Formal Semantics for Authentication Logics

Gabriele Wedel[1]* and Volker Kessler[2]

[1] RWTH Aachen, Math. Grundlagen der Informatik, Ahornstr. 55, D-52074 Aachen
[2] Siemens AG, Corporate Research and Development, ZFE T SN 3, D-81730 Munich
Volker.Kessler@zfe.siemens.de

Abstract. We present a new BAN-like logic and a new formal semantics for logics of authentication. The main focus of this paper is on the foundation of this logic by a possible-worlds semantics. The logic was designed for implementation in the tool AUTLOG and is able to handle most kinds of protocols used in practice. The underlying logic is a K45-logic, including negation. We replace the critical idealization step by changing the set of premises. The formal semantics enables us to detect flaws in previous logics. We apply the logic to a new authentication protocol designed for UMTS.

Key Words. Formal verification, logic of authentication, cryptographic protocols, key management.

1 Motivation

Seven years ago Burrows, Abadi and Needham published their well-known BAN-logic [2] in order to analyze cryptographic protocols. In the meantime, BAN-logic has become the most widely used formal method in the analysis of cryptographic protocols despite its well-known limitations. So, obviously, people find this method a useful one to apply.

Still, so far there has been no complete logical foundation for the underlying concept. The problem is that people while applying the BAN-logic introduce new rules in order to handle the specific features of the investigated protocols. Unfortunately, it often happens that although each rule might seem reasonable alone in isolation the combination of the rules leads to unforeseen and unwanted effects which contradicts the underlying meaning, cf. Sect. 6. Therefore, a proof of the soundness of the logic is absolutely vital.

A first approach was taken by Abadi-Tuttle (AT-logic) [1]. Their idea was to give an independent formal semantics for the calculus of the BAN-logic. Actually, their use of the possible-worlds interpretation gave much insight into what is really happening during the analysis. Unfortunately, the AT-logic has two deficiences. First, the calculus of the AT-logic is not strong enough to handle all the protocols developed in security systems. For example, it is a common practice to digitally sign a hash value of a document rather than the document itself. But hash functions do not fit into the AT-logic.

* Work on this thesis was sponsored by Siemens AG.

Second, their possible-worlds semantics is not exactly compatible with the logic: the issue is the correct semantics of a formula like *P believes P sees M*. The intuitive meaning is that P reads a message M which he can verify to be M and distinguish from all other messages. Therefore P must comprehend the full structure of M. The AT-logic is not able to express conditions for comprehendability. The message-meaning rule (A11 in [1]) leads to a formula *P believes P sees M* and is thus an essential part of the concept but cannot be proved to be sound. Abadi and Tuttle have noticed this problem, but they have not yet published a solution to it.

A first solution to this problem was given by Syverson and van Oorschot (SvO) in [9]. They combine several extensions of the BAN-logic into one logic, and prove that this SvO-logic is sound on the basis of a suitable possible-worlds semantics. However, the SvO-logic is not sufficient for practical purposes because it is not able to handle partly comprehendable messages.

Independent from our activities, Syverson and van Oorschot continued working on this subject. In [10] they introduced a symbol '$*$' in the syntax in order to label incomprehendable submessages. However, they give no formal semantics to these new formulae, and their calculus is not strong enough to compute the comprehended submessages. These messages must be determined during the idealization step and added to the premises.

Our approach marks possibly incomprehensible messages and derives the comprehended submessages by analyzing the properties of the used functions. We give a formal semantics which enables us to prove the soundness of our logic and to decide whether a given rule is valid or not. By applying our model interpretation we found incorrect rules in BAN[2], GNY [6], AT [1], and a former version of AUTLOG [7].

The most criticized point in BAN-logic is the so-called idealization step because of its vagueness and ambiguity. We decided to replace the idealization of messages by additional premises. The way a message is interpreted by the receiver is described in the premises. Consequently, formulae are no longer allowed to be part of messages. This leads to a clear distinction between the pure protocol, i.e. what is actually transmitted, and the assumptions which must hold so that the protocol can work.

Furthermore, we introduce negation in the BAN-logic which gives a clear solution of the symmetry problem and thus reflection attacks can be handled. The applicability of the logic is demonstrated by an example.

2 Syntax

Messages. First of all we introduce the language of our logic. Our goal was to deal with the essential properties of a wide variety of cryptographic tools by introducing a minimum of different names. The so-called set of basic items $\Sigma = \mathcal{P} \cup \mathcal{M}_0 \cup \mathcal{K} \cup \mathcal{F}$ consists of

- a set of agents $\mathcal{P} = \{P, Q, S, T, ...\}$, who communicate with each other,
- a set of public key schemes $\mathcal{K} = \{K_p, K_q, ...\}$,

- a set of basic messages and shared keys $\mathcal{M}_0 = \{M, N, data, K_{pq}, ...\}$, and
- a set of computable functions $\mathcal{F} = \{enc, h, \sigma\}$.

A public key scheme K consists of a private component K^- and a public component K^+ and can be used for signature/verification, encryption/decryption, and key agreement. The symbol h denotes all hash functions including message authentication codes and σ denotes signatures without message recovery. $enc(K, M) = \{M\}_K$ denotes both encryption of a message M with K (symmetric or asymmetric encryption) and a signature with message recovery (e.g. RSA-signature, Rabin-signature). The main issue for messages denoted by $\{M\}_K$ is that cleartext M can be derived from $\{M\}_K$ under knowledge of the inverse key K^{-1} (which is the corresponding component for public key schemes and equal to K in the symmetric case).

For any function $F \in \mathcal{F}$, $F(M)$ denotes the ***structure*** of the message computed by F on M. Even if the *values* of two different computations $F(M)$ and $G(N)$ are the same the *messages* $F(M)$ and $G(N)$ are considered as different because the identifier of a message always includes the way it has been computed.

These basic items can be put together to more complex messages. The set of messages $\mathcal{M}$ consists of

- the names of agents and basic messages in $\mathcal{P} \cup \mathcal{M}_0$,
- the components K^- and K^+ of the public key schemes $K \in \mathcal{K}$,
- lists of messages $(M_1, ..., M_n)$ with all $M_i \in \mathcal{M}$,
- computed messages $F(M)$ for $F \in \mathcal{F}$ and $M \in \mathcal{M}$, and
- derived keys $\alpha(\{K_p, K_q\})$ for key agreement key schemes $K_p, K_q \in \mathcal{K}$.

A derived key $\alpha(\{K_p, K_q\})$ (e.g. Diffie-Hellman key) can be computed either by K_p^- and K_q^+ or by K_q^- and K_p^+. Since both ways lead to the same shared key we chose a common notation for both and separated α from $\mathcal{F}$.

Localized Messages. An important problem is how to get information about the inner structure of a message. An agent receives the *value* of a computation and has to verify the expected structure of the messsage. Let M_P denote a message M in the view of P. We call such a message *localized towards* P. P does not necessarily understand M_P. Especially, we have to consider the case that an agent may only be able to verify parts of this structure. For example, P receives a list including a cryptogram and a hash value: $P\, sees\, (\{X\}_K, h(M))$. Now assume that P cannot decrypt the ciphertext but knows M. We can express his comprehension by the formula $P\, believes\, P\, sees\, (\{X\}_K, h(M))_P$ which is equivalent to $P\, believes\, P\, sees\, ((\{X\}_K)_P, h(M))$ under the described conditions. The latter cannot be further reduced because P does not comprehend $\{X\}_K$.

Let $\mathcal{M}_\mathcal{P}$ be the *set of generalized messages*. This set is built similar to $\mathcal{M}$ with the additional feature being closed under localization, i.e. $\mathcal{M}_\mathcal{P}$ consists of

- the agents and basic messsages in $\mathcal{P} \cup \mathcal{M}_0$,
- the components K^- and K^+ of key schemes $K \in \mathcal{K}$,
- lists of generalized messages $(M_1, ..., M_n)$ with all $M_i \in \mathcal{M}_\mathcal{P}$,
- computed messages $F(M)$ with $F \in \mathcal{F}, M \in \mathcal{M}_\mathcal{P}$,
- derived keys $\alpha(\{K_p, K_q\})$ for key schemes $K_p, K_q \in \mathcal{K}$, and

- localized messages M_P for agents $P \in \mathcal{P}$ and messages $M \in \mathcal{M}_\mathcal{P}$.

