\!\!\bullet A$$

Now $T \bullet\!\!\xrightarrow{t_4}\!\!\bullet A$, $T \bullet\!\!\xrightarrow{t_5[t_4]}\!\!\bullet B$ and $A \xrightarrow{t_5} B$ can be used for key distribution according to (15):

$$\left.\begin{array}{c} T \bullet\!\!\xrightarrow{t_4}\!\!\bullet A \\ T \bullet\!\!\xrightarrow{t_5[t_4]}\!\!\bullet B \\ A \xrightarrow{t_5} B \\ t_5 > t_4 \\ A \text{ and } B \text{ both trust } T \end{array}\right\} \quad \overset{(15)}{\Longrightarrow} \quad A \bullet\!\!\xrightarrow{t_5}\!\!\bullet B$$

It may be desirable for A to generate the session key herself. In this case, a modified version of the above protocol in which $A \bullet\!\!\xrightarrow{t_3}\!\!\bullet T$ is created from $T \bullet\!\!\xrightarrow{t_1}\!\!\bullet A$ and $A \xrightarrow{t_3} T$ by application of (4), $A \bullet\!\!\xrightarrow{t_5[t_3]}\!\!\bullet B$ is created from $A \bullet\!\!\xrightarrow{t_3}\!\!\bullet T$ and $T \bullet\!\!\xrightarrow{t_5[t_4]}\!\!\bullet B$ by application of (13) and (14), and $A \bullet\!\!\xrightarrow{t_5}\!\!\bullet B$ is created from $A \bullet\!\!\xrightarrow{t_5[t_3]}\!\!\bullet B$ and $A \xrightarrow{t_5} B$ by application of (3). This sequence of transformations requires that $t_1 < t_3 < t_4 < t_5$.

7.2. Protocols based on public key certification

The major drawback of the protocols described in the previous section (and of all protocols based solely on symmetric cryptography) is that either a trusted authority T must be available on-line or the initial secure communications between different users and T must be correlated. This problem of relying on an authority for every session can be solved by using certified public keys as mentioned in Section 4. However, in a very large network such as the Internet, several trusted authorities are required to make the system practical, i.e., to provide all the paths of channels required by Corollary 3.

7.2.1. Hierarchical public key certification

The certification authorities can be organized in a hierarchy as suggested in [19], in which each authority can certify the public key of lower-level authorities. Of course, cross-certification links can be introduced when needed [19].

A simple scenario with a two-level hierarchy is shown in Figure 4: T_1 is a system-wide authority which certifies public keys of regional authorities (T_2 and T_3). A and B get certificates for their public keys from T_2 and T_3, respectively. Such a certification step consists of sending the public key, over an authenticated channel, to the higher-level authority and receiving from it over another authenticated channel the certified public key together with the public keys and certificates of all authorities on the path to T_1. Typically, it is realized by a personal registration with mutual identification. It should be pointed out that a user's public key may consist of two components, the public keys for a digital signature scheme and for a public-key cryptosystem or public-key distribution system. One public key suffices if the RSA system [12] is used.

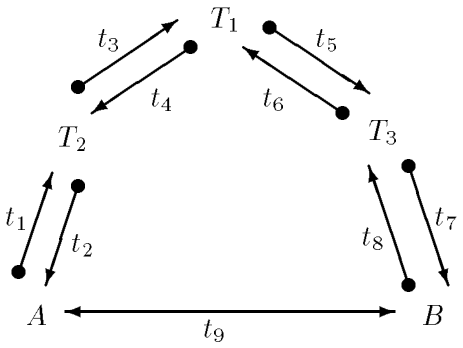

Figure 4. Hierarchical public key certification.

A formal derivation using security transformations demonstrates the constraints on the timing of the channels required to achieve a secure channel between A and B. It further illustrates the fact that all authorities on the path between A and B must be trusted by both A and B. The sequence of transformations leading to an authenticated channel $A \bullet\!\xrightarrow{t_9} B$ from A to B is shown in Figure 5.

