AUTLOG – An advanced logic of authentication

Volker Kessler

Siemens AG
ZFE ST SN 3
D-81730 Munich, Germany
Volker.Kessler@zfe.siemens.de

Gabriele Wedel

RWTH Aachen
Math. Grundlagen der Informatik
Ahornstr. 55
D-52074 Aachen, Germany

Abstract

We present a modified version of the BAN logic which is implemented in PROLOG. The modifications are motivated by the analysis of a lot of protocols. In the paper we analyze a challenge-response protocol and its dual version in order to show the advantages of the modified logic. The analysis shows an interesting difference between two protocols which seem to be very similar. Finally, we discuss the inability of the logic to handle parallel protocol runs.

1 Introduction

Burrows, Abadi and Needham's paper [3] gave a start for developing new methods for analyzing cryptographic protocols. Since then there came up a lot of criticism of the BAN logic, which is partly justified. Along with this criticism there were a lot of improvements. Surprisingly, every new paper refers to the original paper and not to one of its modifications. The situation now is that there are a lot of different modifications of the BAN logic, but everybody goes back to the original.

This is, probably, because the original has a clear concept. It is easy to understand and easy to apply. Often, the modified logic is able to handle some more features but is much more complicated, cf. Needham's own judgement [11, p7] on the so-called GNY logic [5]:

> One approach is to try to construct a logic which will capture every possible aspect of the subject. The GNY effort was a bit like that, and it entails the erection of a vast conceptual apparatus for rather little advance in coverage. ... It may be that the BAN logic gets close enough to complete analysis for common sense to be a reasonable guide to understanding what is left.

Anderson made the same experience while analyzing UEPS [2, p417] :

> Although more elaborate systems (like GNY) exist, a first validation should be carried out with BAN, as it is easy to do, and failure to establish a desired result will indicate either a bug in the protocol, or something which BAN cannot express. It will normally be obvious what to do next.

On the other hand the example UEPS demonstrates that there are protocols which the original BAN logic cannot handle. Certain protocols demand for new inference rules. Therefore it is necessary to change the BAN logic in some aspects. It was our aim to expand the BAN logic while keeping the easy handling. The analysis of about 30 different cryptographic protocols led us to the following modifications which are useful for most of the protocols and do not complicate the logic:

1. introducing the predicate "recognize",
2. replacing the predicate "A believes that B *believes* that ..." by the predicate "A believes that B *has recently said* that ...",
3. enlarging the meaning of the operator "see",
4. not omitting the cleartext messages in the idealized protocol,
5. simulating an eavesdropper Z, and
6. reducing the ambiguous idealiziation step by introducing a special *key rule*.

We implemented a tool, called AUTLOG, in PROLOG in order to automize the deductions.

First we will explain the modifications of the logic and the features of AUTLOG. Then we demonstrate the advantages of these modifications by analyzing a simple challenge-response protocol from the Draft International Standard [8]. Especially, we show a failure in this protocol.

Finally, by analyzing a protocol of [7] we found a further limitation of the logic of authentication which lies in the inability to handle parallel runs of protocols.

The formal model of the advanced logic of authentication is described in appendix A.

In the following we assume that the reader is familiar with the notations from [3]. They are also summed up in appendix A.

2 Modifications to the BAN Logic

2.1 Recognizability

The original message-meaning rule (BAN 1) is as follows

$$\frac{P \models P \overset{K}{\leftrightarrow} Q,\ P \triangleleft \{X\}_K}{P \models Q \models X}$$

But this rule does not reflect our intuitive meaning of authentication. Imagine P receives a random string, deciphers it with some key K known to him, and then gets a random looking string he never saw before. This does not tell him anything! He cannot deduce that the string was enciphered with key K unless the deciphering leads to something which makes sense to him. GNY [5] already introduced a predicate "P believes to recognize X" which we denote by

$$P \models \rho(X)$$

They used it as a further condition for the message-meaning rule and thus could demonstrate the incompleteness of the fourth message of the (original) Needham-Schroeder protocol. The concept of recognizability is too important to be dropped.

Furthermore, we can conclude under these conditions that P believes that Q once used the key K and once said the ciphertext $\{X\}_K$. Summarizing we get the following *authentication rule* which is somewhat between (BAN 1) and the rather complicated rule (I1) of [5]:

$$\textbf{A1}\quad \frac{P \triangleleft \{X\}_K,\ P \models P \overset{K}{\leftrightarrow} Q,\ P \models \rho(X)}{P \models Q \mid\sim X,\ P \models Q \mid\sim K,\ P \models Q \mid\sim \{X\}_K}$$

2.2 The "Has recently said"–Operator

The nonce-verification rule (BAN 4) of BAN states

$$\frac{P \models Q \mid\sim X,\ P \models \sharp X}{P \models Q \models X}$$

It is the only postulate from $\mid\sim$ to $\models$. BAN make the following (hidden) assumption for this rule [3, p7]

> For sake of simplicity, X must be "cleartext", that is it should not include any subformula of the form $\{Y\}_K$. (When this restriction is not met, we can conclude only that Q has recently said X. We might introduce a "has recently said" operator to express this conclusion, should the need arise in an example.)