In contrast to other BAN-logics we do not allow any formulae as messages.

Formulae. The set of formulae of our logic, Φ, consists of the following formulae (with messages $M, N \in \mathcal{M}_\mathcal{P}$, agents $P, Q \in \mathcal{P}$, and formulae $\varphi, \psi \in \Phi$):

$P\overset{K}{\leftrightarrow}Q$ for keys $K \in \mathcal{M}$ [3],

$\epsilon\overset{K}{\mapsto}Q$ for asymmetric encryption key schemes $K \in \mathcal{K}$,

$\sigma\overset{K}{\mapsto}Q$ for signature key schemes $K \in \mathcal{K}$,

$\alpha\overset{K}{\mapsto}Q$ for key agreement key schemes $K \in \mathcal{K}$.

$fresh(M)$: Message M has been created in the current protocol run.

$good_F(M)$: $M \in \mathcal{M}_\mathcal{P}$ is suitable for key derivation with the function $F \in \mathcal{F}$.

$M \equiv N$: The concept of equivalence of generalized messages is necessary in order to axiomize the computing of the comprehendable submessages in a localized submessage M'_P of M.

$P\, sees\, M$: Agent P was able to read M as a submessage of a received message.

$P\, said\, M$: Agent P has sent the message M and has been conscious of sending it at that time.

$P\, says\, M$: Agent P has sent the message M knowingly and recently.

$P\, has\, M$: P knows message M and can use it for further computations.

$P\, recognizes\, M$: Either P has reason to believe that M is not a random string but willingly constructed or that M is a random string already known to P.

$P\, controls\, \varphi$: P is able to decide whether φ is correct or not.

$P\, believes\, \varphi$: P has strong evidence that φ is correct as far as P can understand the messages in φ, that means except the localized submessages.

$\neg\varphi, \varphi \wedge \psi$ negation and conjunction.

It may surprise that we drop formulae from the set of messages. Idealization, i.e. attaching formulae to messages whenever this formula is necessary to describe the meaning of a certain message, seems to be one of the main issues in BAN-Logic. But this process is widely criticized because it is quite arbitrary, scarcely formalized, and it complicates the interpretation of a successful analysis. We do not need this sort of idealization.

Instead of attaching a formula φ to a message M (e.g. substituting a key K by the formula $A\overset{K}{\leftrightarrow}B$) we leave the message transactions unchanged and add another protocol assumption, e.g. $B\, believes\ (A\, says\, K \longrightarrow A\, believes\, A\overset{K}{\leftrightarrow}B)$. Together with "$B\, believes\, A\, says\, K$" this will lead to "$B\, believes\, A\, believes\, A\overset{K}{\leftrightarrow}B$" by application of the rationality rule (axiom K in modal logic). An expression like that one was substituted in the AT-logic [1] by "$B\, believes\, A\, says\, A\overset{K}{\leftrightarrow}B$" for two reasons. First, the meaning of a formula "$B\, believes\, A\, believes$ ..." was not clear and, second, the axiom $P\, said\, M \wedge fresh(M) \longrightarrow P\, believes\, M$ is obviously not valid in general. But, now the se-

[3] Any message $K \in \mathcal{M}$ can be used in the formula $P\overset{K}{\leftrightarrow}Q$.

mantics are clarified and B *believes* (A *says* $M \longrightarrow A$ *believes* φ) is formulated as an ***assumption*** for this protocol, and not as a general rule.

The main issue we have to check in order to derive B *believes* A *believes* φ is whether B does really comprehend the message M.

3 Formal Semantics

We need to define a formal semantics in order to prove soundness. First we concentrate on the non-modal formulae, then we give a semantics for the modal formulae P *believes* ϕ. We have to design models which simulate the run of a protocol. Such models have already been given in [1, 9, 10]. They have to be modified for our purposes.

3.1 The Model

Runs. A run r can be thought as an infinite chain of states starting at some time $k_r \leq 0$ in the past. The point 0 describes the starting point of the current protocol run and divides past and present. A state can be changed by one of the following actions:

$send_P(M, Q)$: P sends message M to agent Q.

$receive_P(M)$: P reads a message which has been sent before. Since we assume an open environment the message could have been meant for another agent.

$generate_P(M)$: P generates a new message. This action does not stand for computing new messages from old messages but for creating new basic items like random strings.

$name_P(M, N)$: P gives a new basic name $N \in \Sigma$ to a received message M. Thereby he ignores any possible structure of M. We introduce this action in order to formalize P's understanding of the inner structure of a compounded message.

Without restriction we can assume that only one action takes places at a certain state. The action at time k in run r is denoted by $H^{(r,k)}$. Thus a run is given by an infinite chain of actions $(H^{(r,k)})_{k \geq k_r}$. An action can change the knowledge base of an agent, including all messages which P can use - either generated or received messages:

Definition 1. For a given sequence of actions $r = (H^{(r,k)})_{k \geq k_r}$ we define the *knowledge base* $S_P^{(r,k)}$ of agent P at time $k \geq k_r$ as a set of all messages which P knows:

$$\begin{aligned} S_P^{(r,k)} \overset{\text{def}}{=} \{M \mid \exists k', N : \quad & k_r \leq k' \leq k \land \\ & H^{(r,k')} \in \{generate_P(M), receive_P(M), name_P(N, M)\}\,\} \end{aligned}$$

Of course, P can use the elements of $S_P^{(r,k)}$ for further computations, so that his knowledge increases. This is described by the set $\overline{S}$:

Definition 2. $\overline{S}$ is defined as the smallest set including S and

- each list $(M_1, ..., M_n)$ if $M_1, ..., M_n \in \overline{S}$,
- each component M_i if $(M_1, ..., M_n) \in \overline{S}$,
- each computation $F(X)$ for $F \in \mathcal{F}$ if $X \in \overline{S}$,
- each derived key $\alpha(\{K_p, K_q\})$ if $K_p^-, K_q^+ \in \overline{S}$, and
- each cleartext M if $\{M\}_K$, $K^{-1} \in \overline{S}$.

Now we can formally describe a run:

Definition 3. A *run* $r = (H^{(r,k)})_{k \geq k_r}$ is a sequence of actions

$$H^{(r,k)} \in \{send_P(M,Q), receive_P(M), generate_P(M), name_P(M,N)\}$$

with $M, N \in \mathcal{M}$,$P, Q \in \mathcal{P}$ and obeying the following conditions:

1. Only computable texts can be sent:
$$H^{(r,k)} = send_P(M,Q) \Longrightarrow M \in \overline{S_P^{(r,k)}}.$$
2. Only sent messages can be received:
$$H^{(r,k)} = receive_P(M) \Longrightarrow \exists k' < k, Q \in \mathcal{P} : \quad H^{(r,k')} = send_Q(M,S)$$
3. Only basic items can be generated:
$$H^{(r,k)} = generate_P(M) \Longrightarrow M \in \Sigma.$$
4. Only known messages can be named with basic names:
$$H^{(r,k)} = name_P(M,N) \Longrightarrow M \in \overline{S_P^{(r,k)}} \wedge N \in \Sigma.$$

Finally we can fix a certain moment $k \geq k_r$ in a run and obtain *points* (r, k) which are suitable as models for non-modal formulae. But first we have to investigate which parts of a message are comprehensible with respect to a given knowledge base.

The Function *sight*. The central concept of our logic lies in deriving information from received messages. Therefore, we have to focus on the question of how much information an agent can get out of a certain message. For example, consider the case where an agent P sees a cryptogram $\{M\}_K$. If P does not know the decryption key K^{-1}, he will not know anything about the structure of this message: it could be either a ciphertext, or a hash value, or simply a random string.

To describe the information which P can derive from a message it is sufficient to replace all unreadable, i.e. not verifiable, submessages by a certain symbol '$*$' representing a bitstring whose meaning or structure is not specified. We obtain such a projection by introducing a function $sight_P^{(r,k)}$ which maps each message M to a message over an extended set of basic messages $\mathcal{M}_0 \cup \{*\}$ which has to be carefully defined for every kind of message.