$$\left.\begin{array}{c} T_2 \bullet\!\xrightarrow{t_2} A \\ A \xrightarrow{t_9} B \\ t_9 > t_2 \end{array}\right\} \overset{(1),(2)}{\Longrightarrow} T_2 \xrightarrow{t_9[t_2]} B \qquad \left.\begin{array}{c} T_2 \bullet\!\xrightarrow{t_3} T_1 \\ T_1 \bullet\!\xrightarrow{t_9[t_4]} B \\ t_4 > t_3 \\ B \text{ trust } T_1 \end{array}\right\} \overset{(13)}{\Longrightarrow} T_2 \bullet\!\xrightarrow{t_9[t_3]} B$$

$$\left.\begin{array}{c} T_1 \bullet\!\xrightarrow{t_4} T_2 \\ T_2 \xrightarrow{t_9[t_2]} B \\ t_2 > t_4 \end{array}\right\} \overset{(1),(2)}{\Longrightarrow} T_1 \xrightarrow{t_9[t_4]} B \qquad \left.\begin{array}{c} T_2 \bullet\!\xrightarrow{t_9[t_3]} B \\ T_2 \xrightarrow{t_9[t_2]} B \end{array}\right\} \overset{(11)}{\Longrightarrow} T_2 \bullet\!\xrightarrow{t_9[t_2]} B$$

$$\left.\begin{array}{c} T_1 \bullet\!\xrightarrow{t_5} T_3 \\ T_3 \bullet\!\xrightarrow{t_7} B \\ t_7 > t_5 \\ B \text{ trusts } T_3 \end{array}\right\} \overset{(13)}{\Longrightarrow} T_1 \bullet\!\xrightarrow{t_7[t_5]} B \qquad \left.\begin{array}{c} A \bullet\!\xrightarrow{t_1} T_2 \\ T_2 \bullet\!\xrightarrow{t_9[t_2]} B \\ t_2 > t_1 \\ B \text{ trust } T_2 \end{array}\right\} \overset{(13)}{\Longrightarrow} A \bullet\!\xrightarrow{t_9[t_1]} B$$

$$\left.\begin{array}{c} T_1 \bullet\!\xrightarrow{t_7[t_5]} B \\ T_1 \xrightarrow{t_9[t_4]} B \\ t_9 > t_7 \end{array}\right\} \overset{(11)}{\Longrightarrow} T_1 \bullet\!\xrightarrow{t_9[t_4]} B \qquad \left.\begin{array}{c} A \bullet\!\xrightarrow{t_9[t_1]} B \\ A \xrightarrow{t_9} B \end{array}\right\} \overset{(11)}{\Longrightarrow} A \bullet\!\xrightarrow{t_9} B$$

Figure 5. Exploiting the certification hierarchy of Figure 4.

Note that this sequence of transformations requires that $t_1 < t_2$, $t_3 < t_4$, $t_5 < t_7$ and $t_4 < t_2 < t_9$ as well as that B trusts T_1, T_2 and T_3. The condition $t_4 < t_2 < t_9$ follows from the fact that T_2 must provide A with T_1's public key as well as with T_1's certificate for his own public key.

For the symmetric conditions $t_8 < t_7$, $t_6 < t_5$, $t_4 < t_2$ and $t_5 < t_7 < t_9$ and if also A trusts T_1, T_2 and T_3 we can similarly obtain a channel $B \bullet\!\xrightarrow{t_9} A$. Hence transformation (10) now provides the desired secure channel: $A \bullet\!\xleftrightarrow{t_9}\!\bullet B$.

7.2.2 Non-hierarchical public key certification

In very large communication systems, hierarchical certification schemes with a tree-shaped topology are critical for two reasons. First, a single failure of one of the authorities suffices to destroy a system's operability. Second, and more importantly, both users must trust *every* authority on the certification path, i.e., a certification path can be at most as strong as its weakest link. A user's trust in an authority T_i is in most cases based on the fact that some higher-level authority once trusted T_i. In other words, a user need not only trust the honesty of the authority T at the root of the tree but, also T's ability to judge the trustworthyness of authorities it either certifies or distributes the public-key of, which again must be trusted to judge the trustworthyness of further authorities, and so on. Trust management is therefore one of the fundamental research areas in distributed system security (e.g., see [17]).

The situation is comparable to that in other large organizations such as a company or government organization: Although the president hires as executive vice-presidents people he or she considers highly capable and trustworthy, who do the same for hiring second-level managers, etc., it is nevertheless unavoidable that the company ends up having some incapable and non-trustworthy employees. The major difference compared to a certification hierarchy is that in the latter, not a single failure can be tolerated, i.e., a single dishonest authority in the path can destroy the system's security. Note that the $X.509$ framework allows for cross-certification between arbitrary intermediate authorities, thus relaxing the described problem.

However, it appears crucial in very large networks that not only the communication links, but also the certification paths be highly redundant. Zimmermann's Pretty Good Privacy (PGP) software [18] allows for a very flexible use of certificates, leaving the responsibility completely in the hands of the users. This approach exploits Theorem 2 and Corollary 3 in their full generality. However, a more general approach allowing to fully express and exploit various degrees of trust and combine certificates of varying trust levels is needed.