This is exactly what we do. But we are going a step further and replace *every* formula

$$P \models Q \models X$$

by the formula

$$P \models Q \mid\approx X$$

(P believes that Q *has recently said* X.)

We do not use the formula "P believes that Q believes X" because P can only guess what Q believes. P might conclude that Q has recently said X but in order to conclude more one needs to assume honesty as stated in [3, p5]:

> ...furthermore, we assume that when principal P says X then he actually believes X.

We do not use this assumption because we do not need statements on beliefs of other's beliefs.

In this way we can apply the nonce-verification rule to ciphertexts as well.

$$\textbf{NV}\quad \frac{P \models Q \mid\sim X,\ P \models \sharp X}{P \models Q \mid\approx X}$$

We simply change the jurisdiction rule to

$$\textbf{J}\quad \frac{P \models Q \mid\Rightarrow X,\ P \models Q \mid\approx X}{P \models X}$$

This modification is justified because Q can convince P of the truth of X only by saying that X is true (see [1, p206f] for a detailed reasoning).

Thus the strong authentication goal

$$A \models B \models A \overset{K}{\leftrightarrow} B, \qquad B \models A \models A \overset{K}{\leftrightarrow} B.$$

is changed to

$$A \models B \mid\approx A \overset{K}{\leftrightarrow} B, \qquad B \models A \mid\approx A \overset{K}{\leftrightarrow} B.$$

2.3 The "See"–operator

Usually, a principal decrypts a ciphertext with a key K if he believes that K is the suitable key. But there are situations, in which a principal has first to decrypt a text before he is convinced that the key he uses is a good key, e.g. the Yahalom protocol [3]. GNY solved this problem by introducing the operator "possess":

> You "possess" anything you can get by decrypting or encrypting something you've seen with something else you "possess".

On the other hand it is desirable to have as few operators as possible. The difference in meaning between "see" and "possess" does not influence the results of logical analysis. Therefore we combine these two operators by introducing further rules for the operator "see" like:

$$\textbf{BS}\quad \frac{P \models X}{P \lhd X}$$

If P believes something he possesses resp. sees it.

$$\textbf{S2}\quad \frac{P \lhd \{X\}_K,\ P \lhd Q \overset{K}{\leftrightarrow} R}{P \lhd X}$$

If P sees a text enciphered with key K and he was told that K is a key for somebody he is able to decrypt the ciphertext and to see the cleartext X. Note that it is not necessary to believe that K is a key at all. You just try it as a key and then you see the result. The idea behind this rule is that a good cryptoalgorithm is resistant against an exhaustive key search because of a large set of keys. Thus keys cannot be guessed. It is only possible to try those bitstrings where you have a special hint that they could be a key.

Of course, this rule is applicable in case that P believes $P \overset{K}{\leftrightarrow} Q$ because then rule **BS** leads especially to $P \lhd P \overset{K}{\leftrightarrow} Q$. But the rule **S2** also enables to use speculative keys, i.e. bitstrings which should be a key according to some message but for which you have not yet got any good reason to believe that they are really keys, cf. Yahalom's protocol in [3].

2.4 The Eavesdropper Z

There has been a great debate, and is still going on, about the question whether BAN logic analyzed protocols are "secure". Nessett [12] gave an example of an obviously insecure protocol, but which achieves the authentication goal according to the BAN logic.

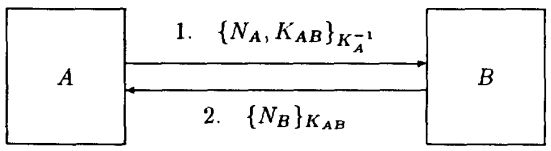

Figure 1: Nessett's counterexample

The key K_{AB} is (RSA-)signed with the private key of A so that everybody can learn the secret key K_{AB} by applying A's public key. In their rejoinder to Nessett BAN [4] pointed out that they did not intend to deal with security flaws and that this protocol did not satisfy the assumptions of the BAN logic. The fact that A publishes the key K_{AB} contradicts his own belief that K_{AB} is a good key for him to communicate with B. It is assumed in the BAN logic that the principals keep their secrets secret.

BAN's rejoinder is correct, of course, but not really satisfying. If one has proved the functionality of a protocol one would also like to prove something about the security. We suggest an easy way out of this dilemma and prove that the protocol is not vulnerable against passive attacks, at least.

In AUTLOG we implemented a passive intruder Z. She is supposed to see the transmitted messages, to know the public keys, and to be able to decrypt ciphertext if she has seen the right key before. We do not make any statements about her belief, because it does not matter if she really believes that K is the secret key between A and B. She just tries to decrypt any ciphertext she sees with every key she has seen before. Therefore we only need for Z the "seeing"-rules **S1**–**S5** (cf. appendix A). Let $\mathcal{S}_Z \subset \mathcal{M}_\mathcal{T}$ be the set of the messages Z sees which is closed corresponding to the following conditions:

$\mathcal{Z}$1 For each transmitted message Y of the protocol $Z \lhd Y$ is in $\mathcal{S}_Z$

$\mathcal{Z}$2 For all public keys $\overset{K}{\mapsto} P$ $Z \lhd \overset{K}{\mapsto} P$ is in $\mathcal{S}_Z$

$\mathcal{Z}$3 If the conditions of a seeing-rule are in $\mathcal{S}_Z$ then the conclusion are in $\mathcal{S}_Z$ as well.