A ciphertext $\{M\}_K$ can be recognized as a ciphertext if it can be decrypted with the decryption key K^{-1} and the result M allows the conclusion that the correct decryption key has been used. Therefore it is sufficient if the result does not look like randomly, i.e. it must contain something recognizable:

$$sight_P^{(r,k)}(\{M\}_K) := \begin{cases} \left\{sight_P^{(r,k)}(M)\right\}_K, & \text{if } K^{-1} \in \overline{S_P^{(r,k)}},\ sight_P^{(r,k)}(M) \neq *, \\ *, & \text{else.} \end{cases}$$

Hash values can only be verified by applying the hash function again, not by inverting the hash function. Therefore the hashed value must be known:

$$sight_P^{(r,k)}(h(M)) := \begin{cases} h(sight_P^{(r,k)}(M)), & \text{if } M \in \overline{S_P^{(r,k)}}, \\ *, & \text{else.} \end{cases}$$

Signatures without message recovery $\sigma(K^-, M)$ must be handled separately. In order to verify them, you have to know M and the public parameter K^+:

$$sight_P^{(r,k)}(\sigma(K^-, M)) := \begin{cases} \sigma(K^-, sight_P^{(r,k)}(M)), & \text{if } K^+, M \in \overline{S_P^{(r,k)}}, \\ *, & \text{else.} \end{cases}$$

Key agreement keys can be verified by using one of the necessary key pairs:

$$sight_P^{(r,k)}(\alpha(\{K_q, K_r\})) := \begin{cases} \alpha(\{K_q, K_r\}), & \text{if } K_q^-, K_r^+ \in \overline{S_P^{(r,k)}}, \\ *, & \text{else.} \end{cases}$$

Concerning lists we assume that there exists information about the format of the list *if* at least one component of the list is recognizable:

$$sight_P^{(r,k)}((M_1, ..., M_n)) := \begin{cases} (sight_P^{(r,k)}(M_1), ..., sight_P^{(r,k)}(M_n)), & \\ \qquad \text{if } \exists i \in \{1...n\} \text{ with } sight_P^{(r,k)}(M_i) \neq *, & \\ *, \text{ else.} & \end{cases}$$

A basic item is recognizable if either it has been generated by the agent himself or the agent has read it anywhere and uses it as a basic item. Note that in the latter case he cannot be sure that it really is a basic item. The receiver might think that a given number is completely random, whereas it is actually the hash value of a contract about a new vacuum cleaner! For $N \in \Sigma$ we define:

$$sight_P^{(r,k)}(N) := \begin{cases} N, \text{ if } \exists k' \leq k : \ H^{(r,k')} \in \{generate_P(N), name_P(M, N)\}, \\ *, \text{ else.} \end{cases}$$

What agent P can recognize in a message localized towards Q depends on what Q can recognize in the message:

$$sight_P^{(r,k)}(X_Q) := sight_P^{(r,k)}(sight_Q^{(r,k)}(X)).$$

And finally, P cannot recognize anything in a hidden message:

$$sight_P^{(r,k)}(*) := *.$$

Instantiation. For the semantics for the non-modal formulae we must link "real" messages in $\mathcal{M}$ with generalized messages in $\mathcal{M}_{\mathcal{P}}$. We call a message $M \in \mathcal{M}$ an *instance* $M \preceq X$ of a generalized message X iff M and X are identical except for all localized submessages in X. Formally:

- Every message in $\mathcal{M}$ is an instance of itself:
$$\forall M \in \mathcal{M} \qquad M \preceq M.$$
- If M is an instance of X, then $F(M)$ is an instance of $F(X)$ as well:
$$M \preceq X \quad \Rightarrow \quad F(M) \preceq F(X).$$
- A list is an instance of a localized list iff all corresponding components are instances of a localized list:
$$(\forall i \in \{1, ..., n\} \quad M_i \preceq X_i) \quad \Leftrightarrow \quad (M_1, ..., M_n) \preceq (X_1, ..., X_n).$$
- Let $X \in \mathcal{M}_{\mathcal{P}}$. Then each message M is an instance of the localized message X_P :
$$\forall X \in \mathcal{M}_{\mathcal{P}} \, \forall M \in \mathcal{M} \qquad M \preceq X_P.$$

It follows directly that the only instance of a message $M \in \mathcal{M}$ is M itself.

3.2 Semantics for the non-modal formulae

Now we can fix the semantics of all non-modal formulae relative to a point (r, k). These semantics are quite straightforward and some of them are known from [1, 9, 10], but they have to be generalized to extended formulae possibly containing localized messages.

The semantics for negation and conjunction are classical:

$$(r, k) \models \neg\varphi \quad \overset{\text{def}}{\Longleftrightarrow} \quad not \ (r, k) \models \varphi,$$
$$(r, k) \models \varphi \wedge \psi \overset{\text{def}}{\Longleftrightarrow} (r, k) \models \varphi \,\text{and}\, (r, k) \models \psi.$$

P has X. P possesses a message iff one of its instances is computable from his knowledge base:

$$(r, k) \models P \, has \, X \overset{\text{def}}{\Longleftrightarrow} \exists M \preceq X \text{ with } M \in \overline{S_P^{(r,k)}}.$$

P sees M. P sees a message M iff he is able to compute it as a submessage of a received message. Let $\text{seensub}_S(M)$ be the smallest set containing M itself, containing each component M_i if it contains the list $(M_1, ..., M_n)$, and containing the deciphered text M if it contains the cryptogram $\{M\}_K$ and K^{-1} is an element of $\overline{S}$, i.e. $\text{seensub}_S(M)$ is the closure of M under projection and decryption.

$$(r, k) \models P \, sees \, X \overset{\text{def}}{\Longleftrightarrow} \exists M \preceq X \; \exists k' \leq k \; \exists N \in \mathcal{M} : \\ H^{(r,k')} = receive_P(N) \,\text{and}\, M \in \text{seensub}_{S_P^{(r,k)}}(N).$$

P said M (resp. **P says M**). P has sent the (sub)message M once (respectively in the current run). The problem is that P might have forwarded some message not knowing the content of it. The idea is to make P "responsible" only for all those submessages he could have known. This depends, of course, on P's knowledge in the state when P sent M. These submessages are given by the following set:

Let $\mathrm{saidsub}_S(M)$ be the smallest set including M itself, and containing each component M_i if it contains $(M_1, ..., M_n)$, containing the decrypted text M, if it contains the cryptogram $\{M\}_K$ and $K^{-1} \in \overline{S}$, containing the preimage X if it contains $F(X)$ and $X \in \overline{S}$. Thus we get:

$$(r,k) \models P\,said\,X \overset{\text{def}}{\Longleftrightarrow} \exists M \preceq X\, \exists k' \leq k\, \exists N \in \mathcal{M} :$$
$$H^{(r,k')} = send_P(N,Q) \text{ and } M \in \mathrm{saidsub}_{S_P^{(r,k')}}(N),$$
$$(r,k) \models P\,says\,X \overset{\text{def}}{\Longleftrightarrow} \exists M \preceq X\; \exists k' \in \{0,...,k\}\; \exists N \in \mathcal{M} :$$
$$H^{(r,k')} = send_P(N,Q) \text{ and } M \in \mathrm{saidsub}_{S_P^{(r,k')}}(N).$$

P recognizes M. This means that there is something within M known by P. It is either redundancy with respect to P's knowledge or M is a random number already known by P. Defining a formal semantics is straightforward and does not need any new concepts in our formal model:

$$(r,k) \models P\,recognizes\,X \overset{\text{def}}{\Longleftrightarrow} sight_P^{(r,k)}(X) \neq *.$$

In section 6 we will compare our concept of recognizability with the GNY-concept [6, p.246].