8. Exploiting $\longrightarrow\!\!\bullet$ channels

The theorems in Sections 4 and 6 illustrate the complete duality between authenticity and confidentiality. While authenticated channels without confidentiality ($\bullet\!\!\longrightarrow$) are used routinely in open systems, the theorems in Sections 4 and 6 suggest that confidential channels without authenticity ($\longrightarrow\!\!\bullet$) could be used equally well. It remains an interesting open question whether such channels exist in a practical application scenario.

To illustrate that such channels need not be unrealistic, consider for instance a user B with several accounts on various machines on a large network with cryptographically protected channels between his terminal and these machines. It may be reasonable to assume that an eavesdropper could simultaneously access messages sent by another user A to a few of these machines, but that he is unable to access all the messages sent to these machines at a given time. In such a scenario, a $A \longrightarrow\!\!\bullet B$ channel to be exploited in one of the transformations (5), (6), (7), (8), (12) and (14) could for instance be established by A by dividing a secret key S into various pieces such that all pieces are required to obtain any information about S, and sending (without authenticity, and preferably from accounts on different machines) the individual pieces to B's mailboxes on the various machines. If B can do the same symmetrically, these confidential channels could replace the authenticated channels needed for public-key certification. This technique may be particularly attractive for establishing secure channels with trusted authorities.

References

1. A. Birell, B. Lampson, R. Needham and M. Schroeder, A global authentication service without global trust, *Proc. IEEE Symposium on Research in Security and Privacy*, 1986, pp. 223–230.
2. M. Burrows, M. Abadi and R. Needham, A logic of authentication, *ACM Transactions on Computer Systems*, Vol. 8, No. 1, 1990, pp. 18–36.

3. W. Diffie and M.E. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, Vol. 22, No. 6, 1976, pp. 644–654.
4. M. Gasser, A. Goldstein, C. Kaufman and B. Lampson, The Digital distributed system security architecture, *Proc. 12th National Computer Security Conference*, NIST/NCSC, Baltimore, 1989, pp. 305–319.
5. J. Glasgow, G. MacEwen and P. Panangaden, A logic for reasoning about security, *ACM Transactions on Computer Systems*, Vol. 10, No. 3, 1992, pp. 226–264.
6. V.D. Gligor, S.-W. Luan and J.N. Pato, On inter-realm authentication in large distributed systems, *Proc. IEEE Conference on security and privacy*, 1992, pp. 2–17.
7. B. Lampson, M. Abadi, M. Burrows and E. Wobber, Authentication in distributed systems: theory and practice, Proc. 13th ACM Symp. on Operating Systems Principles, 1991, pp. 165–182.
8. J. Linn, Privacy enhancement for internet electronic mail: Part I, Message encipherment and authentication procedures, Internet RFC 1421, Feb. 1993.
9. R. Molva, G. Tsudik, E. Van Herreweghen and S. Zatti, "KryptoKnight Authentication and Key Distribution System", *Proc. 1992 European Symposium on Research in Computer Security (ESORICS 92)*, Toulouse (Nov.92).
10. R.M. Needham and M.D. Schroeder, Using encryption for authentication in large networks of computers, *Communications of the ACM*, Vol. 21, 1978, pp. 993–999.
11. D. Otway and O. Rees, Efficient and timely mutual authentication, *Operating systems review*, Vol. 21, No. 1, 1987, pp. 8–10.
12. R.L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, Vol. 21, No. 2, 1978, pp. 120–126.
13. J.G. Steiner, C. Neuman and J.I. Schiller, Kerberos: An authentication service for open network systems, Proceedings of *Winter USENIX 1988*, Dallas, Texas.
14. P. Syverson and C. Meadows, A logical language for specifying cryptographic protocols requirements, *Proc. IEEE Conf. on Research in Security and Privacy*, 1993, pp. 165–180.
15. J.J. Tardo and K. Alagappan, SPX: Global authentication using public key certificates, *Proc. IEEE Conf. on Research in Security and Privacy*, 1991, pp. 232–244.
16. V. Voydock and S. Kent, Security mechanisms in high-level network protocols, *ACM Computing Surveys*, Vol. 15, No. 2, 1983, pp. 135–171.
17. R. Yahalom, B. Klein and T. Beth, Trust relationships in secure systems – a distributed autentication perspective, *Proc. IEEE Conf. on Research in Security and Privacy*, 1993, pp. 150–164.
18. P. Zimmermann, PGP User's Guide, Dec. 1992, available on the Internet.
19. ISO/IEC International Standard 9594-8, Information technology – open systems interconnection – the directory, Part 8: Authentication framework, 1990.

This article was processed using the LaTeX macro package with LLNCS style