Analyzing a protocol one has to determine $\mathcal{S}_Z$, especially it is important to know if $Z \lhd A \overset{K_{AB}}{\leftrightarrow} B$ is an element of $\mathcal{S}_Z$ or if not. Unfortunately, the logic has no negation and it is not complete, e.g., it is not possible to conclude that user A does *not* believe that K is a suitable key. One can only say that it is not

possible to derive the opposite. This is a general problem of the logic of authentication, but regarding the eavesdropper Z we are in a better position because $\mathcal{S}_Z$ can be proved to be finite.

Finiteness of $\mathcal{S}_Z$

Let $\mathcal{S}^0 \subset \mathcal{S}_Z$ be the (finite!) set consisting of all transmitted messages and the public keys of all principals, if asymmetric cryptosystems are used. Then let $\mathcal{S}^{i+1}$ for $i \in \mathcal{N}$ be the set of all conclusions of the "seeing"-rules with the conditions lying in $\mathcal{S}^i$, more exactly define $\mathcal{S}^{i+1}$ by:

$Z \triangleleft X_1, \ldots, Z \triangleleft X_n \in \mathcal{S}^{i+1}$ if
$Z \triangleleft (X_1, \ldots, X_n) \in \mathcal{S}^i$ (Rule **S1**)

$Z \triangleleft X, Z \triangleleft Q \overset{K}{\leftrightarrow} R \in \mathcal{S}^{i+1}$ if
$Z \triangleleft \{X\}_K, Z \triangleleft Q \overset{K}{\leftrightarrow} R \in \mathcal{S}^i$ (Rule **S2**)

$Z \triangleleft X \in \mathcal{S}^{i+1}$ if
$Z \triangleleft \langle X \rangle_S \in \mathcal{S}^i$ (Rule **S3**)

$Z \triangleleft X, Z \triangleleft \overset{K}{\mapsto} Z \in \mathcal{S}^{i+1}$ if
$Z \triangleleft \{X\}_K, Z \triangleleft \overset{K}{\mapsto} Z \in \mathcal{S}^i$ (Rule **S4**)

$Z \triangleleft X, Z \triangleleft \overset{K}{\mapsto} Q \in \mathcal{S}^{i+1}$ if
$Z \triangleleft \{X\}_{K^{-1}}, Z \triangleleft \overset{K}{\mapsto} Q \in \mathcal{S}^i$ (Rule **S5**)

$Z \triangleleft \alpha \in \mathcal{S}^{i+1}$ if
$Z \triangleleft \alpha \in \mathcal{S}^i$ and $Z \triangleleft \alpha$ is not used as a condition of a rule applied in the step from i to $i+1$.

It is clear from the definition that all $\mathcal{S}^n$ are finite and that $\mathcal{S}_Z = \bigcup_{n=0}^{\infty} \mathcal{S}^n$. (Note that a certain list or ciphertext is used only once!) Define

$$\kappa = \begin{cases} \min\{n \in \mathcal{N} | \mathcal{S}^n = \mathcal{S}^{n+1}\} & \text{if } \exists n\ \mathcal{S}^n = \mathcal{S}^{n+1} \\ \infty & \text{if } \forall n\ \mathcal{S}^n \neq \mathcal{S}^{n+1} \end{cases}$$

If $\kappa < \infty$ then we have $\mathcal{S}^i = \mathcal{S}^{i+1}$ for all $i \geq \kappa$ and thus $\mathcal{S}_Z$ is finite.

In appendix A.1 we define the length ℓ of messages to measure their complexity. This enables us to define the *length* of $\mathcal{S}^i$ by

$$\ell(\mathcal{S}^i) = \sum_{\{\alpha | (Z \triangleleft \alpha) \in \mathcal{S}^i\}} \ell(\alpha)$$

Since the transmitted messages have finite length ℓ the length $\ell(\mathcal{S}^0)$ is also finite. Since the "seeing"-rules reduce the length of the formulas, it is straightforward to prove

Lemma 1 *(i)* $\ell(\mathcal{S}^{i+1}) \leq \ell(\mathcal{S}^i)$ *for all* i
(ii) $\ell(\mathcal{S}^{i+1}) = \ell(\mathcal{S}^i) \iff \mathcal{S}^{i+1} = \mathcal{S}^i$ *for all* i
(iii) $\kappa \leq \ell(\mathcal{S}^0)$

Thus $\mathcal{S}_Z$ is proved to be finite.[1]

AUTLOG derives automatically everything which Z can see starting from the initial axioms $\mathcal{S}^0$. A protocol is regarded as *secure* if Z does not learn any secret keys or other secrets. Especially, AUTLOG realizes that Nessett's protocol is insecure because Z can see the secret key K_{AB}, more exactly $Z \triangleleft A \overset{K_{AB}}{\leftrightarrow} B \in \mathcal{S}^2$.