P controls φ. P is competent in judging φ. It is sufficient and realistic to define this condition to hold only at the actual point:

$$(r,k) \models P\,controls\,\varphi \overset{\text{def}}{\Longleftrightarrow} \text{If } (r,k) \models P\,believes\,\varphi \text{ then } (r,k) \models \varphi.$$

Note that our definition has to be different from AT and SvO because we do not allow formulae as messages.

fresh(M). M has not been used before the current run has started. Let submsgs (M) be the set of all syntactic submessages of M. (For example, if $M = F(N)$, then N is a syntactical submessage of M, etc.) A generalized message shall be *fresh* if every instance is fresh (so there is no doubt about its freshness):

$$(r,k) \models fresh(X) \overset{\text{def}}{\Longleftrightarrow} \forall M \preceq X :$$
$$M \notin \bigcup \{ \mathrm{submsgs}(N) \mid \exists k \leq 0 : \quad H^{(r,k)} = send_P(N) \}$$

Since keys are elements of $\mathcal{M}$ they do not contain localized submessages:

$\mathbf{P}\overset{\mathbf{K}}{\leftrightarrow}\mathbf{Q}$. K is a good shared key if it is only used by P and Q:

$$\begin{aligned}(r,k) \models P\overset{K}{\leftrightarrow}Q \overset{\text{def}}{\iff} \forall k' \leq k : \text{If } (r,k') \models R\, said\, F(K,M) \text{ then}\\ (r,k') \models R\, sees\, F(K,M) \text{ or } R \in \{P,Q\}.\end{aligned}$$

$\epsilon\overset{\mathbf{K}}{\mapsto}\mathbf{P}$. K is a good public encryption key for P in the sense that only P does use the corresponding secret key K^- for decryption. Similar to [9, p.19] we define:

$$\begin{aligned}(r,k) \models \epsilon\overset{K}{\mapsto}P \overset{\text{def}}{\iff} \forall k' \leq k :\ (\forall X : (r,k') \models Q\, sees\, \{X\}_{K^+} \Rightarrow (r,k') \models Q\, sees\, X)\\ \Rightarrow\ Q = P.\end{aligned}$$

$\sigma\overset{\mathbf{K}}{\mapsto}\mathbf{P}$. P's signature key should only be used by P:

$$\begin{aligned}(r,k) \models \sigma\overset{K}{\mapsto}P \overset{\text{def}}{\iff} \forall k' \leq k : \text{If } (r,k') \models Q\, said\, F(K^-,M) \text{ then}\\ Q = P \text{ or } (r,k') \models Q\, sees\, F(K^-,M).\end{aligned}$$

$\alpha\overset{\mathbf{K}}{\mapsto}\mathbf{P}$. Key agreement keys are difficult to handle. We refer to [9, p.20]:

$$(r,k) \models \alpha\overset{K_p}{\mapsto}P \quad \overset{\text{def}}{\iff}$$

There exists a second key-agreement-scheme K_q of agent Q building a good key together with K_p:

$$\exists Q, K_q \quad (r,k) \models P\xleftrightarrow{\alpha(\{K_p,K_q\})}Q,$$

whereas for all agents S and their key schemes K_s it is the case that if $(r,k) \not\models P\xleftrightarrow{\alpha(\{K_p,K_s\})}S$ then there is no agent R being able to derive a good shared key for R and S using K_s:

$$\forall R, K_r \quad (r,k) \not\models R\xleftrightarrow{\alpha(\{K_r,K_s\})}S.$$

$\mathbf{good_F(X)}$. This formula is used for key derivation parameters ([4]):

$$\begin{aligned}(r,k) \models good_F(X) \overset{\text{def}}{\iff} \forall M \preceq X\ \forall k' \leq k \text{ and for all keys } K\text{:}\\ \text{If } (r,k') \models P\overset{K}{\leftrightarrow}Q \text{ then } (r,k') \models P\xleftrightarrow{F(K,M)}Q.\end{aligned}$$

This definition leads back to the semantics for good shared keys. The essential condition for M being a good parameter for key derivation is that for all good secret keys K for P and Q neither P nor Q gives away the message $F(K,M)$, so nobody except P and Q can know about it.

$\mathbf{X} \equiv \mathbf{Y}$. Two generalized messages are called equivalent iff they are identical after substituting each localized subformula X_P by $sight_P^{(r,k)}(X)$. Therefore we introduce the function $sight^{(r,k)}$ replacing each localized submessage N_P by $sight_P^{(r,k)}(N) \in \mathcal{M}^*$.

$$(r,k) \models X \equiv Y \quad \overset{\text{def}}{\Longleftrightarrow} \quad sight^{(r,k)}(X) = sight^{(r,k)}(Y).$$

3.3 Semantics for the Modal Formulae

The basic idea is to construct a model consisting of a set of *possible worlds* for the non-modal formulae:

$$\mathcal{W} = \{(r,k) \mid r \text{ is a run and } k \geq k_r\}.$$

The agents stay in one of these worlds but they do not know which one because the agents only realize parts of their reality letting a set of worlds seem possible. In order to formalize this, we have to fix a possibility relation $\sim_P \subset \mathcal{W} \times \mathcal{W}$ for each agent. $w \sim_P w'$ should be the case iff agent P staying in world w keeps w' as possible. This is the case if P cannot distinguish between w and w'. Now we say that 'P believes that φ is true' means that φ is true in all worlds P considers possible.

The main issue is to define such a possibility relation $\sim_P$. The question is how much an agent realizes about the point (r,k) at which he is a member.

First, each agent has some assumptions like the existence of good keys or confidential key servers etc. These assumptions restrict his set of possible worlds. In order to model this formally we have to fix a subset $\mathcal{W}_P \subset \mathcal{W}$ of "good-natured" worlds obeying all these assumptions. The only restriction we make on this set in our formal model is that a "bad-natured" world cannot become "good-natured" by adding an action:

$$k \geq k_r \text{ and } (r,k) \notin \mathcal{W}_P \quad \Rightarrow \quad (r,k+1) \notin \mathcal{W}_P.$$

Second, we have to extract all information about (r,k) of which P is aware and which helps P to distinguish his real world from others. Therefore, we restrict the "*global history*" of all actions until moment k to the "*local history*" consisting of all those actions P has performed himself considering the chronological order: let $\mathcal{H}_P^{(r,k)}$ be the sequence $(H^{(r,k_0)}, \ldots, H^{(r,k_n)})$, $H^{(r,k_i)} \in \{send_P(M,Q), receive_P(M), generate_P(N), name_P(M,N)\}$, with $Q \in \mathcal{P}, M \in \mathcal{M}, N \in \Sigma$ of all actions $H^{(r,k')}$, $k_r \leq k' \leq k$, performed by P himself.

In addition, we have to restrict the local history to all the information of which P is aware. Therefore, we extend the definition of $sight_P^{(r,k)}$ to sequences of actions in the canonical manner by replacing all messages M by $sight_P^{(r,k)}(M)$. A point (r',k') shall be possible for P in (r,k) if (r',k') belongs to the good-natured worlds and if P cannot distinguish between (r,k) and (r',k'):

Definition 4.
$(r,k) \sim_P (r',k') \overset{\text{def}}{\Longleftrightarrow} (r',k') \in \mathcal{W}_P \wedge sight_P^{(r,k)}(\mathcal{H}_P^{(r,k)}) = sight_P^{(r',k')}(\mathcal{H}_P^{(r',k')})$.

Definition 5. P believes that ϕ is true iff it is true in every possibly true world:

$$(\mathcal{W}, (\mathcal{W}_P)_{P\in\mathcal{P}}) \models_{(r,k)} P\, believes\, \phi \overset{\text{def}}{\Longleftrightarrow}$$
$$\forall r', k' \quad (r,k) \sim_P (r',k') \Longrightarrow (\mathcal{W}, (\mathcal{W}_P)_{P\in\mathcal{P}}) \models_{(r',k')} \phi.$$

For non-modal formulae the expression $(\mathcal{W}, (\mathcal{W}_P)_{P\in\mathcal{P}}) \models_{(r',k')} \phi$ is the same as $(r',k') \models \phi$. If the system $(\mathcal{W}_P)_{P\in\mathcal{P}}$ is fixed $(\mathcal{W}, (\mathcal{W}_P)_{P\in\mathcal{P}}) \models_{(r,k)} P\, believes\, \phi$ will be abbreviated by $(r,k) \models P\, believes\, \phi$.