2.5 Cleartext Messages

In the BAN logic cleartext communication is omitted in the idealized protocol "because it can be forged, and so its contribution to an authentication protocol is mostly one of providing hints as to what might be placed in encrypted messages" [3, p9]. This statement may be correct for the protocols of [3]. Nevertheless we decided not to omit cleartext messages in the idealization because of two reasons:

1. There are protocols in which the authentication is done by sending a *cleartext* message, e.g. see below the protocol of figure 2. In those cases omitting the cleartext message would prevent proving any authentication goal at all.
2. Of course, cleartext messages could be very valuable for an intruder. Therefore we have to pay attention to cleartext messages in order to derive all the information Z can see.

2.6 Key Rule

The idealization process in the BAN logic is probably the most critical step in the whole formal analysis and there has been a lot of criticsm of it because of its vagueness. Of course, the idealization step is necessary to represent the protocol in a logical language, and the step from real world to a formal model *cannot be formal itself*[2]. Therefore we cannot omit the idealization but we can eliminate some parts of it. Look, for example, at the analysis of the Needham-Schroeder

[1] Note that the corresponding set $\mathcal{B}_A$ of all beliefs of A is not finite, because, for example, the freshness rule

F2 $\dfrac{P \models \sharp X, P \triangleleft P \overset{K}{\leftrightarrow} Q}{P \models \sharp \{X\}_K}$

blows up the length of the formulas and leads to infinitely many beliefs of the kind

$$A \models \sharp\{X\}_K, A \models \sharp\{\{X\}_K\}_K, A \models \sharp\{\{\{X\}_K\}_K\}_K, \ldots$$

[2] Thus the criticsm in [13, p30] "that the process of formalizing the requirements is itself not formal" is not justified.

protocol in [3]. In the fourth message B encrypts the challenge N_B with the key K_{AB}. This is idealized as

$$A \lhd \{N_B, A \overset{K_{AB}}{\leftrightarrow} B\}_{K_{AB}}$$

The inclusion of $A \overset{K_{AB}}{\leftrightarrow} B$ in the idealized message enables to conclude

$$A \models B \models\!\sim A \overset{K_{AB}}{\leftrightarrow} B$$

This inclusion is justified because B tells A implicitly by using K_{AB} as enciphering key that he believes K_{AB} to be a good key for communication between A and B. The idealization is used to make this point explicit. This can be avoided by introducing a key rule

$$\textbf{K1} \quad \frac{P \lhd \{X\}_K,\ P \models P \overset{K}{\leftrightarrow} Q,\ P \models Q \models\!\sim X}{P \models Q \models\!\sim P \overset{K}{\leftrightarrow} Q}$$

If P sees the ciphertext $\{X\}_K$ and P believes that K is a good key to communicate with Q and that Q has recently said X, then P believes that Q has recently used K as a shared key for communication with P.

Thus we have transferred a part from the idealization to the formal deduction.

3 Analysis of Challenge–Response Protocols

The advantages of our modifications are best shown by analyzing some protocols.

3.1 An ISO 10181 Protocol

Look at the following challenge-response protocol with encrypted challenge which is recommended in [8, 8.1.5.3 Subclass 4c]:

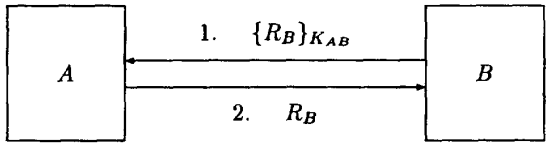

Figure 2: Challenge-response protocol with encrypted challenge

B sends A a challenge R_B which is encrypted by K_{AB} which is supposed to be the shared key between A and B. Thus A is the only one who is able to decrypt the message and he authenticates himself by sending back R_B.

Cleartexts

The authentication is made by the second message which is simply a cleartext. If cleartexts were omitted, as it is done in [3], the most important transaction would be ignored and the authentication goal could not be achieved according to the formal analysis.

Idealization

In most protocols challenges are used in order to guarantee freshness. In this protocol the challenge is also used as a secret between A and B which A uses to authenticate himself. Thus this protocol is idealized as follows:

1. $A \lhd \{A \overset{R_B}{\rightleftharpoons} B\}_{K_{AB}}$
2. $B \lhd \langle R_B \rangle_{R_B}$

We need two initial assumptions

3. $B \models \sharp R_B$
4. $B \models A \overset{R_B}{\rightleftharpoons} B$

Applying the authentication rule $A4$ to (2) and (4) yields

5. $B \models A \mid\!\sim R_B$

and then applying the nonce-verification rule **NV** to (5) and (3) yields the authentication goal

6. $B \models A \models\!\sim R_B$

Note that the first message is not needed for the deduction of the authentication goal. It would be the same as if A had acted spontaneously. The deduction of the authentication goal would not change even if B sent the message $A \overset{R_B}{\rightleftharpoons} B$ without encryption. But, of course, in this case the protocol would be insecure because everybody would learn the secret R_B. The BAN logic would not find any difference between these two variants. But in AUTLOG the failure is noticed because Z could see a secret which is not allowed.

Note that one cannot deduce the following formula

$$A \models B \mid\!\sim A \overset{R_B}{\rightleftharpoons} B$$

because A does not recognize $A \overset{R_B}{\rightleftharpoons} B$.

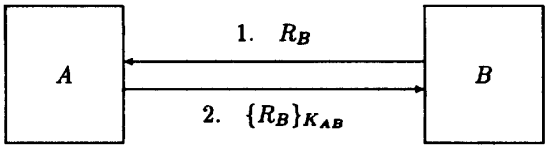

Figure 3: Challenge-response with encrypted response

The disadvantage

The weakness or disadvantage of this protocol becomes clear by comparing this protocol with the usual challenge-response protocol with encrypted response, see figure 3. This variant is recommended in [7, 5.1.2] and in [8, 8.1.5.2 Subclass 4b][3]. It is idealized as follows

1. $A \lhd R_B$
2. $B \lhd \{R_B\}_{K_{AB}}$

and the following assumptions are necessary

3. $B \models \sharp R_B$
4. $B \models \rho(R_B)$
5. $B \models A \overset{K_{AB}}{\leftrightarrow} B$

The authentication goal

$$B \models A \mid\!\sim R_B$$

is achieved by first applying the authentication rule **A1** to (2), (4), and (5) and then applying the nonce-verification rule **NV**.

At first glance both variants seem to be very similar. In [8] it is only remarked that the first variant cannot be used for data origin authentication. But the formal analysis makes clear that there are less requirements on the random number R_B in the second variant than in the first one because it is not supposed to be a shared secret between A and B.

Assume that an intruder Z is able to forecast the challenge R_B. (This is possible if a bad pseudo random number generator like the linear congruential generator is used and the intruder has eavesdropped the last outputs which have been sent in cleartext.) Using the first protocol he can then masquerade himself as A by sending R_B to B without decrypting $\{R_B\}_{K_{AB}}$. But in the second protocol this information does not help Z because he has to reply with the ciphertext $\{R_B\}_{K_{AB}}$. In the second protocol the challenge is only used for freshness. It suffices that the challenge is new but, in general, it needs not to be unpredictable[4]. The first protocol requires unpredictable random numbers and thus makes for more demands on the pseudo random number generator than the second protocol. This difference should be noted and the first protocol should be recommended only with an additional remark that the pseudo random number generator must be cryptographically secure.

Note that symmetric enciphering can preserve confidentiality and authenticity. Since in the first protocol only confidentiality has to be protected one could use asymmetric enciperhing instead. In the second protocol the authenticity could alternatively be protected by either a signature or by a message authentication code.

3.2 Parallel Protocol Runs

Although the logic of authentication is a good tool for analyzing cryptographic protocols this method has some limitations. One limitation is the inability to handle parallel runs of a protocol. This again can be demonstrated by the protocol shown in figure 3.

The response in this protocol must contain the name of receiver B in order to prevent a so-called *reflection attack*, see figure 4. In this attack an intruder Z tries to masquerade himself as A and B challenges Z to authenticate himself. Z initiates a second run of the authentication protocol by reflecting this challenge R_B to B. B authenticates himself by enciphering R_B. Z reflects this ciphertext to B as if he had created this response.

But in our analysis the authentication goal was achieved without the integration of the parameter B. The problem lies in the assumed freshness of the challenge R_B. According to [3] a formula X is *fresh*, if "X has not been sent in a message at any time before the current run of the protocol".

Now look again at the reflection attack shown in figure 4. At the time B sends the first message, the

[3] This is not exactly the recommended protocol because the name of the receiver is missing in the second message, cf. subsection 3.2

[4] Using the second protocol the unpredictability is important under the following conditions: Imagine that an active intruder Z is able to get ciphertexts $\{R\}_{K_{AB}}$ from A *before* he receives the challenge from B. If Z forecasts the challenge R_B he can get the right answer $\{R_B\}_{K_{AB}}$ from A in advance, i.e., in this case the challenge should also be unpredictable. If an intruder can even get ciphertexts from A in the meantime after receiving the challenge and before he has to reply (this happens in the so-called *mafia attack*), the unpredictability does not help anyway.

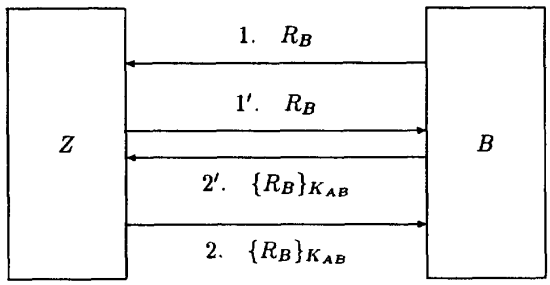

Figure 4: Reflection attack

challenge was not used before, i.e., the assumption

3. $B \models \sharp R_B$

is correct at this point of the protocol. But then R_B is used in another run of the protocol, where B has to authenticate himself. Therefore the assumption (3) is not correct any more at the end of the protocol.

The GNY logic [5] tries to improve the notion of freshness by the concept of a "parser". The parser has to check whether a message of a protocol has been sent before. But the parser controls only the current run of the protocol. The problem arises because there are two parallel runs of the protocol which is possible in some applications. Therefore the GNY logic does not find this failure either.

There does not seem to be any possibility to manage this problem within the logic of authentication.

A The Formal Model

A.1 Syntax

The concept of the syntax is partly based on [1].