Let $\mathcal{AUT}$ consist of all structures $(\mathcal{W}, (\mathcal{W}_P)_{P\in\mathcal{P}})$ with $\mathcal{W}_P \subset \mathcal{W}$ for all $P \in \mathcal{P}$. Then every world of every structure satisfies the well-known axioms

K (Rationality Rule) $P\, believes\, (\varphi \rightarrow \psi) \longrightarrow (P\, believes\, \varphi \rightarrow P\, believes\, \psi)$,

4 (Positive Introspection) $P\, believes\, \varphi \longrightarrow P\, believes\, P\, believes\, \varphi$,

5 (Negative Introspection) $\neg P\, believes\, \varphi \longrightarrow P\, believes\, \neg P\, believes\, \varphi$.

characterizing our modal logic as a K45-logic ([3]). This logic corresponds to models with transitive and euclidian possibility relation.

It can happen that these structures are not serial: if the assumptions of agent P defining the set $\mathcal{W}_P$ are not consistent with the information P has about his real world (r,k), then there is no world he considers possible. In this case the antecedent in Definition 5 is never true which implies that $(r,k) \models P\, believes\, \varphi$ holds for all formulae φ. This result does not mean that the logic is inconsistent, it simply means that the so-called axiom D: $P\, believes\, \varphi \longrightarrow \neg P\, believes\, \neg\varphi$ does not hold.

3.4 Stability

Since we allow negation our logic is not monotone. According to their semantical definition the formulae *has*, *sees*, *said*, *says*, *recognizes*, and *fresh* are *stable*, i.e. $(r,t) \models \varphi$ implies $(r,t+1) \models \varphi$. Our restriction on "good-natured" worlds (sect. 3.3) makes sure that formulae like $P\, believes\, \varphi$ with a stable formula φ are also stable.

According to the chosen definition the instable formulae are *controls*, $P \overset{K}{\leftrightarrow} Q$, $\overset{K}{\mapsto} P$, *good*, and some negated formulae $\neg\varphi$. In the practical analysis of a protocol, these instable formulae only occur as initial beliefs, i.e. they are within the scope of a *believe*-operator and are stated as an intial assumption (cf. sect. 8).

So, whenever a possibly instable formula is applied during a derivation after the receipt of a message, one has to check if this initial belief is still justified. For example, a belief like $A\, believes\, \neg A\, said\, \{M\}_K$ might be reasonable during the first two protocol steps but it might be in contradiction to the third message where A actually does send $\{M\}_K$.

4 Calculus

The symbol F represents any function in $\mathcal{F} = \{enc, \sigma, h\}$ and H is a one way function out of $\{h, \sigma\}$. X, X_i, Y, Z represent generalized messages in $\mathcal{M}_{\mathcal{P}}$ whereas M, M_i, K, K_p, K_q belong to the message set $\mathcal{M}$. P and Q represent agents in $\mathcal{P}$ and φ, ψ are formulae in Φ. $(K^+)^{-1}$ $/(K^-)^{-1}$ stand for the corresponding inverse keys $K^-/\ K^+$. For symmetric cryptosystems the decryption key K^{-1} equals the encryption key K .

Inference Rules.

MP If φ and $(\varphi \rightarrow \psi)$ then ψ

M If φ is a theorem then $P\ \mathit{believes}\ \varphi$ is a theorem.

A theorem is a formula which can be derived from axioms alone.

The axioms are all instances of tautologies of propositional calculus and the following axiom schemas:

Modalities.

K $P\ \mathit{believes}\ \varphi \wedge P\ \mathit{believes}\ (\varphi \rightarrow \psi) \longrightarrow P\ \mathit{believes}\ \psi$

4 $P\ \mathit{believes}\ \varphi \longrightarrow P\ \mathit{believes}\ P\ \mathit{believes}\ \varphi$

5 $\neg P\ \mathit{believes}\ \varphi \longrightarrow P\ \mathit{believes}\ \neg P\ \mathit{believes}\ \varphi$

Jurisdiction. If P controls φ and believes that φ is true then it is true indeed:

J $(P\ \mathit{controls}\ \varphi \wedge P\ \mathit{believes}\ \varphi) \longrightarrow \varphi$

Possession.

H1 $P\ \mathit{sees}\ X \longrightarrow P\ \mathit{has}\ X$

H2 $P\ \mathit{has}\ X_1 \wedge \ldots \wedge P\ \mathit{has}\ X_n \longrightarrow P\ \mathit{has}\ (X_1, \ldots, X_n)$

H3 $P\ \mathit{has}\ X \longrightarrow P\ \mathit{has}\ F(X)$

H4 $R\ \mathit{has}\ K_p^- \wedge R\ \mathit{has}\ K_q^+ \longrightarrow R\ \mathit{has}\ \alpha(\{K_p, K_q\})$

Recognizability. A message is recognizable if any component is recognizable or if it can be verified by a specific computation:

R1 $P\ \mathit{recognizes}\ X_i \longrightarrow P\ \mathit{recognizes}\ (X_1, \ldots, X_n)$

R2 $P\ \mathit{recognizes}\ X \wedge P\ \mathit{has}\ K^{-1} \longrightarrow P\ \mathit{recognizes}\ enc(K, X)$

R3 $P\ \mathit{has}\ M \longrightarrow P\ \mathit{recognizes}\ H(M)$

R4 $P\ \mathit{has}\ (K^+, M) \longrightarrow P\ \mathit{recognizes}\ \sigma(K^-, M)$

R5 $R\ \mathit{has}\ K_p^- \wedge R\ \mathit{has}\ K_q^+ \longrightarrow R\ \mathit{recognizes}\ \alpha(\{K_p, K_q\})$

Freshness. A message is fresh if any component having been used to compute it is fresh:

F1 $\quad fresh(X_i) \longrightarrow fresh((X_1, ..., X_n))$

F2 $\quad fresh(X) \longrightarrow fresh(F(X))$

F3 $\quad fresh(K_p) \longrightarrow fresh(\alpha(\{K_p, K_q\}))$

Seeing.

SE1 $\quad P\, sees\, (X_1, ..., X_n) \longrightarrow P\, sees\, X_i$

SE2 $\quad P\, sees\, enc(K, X) \wedge P\, has\, K^{-1} \longrightarrow P\, sees\, X$

Saying.

NV $\quad P\, said\, X \wedge fresh(X) \longrightarrow P\, says\, X$

SA1 $\quad P\, said\, (X_1, ..., X_n) \longrightarrow P\, said\, X_i$

SA2 $\quad P\, says\, (X_1, ..., X_n) \longrightarrow P\, says\, X_i$

Suppose P said a hash value $h(X)$. In order to conclude that P also said X we must exclude the case that P has forwarded $h(X)$ without knowing about its structure. One possibility is to introduce a notation for forwarded messages as it was done in [1]. But this implies that we have to decide during the idealization process which message is *expected* to be forwarded, thus excluding the possibility that an intruder might transmit a stolen hash value without knowing the content. So we have to suppose that P must have computed $h(X)$ himself:

SA3 $\quad P\, said\, h(X) \wedge \neg P\, sees\, h(X) \longrightarrow P\, said\, X$

SA4 $\quad P\, says\, h(X) \wedge \neg P\, sees\, h(X) \longrightarrow P\, says\, X$

Authentication and Key Confirmation. There is a general problem in using secret keys: If P sharing a secret key K with Q receives a cryptogram $\{M\}_K$ he has to exclude himself as originator in order to protect against a reflexion attack. AT and SvO suggest to use a special notation naming the originator. The disadvantage is that this notation was included in the idealization process and that it has no counterpart in the message. Setting this field leaves open the question of how P can exclude the case that he has encrypted M himself. We think that it is preferable to set out all assumptions about the protocol explicitely and to use no more notation than necessary. Therefore we choose the following authentication rules:

A1 $\quad R\, sees\, F(K, X) \wedge P \overset{K}{\leftrightarrow} Q \wedge \neg P\, said\, F(K, X) \longrightarrow Q\, said\, (K, X)$

A2 $\quad R\, sees\, F(K^{-}, X) \wedge \sigma \overset{K}{\mapsto} Q \longrightarrow Q\, said\, (K^{-}, X)$