The set $\mathcal{T}$ of *atomic terms* simply consists of bit-strings. An atomic term can be a name of a principal, a nonce, a key, or some other data which is not divided into smaller parts. We then define the set $\mathcal{M}_\mathcal{T}$ of *messages* and the subset $\mathcal{F}_\mathcal{T} \subset \mathcal{M}_\mathcal{T}$ of *formulas* by mutual induction. A formula is a message to which a truth value can be assigned. We also define a complexity measure ℓ on the set $\mathcal{M}_\mathcal{T}$.

The language $\mathcal{M}_\mathcal{T}$ is the smallest language over $\mathcal{T}$ satisfying the following conditions:

$\mathcal{M}1$ Each atomic term X is a message with $\ell(X) = 0$.

$\mathcal{M}2$ Each formula is a message.

$\mathcal{M}3$ If $X_1, \ldots, X_n$ are messages then the *list* $(X_1, \ldots, X_n)$ is a message with $\ell((X_1, \ldots, X_n)) = \sum_{i=1,\ldots,n} \ell(X_i) + 1$.

$\mathcal{M}4$ If X is a message and K is a key then $\{X\}_K$ ("The enciphering/signing of X with K") is a message with $\ell(\{X\}_K) = \ell(X) + \ell(K) + 1$.

$\mathcal{M}5$ If X is a message and K is a key then $\mathrm{mac}(K, X)$ ("The one-way enciphering of X with K") is a message with $\ell(\mathrm{mac}(K, X)) = \ell(X) + \ell(K) + 1$. (The expression $\mathrm{mac}(K, X)$ is used for one-way enciphering and for message authentication codes as well because they both lead to the same logical implications which are independent of the length of the ciphertext.)

$\mathcal{M}6$ If X is a message and S is a secret then $\langle X \rangle_S$ ("The combination of X with S") is a message with $\ell(\langle X \rangle)_S = \ell(X) + \ell(S) + 1$.

$\mathcal{M}7$ If X is a message and hash is a hash function then $\mathrm{hash}(X)$ ("The hash value of X ") is a message with $\ell(\mathrm{hash}(X)) = \ell(X) + 1$. (From the logical view the hash function is simply a one-way function because reducing the length of the text has no authentication logical consequences.)

The set of keys and the set of secrets are subsets of $\mathcal{M}_\mathcal{T}$. It depends on the cryptographic algorithm whether a message can be used as a key or not. Note that a key or a secret need not to have the length 0, e.g. a message $\{X\}_K$ of length $\ell(\{X\}_K) \geq 1$ could be used as a key [2].

The language $\mathcal{F}_\mathcal{T}$ of formulas is the smallest language over $\mathcal{T}$ satisfying the following conditions

$\mathcal{F}1$ If P is a principal and X is a formula then $P \models X$ ("P *believes* X") and $P \Rightarrow X$ ("P has *jurisdiction* over X") are formulas with $\ell(P \models X) = \ell(P \Rightarrow X) = \ell(X) + 1$.

$\mathcal{F}2$ If P is a principal and X is a message then $P \lhd X$ ("P *sees* X"), $P \mid\sim X$ ("P *once said* (or used) X"), and $P \mid\approx X$ ("P *has recently said* (or used) X") are formulas with $\ell(P \lhd X) = \ell(P \mid\sim X) = \ell(P \mid\approx X) = \ell(X) + 1$.

$\mathcal{F}3$ If P, Q are principals and K is a message then $P \overset{K}{\leftrightarrow} Q$ ("K is a shared key for communication between P and Q") is a formula with $\ell(P \overset{K}{\leftrightarrow} Q) = \ell(K)$.

$\mathcal{F}4$ If P, Q are principals and S is a message then $P \overset{S}{=\!=} Q$ ("S is a secret between P and Q") is a formula with $\ell(P \overset{S}{=\!=} Q) = \ell(S)$.

$\mathcal{F}5$ If P is a principal and K is a text then $\overset{K}{\mapsto} P$ ("K is the public key of P") is a formula with $\ell(\overset{K}{\mapsto} P) = \ell(K)$.

$\mathcal{F}6$ If X is a message then $\sharp X$ ("X is fresh") and $\rho(X)$ ("X is recognizable") are formulas with $\ell(\sharp X) = \ell(\rho(X)) = \ell(X)$.

The notation in $\mathcal{F}4$ and $\mathcal{F}5$ is symmetrical: $P \overset{K}{\leftrightarrow} Q$ and $Q \overset{K}{\leftrightarrow} P$ respectively $P \overset{S}{\rightleftharpoons} Q$ and $Q \overset{S}{\rightleftharpoons} P$ can be used interchangeably.

We are especially interested in the subset $\mathcal{B}_\mathcal{T} \subset \mathcal{F}_\mathcal{T}$ containing all *believing formulas* $P \models \ldots$ and *seeing formulas* $P \lhd \ldots$ The following axioms work on this set, i.e., both the conditions and the conclusions ly in $\mathcal{B}_\mathcal{T}$.

A.2 Axiom Schemata

For practical reasons we divide the rules into different groups. PROLOG tries to apply any given rule and it takes too long time if there are too many rules. Therefore it is better to load only those groups of rules which are needed for a given protocol. The *basic rules* are always loaded and the choice of the other packages depend on the cryptographic mechanisms which are used in the protocol, e.g. symmetric enciphering, asymmetric enciphering, etc.