Comprehension. In order to compute the comprehended submessages of a received message we have to compute the localized message:

C $P\, sees\, M \wedge M_P \equiv Y \longrightarrow P\, believes\, P\, sees\, Y$

We assume that the formats of any sent lists are available and that it is sufficient to recognize any component in order to find the correct format:

C1 $P\, recognizes\, X_i \longrightarrow (X_1, ..., X_n)_P \equiv ((X_1)_P, ..., (X_n)_P)$

Decrypting a cryptogram and recognizing the deciphered text is enough evidence to comprehend the structure of a cryptogram:

C2 $P\, recognizes\, X \wedge P\, has\, K^{-1} \longrightarrow (enc(K, X))_P \equiv enc(K, X_P)$

Whoever knows M is able to verify the one-way computation $H(M)$:

C3 $P\, has\, M \longrightarrow H(M)_P \equiv H(M_P)$

C4 $P\, has\, K_q^- \wedge P\, has\, K_r^+ \longrightarrow \alpha(\{K_q, K_r\})_P \equiv \alpha(\{K_q, K_r\})$

Signatures without message recovery can be verified by using the contents and the corresponding public key:

C5 $P\, has\, (K^+, M) \longrightarrow \sigma(K^-, M)_P \equiv \sigma(K^-, M_P)$

Equivalences. The following group of axioms allows to compute the comprehended submessages of any seen message:

E1 $X \equiv X$

E2 $X \equiv Y \wedge Y \equiv Z \longrightarrow X \equiv Z$

E3 $X \equiv Y \longrightarrow F(X) \equiv F(Y)$

E4 $X_1 \equiv Y_1 \wedge ... \wedge X_n \equiv Y_n \longrightarrow (X_1, ..., X_n) \equiv (Y_1, ..., Y_n)$

Key Derivation.
Symmetry property of shared keys:

S $P \overset{K}{\leftrightarrow} Q \longrightarrow Q \overset{K}{\leftrightarrow} P$

Applying a shared key K and a suitable key derivation parameter M for the key derivation function F yields a new good shared key $F(K, M)$:

KD $P \overset{K}{\leftrightarrow} Q \wedge good_F(M) \longrightarrow P \xleftrightarrow{F(K,M)} Q$

Key agreement:

KA $\alpha \overset{K_p}{\mapsto} P \wedge \alpha \overset{K_q}{\mapsto} Q \longrightarrow P \xleftrightarrow{\alpha(\{K_p, K_q\})} Q$

We did not try to find a complete axiomatisation because our goal is to find a tool as small as possible for analysing protocols whose soundness can be proven.

5 Proof of Correctness

Theorem 1. *Every derivable formula is valid in $\mathcal{AUT}$.*

It is sufficient to show that all axioms are valid in $\mathcal{AUT}$ and that the rules transfer valid formulae into valid formulae. A formula is called to be *valid* in $\mathcal{AUT}$ iff it is valid in every world of every structure in $\mathcal{AUT}$. Therefore we fix some $(\mathcal{W}, (\mathcal{W}_P)_{P \in \mathcal{P}}) \in \mathcal{AUT}$ and $(r,k) \in \mathcal{W}$.

Most axioms follow directly from the definitions in section 3. Soundness of **MP**, **M**, **K**, **4**, and **5** follows easily from the properties of $\mathcal{AUT}$.

Soundness of **H1** can be proven by an easy induction on the structure of $seensub_{S_P^{(r,k)}}(X)$. A proof of **A1** and **A2** is similar to the proof in [9, p.8].
The most interesting innovation is Axiom

C $\quad P\, sees\, M \wedge M_P \equiv X \longrightarrow P\, believes\, P\, sees\, X$

In order to prove its soundness we need the following two lemmas:

Lemma 2. *Let X and X' be arbitrary generalized messages satisfying $sight_P^{(r,k)}(X) = sight_P^{(r',k')}(X')$ and S resp. S' be abbreviations for the knowledge bases $S_P^{(r,k)}$ resp. $S_P^{(r',k')}$. For all submessages $M \in \mathrm{seensub}_S(X)$ there exists a $M' \in \mathrm{seensub}_{S'}(X')$ satisfying $sight_P^{(r',k')}(M') = sight_P^{(r,k)}(M)$.*

Lemma 3. *Let $(r,k), (r',k') \in \mathcal{W}$ be two points, $M \in \mathcal{M}$ a message and $X \in \mathcal{M}_\mathcal{P}$ a generalized message:*

$$sight^{(r',k')}(M_P) = sight^{(r,k)}(X) \quad \Rightarrow \quad M \preceq X.$$

Proof (Soundness of Axiom **C***).* Suppose $(r,k) \models P\, sees\, M \wedge M_P \equiv X$. By definition there exists a time $k^* \leq k$ so that $H^{(r,k^*)} = receive_P(N)$ and $M \in seensub_{S_P^{(r,k)}}(N)$. Let (r',k') be an arbitrary point satisfying $(r,k) \sim_P (r',k')$. We have to show that $(r',k') \models P\, sees\, X$. By definition there exists an action $H^{(r',k'')}$ so that

$$sight_P^{(r',k')}(H^{(r',k'')}) \;=\; sight_P^{(r,k)}(H^{(r,k^*)}) \;=\; receive_P(sight_P^{(r,k)}(N)).$$

Therefore $H^{(r',k'')} = receive_P(N')$ and $sight_P^{(r,k)}(N) = sight_P^{(r',k')}(N')$. Lemma 2 shows the existence of $M' \in seensub_{S_P^{(r',k')}}(N')$ satisfying $sight_P^{(r',k')}(M') = sight_P^{(r,k)}(M)$. In order to prove $(r',k') \models P\, sees\, X$ we have to show the existence of an instance $M'' \preceq X$, so that $(r',k') \models P\, sees\, M''$. By the assumption $(r,k) \models M_P \equiv X$ it is $sight^{(r,k)}(X) = sight^{(r,k)}(M_P) = sight_P^{(r,k)}(M) = sight_P^{(r,k)}(M')$. By Lemma 3 we can conclude $M' \preceq X$ and because of $(r',k') \models P\, sees\, M'$ this completes the proof. □

Proof (Lemma 2). The proof is an induction on the structure of $\text{seensub}_S(X)$. We restrict to the case of a ciphertext $C = enc(K, M) \in \text{seensub}_S(X)$ satisfying the lemma, $K^{-1} \in \overline{S}$ and, thus, $M \in \text{seensub}_S(X)$. We have to show that also M satisfies the lemma.

Because C satisfies the lemma, there exists $C' \in \text{seensub}_{S'}(X')$ such that $sight_P^{(r',k')}(C') = sight_P^{(r,k)}(C)$. First we consider the case that $sight_P^{(r,k)}(M) \neq *$. Thus we get

$$sight_P^{(r',k')}(C') \;=\; sight_P^{(r,k)}(enc(K,M)) \;=\; enc(K, sight_P^{(r,k)}(M)).$$

Thus C' must be a cryptogram $C' = enc(K', M')$ satisfying

$$sight_P^{(r',k')}(C') = enc(K', sight_P^{(r',k')}(M')).$$

It follows $K'^{-1} \in \overline{S'}$ and thus $M' \in \text{seensub}_{S'}(X')$. $sight_P^{(r',k')}(M') = sight_P^{(r,k)}(M)$ completes the case.
Second, let $sight_P^{(r,k)}(M) = *$. Then $M' = C'$ does the job because $sight_P^{(r,k)}(enc(K,M)) = * = sight_P^{(r,k)}(C')$. □

Lemma 3 can be proved by induction on the complete structure of X by considering the cases $X = Y_P$, $X \in \mathcal{M}$, $X = (X_1, ..., X_n)$, and $X = F(Y)$.

6 Detection of Invalid Axioms

The formal semantics enables us to detect invalid axioms of other logics. It confirms that the axiom A11 of [1], is invalid as already noted by Abadi-Tuttle themselves (cf. [9]). The original message-meaning-rule of BAN [2] is also invalid if it is interpreted according to our semantics.