A.2.1 Basic Rules

Seeing

S1 $\dfrac{P \lhd (X_1, \ldots, X_n)}{P \lhd X_1, \ldots, P \lhd X_n}$

Believing and seeing

BS $\dfrac{P \models X}{P \lhd X}$

Freshness

F1 $\dfrac{P \models \sharp X_1; \ldots; P \models \sharp X_n}{P \models \sharp(X_1, \ldots, X_n)}$

Lists

L1 $\dfrac{P \models Q \mid\sim (X_1, \ldots, X_n)}{P \models Q \mid\sim X_1, \ldots, P \models Q \mid\sim X_n}$

L2 $\dfrac{P \models Q \mid\approx (X_1, \ldots, X_n)}{P \models Q \mid\approx X_1, \ldots, P \models Q \mid\approx X_n}$

Recognizing

R1 $\dfrac{P \models \rho(X_1); \ldots; P \models \rho(X_n)}{P \models \rho((X_1, \ldots, X_n))}$

Nonce Verification

NV $\dfrac{P \models Q \mid\sim X,\ P \models \sharp X}{P \models Q \mid\approx X}$

Jurisdiction

J $\dfrac{P \models Q \Rightarrow X,\ P \models Q \mid\approx X}{P \models X}$

A.2.2 Symmetric Enciphering

Seeing

S2 $\dfrac{P \lhd \{X\}_K,\ P \lhd Q \overset{K}{\leftrightarrow} R}{P \lhd X}$

Authentication

A1 $\dfrac{P \lhd \{X\}_K,\ P \models P \overset{K}{\leftrightarrow} Q,\ P \models \rho(X)}{P \models Q \mid\sim X,\ P \models Q \mid\sim K,\ P \models Q \mid\sim \{X\}_K}$

Key

K1 $\dfrac{P \lhd \{X\}_K,\ P \models P \overset{K}{\leftrightarrow} Q,\ P \models Q \mid\approx X}{P \models Q \mid\approx P \overset{K}{\leftrightarrow} Q}$

Freshness

F2 $\dfrac{(P \models \sharp X;\ P \models \sharp(P \overset{K}{\leftrightarrow} Q)),\ P \lhd P \overset{K}{\leftrightarrow} Q}{P \models \sharp\{X\}_K}$

Recognizing

R2 $\dfrac{P \models \rho(X),\ P \lhd P \overset{K}{\leftrightarrow} Q}{P \models \rho(\{X\}_K)}$

Contents[5]

C1 $\dfrac{P \models Q \mid\approx \{X\}_K,\ P \models P \overset{K}{\leftrightarrow} Q}{P \models Q \mid\approx X}$

A.2.3 Message Authentication Code

Authentication

A2 $\dfrac{P \lhd \mathrm{mac}(K, X),\ P \models P \overset{K}{\leftrightarrow} Q,\ P \lhd X}{P \models Q \mid\sim X,\ P \models Q \mid\sim K,\ P \models Q \mid\sim \mathrm{mac}(K, X)}$

Key

K2 $\dfrac{P \lhd \mathrm{mac}(K, X),\ P \models P \overset{K}{\leftrightarrow} Q,\ P \models Q \mid\approx X}{P \models Q \mid\approx P \overset{K}{\leftrightarrow} Q}$

Freshness

F3 $\dfrac{(P \models \sharp X;\ P \models \sharp(P \overset{K}{\leftrightarrow} Q)),\ P \lhd P \overset{K}{\leftrightarrow} Q}{P \models \sharp(\mathrm{mac}(K, X))}$

[5] This rule is needed whenever the freshness of a ciphertext is derived from the freshness of the key, like in [2].

Recognizing

R3 $\dfrac{P \models \rho(X),\ P \lhd P \overset{K}{\mapsto} Q}{P \models \rho(\mathrm{mac}(X,K))}$

Contents

C2 $\dfrac{P \models Q \mid\!\approx \mathrm{mac}(X,K),\ P \models P \overset{K}{\mapsto} Q}{P \models Q \mid\!\approx X}$

A.2.4 Shared Secrets

Seeing

S3 $\dfrac{P \lhd \langle X \rangle_S}{P \lhd X}$

Authentication

A3 $\dfrac{P \models P \overset{S}{\rightleftharpoons} Q,\ P \lhd \langle X \rangle_S}{P \models Q \mid\sim X,\ P \models Q \mid\sim S,\ P \models Q \mid\sim \langle X \rangle_S}$

Key

K3 $\dfrac{P \lhd \langle X \rangle_S,\ P \models P \overset{S}{\rightleftharpoons} Q,\ P \models Q \mid\!\approx X}{P \models Q \mid\!\approx P \overset{S}{\rightleftharpoons} Q}$

Freshness

F4 $\dfrac{P \models \sharp X;\ P \models \sharp(P \overset{S}{\rightleftharpoons} Q)}{P \models \sharp\langle X \rangle_S}$

Recognizing

R4 $\dfrac{P \models \rho(X)}{P \models \rho(\langle X \rangle_S)}$

Contents

C3 $\dfrac{P \models Q \mid\!\approx \langle X \rangle_S,\ P \models P \overset{S}{\rightleftharpoons} Q}{P \models Q \mid\!\approx X}$