A further interesting example concerns the recognizability operator of GNY-logic [6]. Because GNY do not give a formal semantics we can only examine whether their axioms satisfy our given semantics. It is easy to see that [4]

$R6 \qquad P\,has\,H(M) \longrightarrow P\,recognizes\,M$

is not valid with respect to our semantical interpretation because $H(M) \in \overline{S_P^{(r,k)}}$ does not imply $sight_P^{(r,k)}(M) \neq *$. At this point it is not clear if the rule is incorrect or if our semantics is inappropiate.

Iterative application of the GNY-axioms $P1: \quad P\,sees\,M \longrightarrow P\,has\,M$, $P4: \quad P\,has\,M \longrightarrow P\,has\,H(M)$ and $R6$ yields that $P\,sees\,M$ will always imply $P\,recognizes\,M$. This implication does not fit to our intuitive meaning of recognition. Therefore we argue that R6 should be omitted. (The issue of GNY-rule R6 was also discussed in [9, 10].)

[4] The GNY-Expressions $P \ni M$ and $P \models \phi(M)$ are substituted by $P\,has\,M$ and $P\,recognizes\,M$ respectively.

We note that some axioms of a previous version of AUTLOG [7] are invalid as well, for example the key confirmation rule

$$K1 \quad P\,sees\,\{M\}_K \wedge P\,believes\,P\overset{K}{\leftrightarrow}Q \wedge P\,believes\,Q\,says\,M$$

$$\longrightarrow P\,believes\,Q\,says\,P\overset{K}{\leftrightarrow}Q$$

The point is that Q might have sent the cryptogram $\{M\}_K$ in a previous run. Imagine that in the meantime Q has lost confidence in key K but he has again sent the message M in the current run. Then the premises are fulfilled but the conclusion does not hold.

7 AUTLOG

The calculus was designed in order to be implemented in the Siemens tool AUTLOG which is written in PROLOG. Of course, a formal analysis using a BAN-like logic can still be made by hand within a reasonable amount of time. The main advantage of using an automatized tool like AUTLOG is the correctness of the analysis. Human beings tend to mixture syntactical and semantical reasoning. Therefore one risk of hand-made formal deductions is that rules are applied which are not explicitly stated in the calculus (e.g. [4, p.10]).

Since our logic is designed for automated derivations we have to be more precise about the functions than the SvO-logic is, e.g. the rule 10 of [10] is split up into two rules H2 and H3. Therefore the number of rules of our logic increases the number of SvO-rules.

We note that the new calculus compared to [7] significantly increases the speed of the automated derivations. PROLOG tries to satisfy the goal by looking for candidates using a backward search. Our new calculus has reduced the number of possible candidates.

8 Example

The following protocol was designed as an authentication protocol between network operator, N, and user, U, in a mobile net like UMTS. We follow the description in [8] in order to demonstrate how the prerequisites, the transactions, and the goals are expressed in the formal language.

8.1 Transactions

1. User U generates a random number t and computes his public key agreement key $Ku^+ = g^t$ which he sends to N:

$$U \longrightarrow N : \quad g^t$$

2. At this stage N does not know the identity of the sender of the first message. N computes the agreed key $\alpha(\{Kn, Ku\}) = (g^t)^s$. He then generates a random number r which he uses to compute a fresh shared secret key

$K = h1((g^t)^s, r) = h1(\alpha(\{Kn, Ku\}), r)$ by applying the one-way function $h1$. N confirms the possession of K by applying the hash function $h2$. If required N sends encrypted *data* to U.

$$N \longrightarrow U: \quad r,\ h2(K),\ \{data\}_K$$

3. Since U knows N's public key agreement key g^s he can compute the key agreement key $\alpha(Ku^-, Kn^+) = (g^s)^t$ and the derived key $K = h1((g^s)^t, r)$. He is thus able to check $h2(K)$ and to read *data*. He now signs the hash value $h3(K, data)$ and thus confirms the possession of K. (Since the field *data* is optional we have to choose $h3 \neq h2$ in order to avoid a simple reflection of $h2(K)$). The possession of K confirms that U has indeed chosen g^t. U encrypts his identitiy $IMUI$ in order to ensure anonymity.

$$U \longrightarrow N: \{IMUI\}_K, \{\sigma(KU^-; h3(K, data))\}_K$$

4. N decrypts the ciphertexts, gets to know $IMUI$, and can now verify the signature.

8.2 Prerequisites

U generates a random number t which he believes to be fresh, i.e., not used in a run before. He chooses the pair $Ku := (Ku^-, Ku^+) := (t, g^t)$ as a temporary key agreement key.

$$U\ has\ Ku^-\ (U1) \quad U\ believes\ fresh(Ku^-)\ (U2) \quad U\ believes\ \alpha \overset{Ku}{\mapsto} U\ (U3)$$

U has a copy of the public long-term key agreement key $Kn^+ := g^s$ of N and believes that this is the right key

$$U\ has\ Kn^+ \quad (U4) \qquad U\ believes\ \alpha \overset{Kn}{\mapsto} N \quad (U5)$$

U is able to convince himself that r is *good* for key derivation.

$$U\ believes\ good_{h1}(r) \quad (U6)$$

U names the derived key with K. Furthermore he can check that he did not send $h2(K)$ himself.

$$(K)_U \equiv K \quad (U7) \qquad U\ believes\ \neg U\ said\ h2(K) \quad (U8)$$

Of course, N has a copy of his key agreement key and believes that this is the right key.

$$N\ has\ Kn^- \quad (N1) \qquad N\ believes\ \alpha \overset{Kn}{\mapsto} N \quad (N2)$$

N generates a random number which he believes to be fresh and to be good for key derivation using the hash function $h1$.

$$N\ has\ r\ (N3) \quad N\ believes\ fresh(r)\ (N4) \quad N\ believes\ good_{h1}(r)\ (N5)$$

N has got a copy of U's public verification key KU and believes that this is an authentic copy but he first has to learn $IMUI$ before he knows which key he has to take.

$$N\,sees\,IMUI \longrightarrow (N\,has\,KU^{+} \wedge N\,believes\,\sigma \overset{KU}{\mapsto} U) \quad (N6)$$

N believes that U has the jurisdiction to choose his own key agreement keys and N believes that if U says $K = h1(\alpha(\{Ku, Kn\}), r)$, then U believes that Ku is U's key agreement key.

$$N\,believes\,U\,controls\,\alpha \overset{Ku}{\mapsto} U \qquad (N7)$$

$$N\,believes\,\,(U\,says\,K \longrightarrow U\,believes\,\alpha \overset{Ku}{\mapsto} U) \quad (N8)$$

N has and comprehends $data$. He names the derived key by K.

$$N\,has\,data \quad (N9) \qquad N\,recognizes\,data \quad (N10)$$

$$(data)_N = data\,(N11) \qquad (K)_N = K \quad (N12)$$

N believes that the value $h3(K, data)$ was not sent to U, i.e., U has generated this hash value himself. Since $data$ is optional this belief is only justified if $h3(K)$ cannot be computed from the knowledge of $h2(K)$, especially $h2 \neq h3$.

$$N\,believes\,\neg U\,sees\,h3(K, data) \quad (N13)$$

8.3 Goals

According to [8, p2] the protocol is supposed to meet the following goals:

1. Mutual explicit authentication:

$$U\,believes\,N\,says\,X, \quad N\,believes\,U\,says\,X$$

2. Agreement on a shared secret key with mutual implicit key authentication:

$$U\,has\,K,\;\; N\,has\,K,\;\; U\,believes\,U \overset{K}{\leftrightarrow} N,\;\; N\,believes\,N \overset{K}{\leftrightarrow} U$$

3. Mutual key confirmation:

$$U\,believes\,N\,says\,K,\;\; N\,believes\,U\,says\,K$$

4. Mutual assurance of key freshness:

$$U\,believes\,\;fresh(K),\;\; N\,believes\,\;fresh(K)$$

5. Non-repudation by U of data sent by U to N, i.e., N has obtained a signature by U on $data$ and N believes that U has recently sent the data:[5]

$$N\,sees\,\sigma(KU^{-}; h3(K, data)),\;\; N\,believes\,U\,says\,data$$

6. N knows the identity of U:

$$N\,believes\,N\,sees\,IMUI$$

[5] A compromise of the private signature key KU^{-} is not taken into account

8.4 Proving the goals

We have listed the numbers of the formulas which lead to the new formula. Since the rules MP and K are applied very often we do not always mention it.