A.2.5 Asymmetric Enciphering

Seeing

S4 $\dfrac{P \lhd \{X\}_K,\ P \lhd \overset{K}{\mapsto} P}{P \lhd X}$

Freshness

F5 $\dfrac{(P \models \sharp X;\ P \models \sharp \overset{K}{\mapsto} Q),\ P \lhd \overset{K}{\mapsto} Q}{P \models \sharp\{X\}_K}$

Recognizing

R5 $\dfrac{P \models \rho(X),\ P \models \overset{K}{\mapsto} P}{P \models \rho(\{X\}_K)}$

A.2.6 Signature

Seeing

S5 $\dfrac{P \lhd \{X\}_{K^{-1}},\ P \lhd \overset{K}{\mapsto} Q}{P \lhd X}$

Authentication

A4 $\dfrac{P \lhd \{X\}_{K^{-1}},\ P \models \overset{K}{\mapsto} Q,\ P \models \rho(X)}{P \models Q \mid\sim X,\ P \models Q \mid\sim K^{-1},\ P \models Q \mid\sim \{X\}_{K^{-1}}}$

Key

K4 $\dfrac{P \lhd \{X\}_{K^{-1}},\ P \models \overset{K}{\mapsto} Q,\ P \models Q \mid\!\approx X}{P \models Q \mid\!\approx \overset{K}{\mapsto} Q}$

Freshness

F6 $\dfrac{(P \models \sharp X;\ P \models \sharp \overset{K}{\mapsto} Q),\ P \lhd \overset{K}{\mapsto} Q}{P \models \sharp\{X\}_{K^{-1}}}$

Recognizing

R6 $\dfrac{P \models \rho(X),\ P \lhd \overset{K}{\mapsto} Q}{P \models \rho(\{X\}_{K^{-1}})}$

Contents

C4 $\dfrac{P \models Q \mid\!\approx \{X\}_{K^{-1}},\ P \models \overset{K}{\mapsto} Q}{P \models Q \mid\!\approx X}$

A.2.7 Hashing

Freshness

F7 $\dfrac{P \models \sharp X}{P \models \sharp(\mathrm{hash}(X))}$

Recognizing

R7 $\dfrac{P \models \rho(X)}{P \models \rho(\mathrm{hash}(X))}$

Contents

C5 $\dfrac{P \models Q \mid\sim \mathrm{hash}(X),\ P \lhd X}{P \models Q \mid\sim X}$

A.3 Formal Semantics

Obviously, it is straightforward to define a corresponding "operational" semantics as it was done in [3, 5]. We omit the details because we doubt if a semantics defined in this way is really helpful. Since these semantics are developed strictly according to the rules they can hardly serve as an independent proof for the soundness of the rules. Probably, the work of [1] was a step in the right direction.
We remark that it is still an open question in logic if the semantical approach is useful at all, cf. [9].

Acknowledgements

We thank our colleagues Uwe Blöcher, Hendrik Decker, Walter Fumy, and Sibylle Mund (Siemens AG) for helpful discussions and hints.

References

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

[2] R. Anderson, "UEPS - A Second Generation Electronic Wallet," *Computer Security - ESORICS 92*, Springer LNCS 648, pp. 411-418

[3] M. Burrows, M. Abadi, R. Needham, *A Logic of Authentication*, Report 39 Digital Systems Research Center, Pao Alto, California, 1989

[4] M. Burrows, M. Abadi, R. Needham, "Rejoinder to Nessett," *ACM Operating Systems Review*, Vol. 24 No. 2, pp. 39-40, 1990

[5] L. Gong, R. Needham, R. Yahalom, "Reasoning about Belief in Cryptographic Protocols," *Proc. of the 1990 IEEE Symposium on Research in Security and Privacy*, pp. 234-248, 1990

[6] R.C. Hauser, E.S. Lee, "Verification and Modelling of Authentication Protocols," *Computer Security - ESORICS 92* , Springer LNCS 648, pp. 141-154

[7] ISO/IEC Draft International Standard 9798-2 *Entity Authentication Using Symmetric Techniques*, 1993

[8] ISO/IEC Draft International Standard 10181-2.2 *Information Technology – Open Systems Interconnection – Security Frameworks for Open Systems: Authentication Framework*, 1993

[9] R. Kowalski, *Logic without Model Theory*, Imperial College, Department of Computing, London, 1993

[10] W. Mao, C. Boyd, "Towards Formal Analysis of Security Protocols," *Proc. of the Computer Security Foundations Workshop VI*, Franconia, IEEE Computer Society Press, pp. 147-158, 1993

[11] R. Needham, "Reasoning about Cryptographic Protocols," distributed at ESORICS 92, (but not in the proceedings)

[12] D.M. Nessett, "A Critique of the Burrows, Abadi, and Needham Logic," *ACM Operating Systems Review*, Vol. 24, No.2, pp. 35-38, 1990

[13] A.D. Rubin, P. Honeyman, *Formal Methods for the Analysis of Authentication Protocols*, Report of Center for Information Technology Integration, Univ. of Michigan, 1993