The transactions are written as

$$N\ \mathit{sees}\ Ku^{+} \tag{1}$$

$$U\ \mathit{sees}\ (r, h2(K), \{\mathit{data}\}_K) \tag{2}$$

$$N\ \mathit{sees}\ (\{IMUI\}_K,\ \{\sigma(KU^{-}; h3(K, \mathit{data}))\}_K) \tag{3}$$

We start looking at U's state.

$$U1, U4 \xrightarrow{H4} U\ \mathit{has}\ \alpha(\{Ku, Kn\}) \tag{4}$$

$$2, 4 \xrightarrow{SE1, H1-H3} \boxed{U\ \mathit{has}\ h1(\alpha(\{Ku, Kn\}), r)} \tag{5}$$

$$U3, U5 \xrightarrow{KA} U\ \mathit{believes}\ U \xleftrightarrow{\alpha(\{Ku, Kn\})} N \tag{6}$$

$$U6, 6 \xrightarrow{KD} \boxed{U\ \mathit{believes}\ U \xleftrightarrow{h1(\alpha(\{Ku, Kn\}), r)} N} \tag{7}$$

$$U2 \xrightarrow{F3, F2} \boxed{U\ \mathit{believes}\ \ \mathit{fresh}(h1(\alpha(\{Ku, Kn\}), r))} \tag{8}$$

Setting $K = h1(\alpha(\{Ku, Kn\}), r))$ and $K^{-1} = K$ we get

$$2 \xrightarrow{SE1} U\ \mathit{sees}\ h2(K) \tag{9}$$

$$5 \xrightarrow{C3} (h2(K)_U) \equiv h2(K_U) \tag{10}$$

$$U7 \xrightarrow{E3} h2(K_U) \equiv h2(K) \tag{11}$$

$$10, 11 \xrightarrow{E2} (h2(K))_U \equiv h2(K) \tag{12}$$

$$9, 12 \xrightarrow{C} U\ \mathit{believes}\ U\ \mathit{sees}\ h2(K) \tag{13}$$

$$U8, 7, 13 \xrightarrow{A1, K} U\ \mathit{believes}\ N\ \mathit{said}\ K \tag{14}$$

$$8, 14 \xrightarrow{NV} \boxed{U\ \mathit{believes}\ N\ \mathit{says}\ K} \tag{15}$$

Since the last statement refers to both goal 1 and goal 3 we have proven that the protocol meets the goals concerning U.

For N we can prove the following:

$$1, N1, N3 \xrightarrow{H4, H3} \boxed{N\ \mathit{has}\ h1(\alpha(\{Kn, Ku\}), r)} \tag{16}$$

$$N4 \xrightarrow{F2, F3} \boxed{N\ \mathit{believes}\ \ \mathit{fresh}(h1(\alpha(\{Kn, Ku\}), r))} \tag{17}$$

$$3 \xrightarrow{SE1} N\ \mathit{sees}\ \{IMUI\}_K \tag{18}$$

$$16, 18 \xrightarrow{SE2} \boxed{N\ \mathit{sees}\ IMUI} \tag{19}$$

$$N6, 19 \xrightarrow{MP} N\ \mathit{has}\ KU^{+}\ \wedge\ N\ \mathit{believes}\ \sigma \overset{KU}{\mapsto} U \tag{20}$$

$$3, 16 \xrightarrow{SE1, SE2} \boxed{N\ \mathit{sees}\ \sigma(KU^{-}; h3(K, \mathit{data}))} \tag{21}$$

By iterated applying of the rules for comprehending and equivalence it follows from N9 – N12, 16, and 20

$$\ldots \longrightarrow (h3(K, data))_N \equiv h3(K, data) \tag{22}$$

$$\ldots \longrightarrow \sigma(KU^-; h3(K, data))_N \equiv \sigma(KU^-; h3(K, data)) \tag{23}$$

$$21, 23 \xrightarrow{C} N\, believes\, N\, sees\, \sigma(KU^-; h3(K, data)) \tag{24}$$

$$24, 20 \xrightarrow{A2,K} N\, believes\, U\, said\, h3(K, data) \tag{25}$$

$$17, 25 \xrightarrow{F2,NV} N\, believes\, U\, says\, h3(K, data) \tag{26}$$

$$N13, 26 \xrightarrow{SA4} \boxed{N\, believes\, U\, says\, (K, data)} \tag{27}$$

$$N8, 27 \xrightarrow{K} N\, believes\, U\, believes\, \alpha \overset{Ku}{\mapsto} U \tag{28}$$

$$N7, 28 \xrightarrow{J} N\, believes\, \alpha \overset{Ku}{\mapsto} U \tag{29}$$

$$N2, 29 \xrightarrow{KA,K} N\, believes\, U \xleftrightarrow{\alpha(\{Kn, Ku\})} N \tag{30}$$

$$N5, 30 \xrightarrow{KD,K} \boxed{N\, believes\, U \xleftrightarrow{h1(\alpha(\{Kn, Ku\}), r)} N} \tag{31}$$

We have thus proven that the protocol meets the goals.

9 Conclusion

We present a logic for the analysis of authentication protocols and give a formal semantics which enables us to prove its soundness.

The logic can handle a wide variety of cryptographic mechanisms using a minimum of notation. The use of negative formulae enables us to deal with hash values without introducing an additional notation for forwarded messages and allows a satisfying solution for the symmetry problem.

The elimination of the formulae out of the idealized messages leads to a clear distinction between the protocol itself and the assumptions about it. A careful examination of all initial assumptions gives a much deeper insight to the real outcome of the analysis and allows the detection of a wide variety of protocol flaws. In addition, this distinction is necessary in order to solve the issues of recognizability, computability, and comprehension.

Acknowledgements. The first author thanks Erich Graedel for the supervision of her diploma thesis. The second author thanks Martin Abadi, Paul Syverson, and Pekka Nikander for fruitful comments they gave to him during the participation at the Coding, Cryptography and Computer Security programme at the Newton Institute, Cambridge. Finally, we thank the anonymous referees for suggestions concerning the presentation.

References

1. M. Abadi, M. Tuttle, "A Semantics for a Logic of Authentication," *Proc. of the ACM Symposium of Principles of Distributed Computing*, 1991, 201-216.

2. M. Burrows, M. Abadi, R. Needham, *A Logic of Authentication,* Report 39 Digital Systems Research Center, Pao Alto, California, 1989.
3. Chellas, *Modal Logic,* Cambridge University Press, Cambridge, England, 1980.
4. L. Chen, D. Gollmann, Y. Han, C. Mitchell, ***Formal Verification of a Mutual Authentication Protocol,*** Royal Holloway, University of London 3GS3/IREP/ RHUL/032/A(draft), 1995.
5. R. Fagin, J. Halpern, Y. Moses, M. Vardi, ***Reasoning about knowledge,*** MIT Press, Cambridge, Mass., 1995.
6. L. Gong, R. Needham, R. Yahalom, "Reasoning about Belief in Cryptographic Protocols," ***Proc. of the 1990 IEEE Symp. on Research in Security and Privacy,*** 234-248.
7. V. Kessler, G. Wedel, "AUTLOG - An Advanced Logic of Authentication," ***Proc. of the Computer Security Foundations Workshop VII,*** Franconia, IEEE Computer Society Press 1994, 90-99.
8. ETSI SMG/SG/TD 73/95 ***Protocols for UMTS Providing Mutual Authentication and Key Establishment Using Asymmetric Techniques.***
9. P. Syverson, P. van Oorschot, "On Unifying Some Cryptographic Protocol Logics", *Proc. of the IEEE Computer Society Symp. on Security and Privacy 1994,* 14-28.
10. P. Syverson, P. van Oorschot, *A Unified Cryptographic Protocol Logic,* Unpublished preprint, March 1996.
11. G. Wedel, *Formale Semantik für Authentifikationslogiken,* Diplomarbeit FB Mathematik der RWTH Aachen, Nov. 1995.