On BAN Logics for Industrial Security Protocols

Nesria Agray[1] Wiebe van der Hoek[1,2] Erik de Vink[3,4]

[1] Institute of Information and Computing Sciences, Utrecht University, the Netherlands N.Agray@fss.uu.nl, wiebe@cs.uu.nl
[2] Department of Computer Science, University of Liverpool, United Kingdom
[3] Department of Computer Science, Technische Universiteit Eindhoven, the Netherlands evink@win.tue.nl
[4] Leiden Institute of Advanced Computer Science, Leiden University, the Netherlands

Abstract. This paper reports on two case-studies of applying BAN logic to industrial strength security protocols. These studies demonstrate the flexibility of the BAN language, as it caters for the addition of appropriate constructs and rules. We argue that, although a semantical foundation of the formalism is lacking, BAN logic provides an intuitive and powerful technique for security analysis.

1 Introduction

New and upcoming techniques for the internet and also wireless telecommunication require or encourage agents to interchange more and more sensitive data, like payment instructions in e-commerce, strategic information between commercial partners, or personal information in for instance medical applications. Issues like authentication of the partners in a protocol, and the confidentiality of information therefore become of increasing importance: cryptographic protocols are used to distribute keys and authenticate agents and data over hostile networks. Although many of the protocols used look very intricate and hence waterproof, many examples are known of sensitive applications that were cracked and then furnished with new, 'improved' protocols.

The application of logical tools to the analysis of security protocols was pioneered by Burrows, Abadi and Needham. In [4] and [9] specific epistemic logics, collectively referred to as BAN logic, are proposed to deal with authentication issues. Earlier, standard epistemic logic ([6, 12]) has been applied successfully to reason about communication protocols, cf. the derivation of the alternating bit protocol in [11] or, more recently, the analysis of TCP [14]). In this setting, contrary to the security set-up of BAN, the implicit assumption is always that the network is not hostile. The work on epistemic logic has the last decades mainly focused on theoretical issues like variants of modal logic (cf. also [7] where epistemic notions are mixed with deontic ones), completeness, and derived notions like Distributed Knowledge and Common Knowledge.

In this paper we discuss the application of the BAN-framework to two of the most widely used security and authentication protocols. The first is applied in the Global System for Mobile communication (GSM, see [10]), an integration of several national cellular networks, which is now world-wide in use to transmit voice-data on the air, but also to send and receive e-mail, faxes and to get access to the internet. The second protocol, Secure Electronic Transactions (SET, see [13]) is an electronic commerce security protocol for doing business over the internet. It is developed under auspices of Visa and Mastercard, and well accepted to support credit card payments over the internet.

The aim of our exercise to apply BAN to those two protocols is the following. First, it goes without saying that such widely used but sensitive protocols have at least partially to be formally validated. Since the protocols under consideration are apparently not derived from a formal specification, before verifying any security goals on them one has to first trim down the massive manuals about the protocols to their bare essentials. Second, we want to demonstrate the flexibility of the BAN-framework by showing how specific issues of the two protocols can be smoothly accounted for by suitable extensions of the BAN-language and reasoning rules Here, we give an indication of the needed extensions for each of the protocols; full proofs are provided in [2]. Finally, the main research question behind our project is in which direction further fundamental work on BAN, or any logic for security and authentication, should proceed to sensibly deal with protocols that are in development or applied on a scale like GSM and SET.

2 BAN logic

BAN logic is a modal logic with primitives which describe the beliefs of agents involved in a cryptographic protocol. Using the inference rules of BAN logic the evolution of the beliefs of agents during a cryptographic protocol can be studied. Here we present some typical rules that we put to work, on the fly explaining the language of BAN.

The BAN-formalism is built on three sorts of objects: the agents involved in a security protocol, the encryption/decryption and signing/verification keys that the agents possess, and the messages exchanged between agents. The notation $\{M\}_K$ denotes a message encrypted using a key K. For a symmetric key K we have $\{\{M\}_K\}_K = M$ for any message M i.e., decrypting with key K a message M that is encrypted with K reveals the contents M. For a key pair $\langle EK, DK\rangle$ of a public encryption key EK and a private decryption key DK it holds that $\{\{M\}_{EK}\}_{DK} = M$ for any message M. Likewise, for a key pair $\langle SK, VK\rangle$ of a private signing key SK and a public verification key VK it holds that $\{\{H\}_{SK}\}_{VK} = H$ for any hash value H. Hash values are obtained by application of a one-way collision-free hash-function. Proving the hash value $H(m)$ of a message m is a means to demonstrate that m is known, without revealing it (cf. also [2]).

In BAN, we have a number of operators describing the beliefs of agents, for which the usual modal properties apply, like P `believes` $(A \rightarrow B) \rightarrow (P$ `believes` $A \rightarrow P$ `believes` $B)$. On top of that we have the operators sees and possesses (cf. [4, 9]). The following rules are an illustration of some of the authentication and encryption rules:

(1) P believes secret$(K, P, Q) \wedge P$ sees $\{X\}_K$
$\rightarrow P$ believes Q said X

(2) P believes belongs_to $(VK, Q) \wedge P$ sees $\{X\}_{SK}$
$\rightarrow P$ believes Q said X

(3) P believes fresh$(X) \wedge P$ believes Q said X
$\rightarrow P$ believes Q believes X

(4) P possesses $DK \wedge P$ sees $\{X\}_{EK} \rightarrow P$ sees X

Intuitively, **(1)** says that if an agent P believes that it shares the symmetric key K with an agent Q, and agent P receives a message X encrypted under K, then agent P believes that agent Q once said message X. This rule addresses symmetric encryption. In a similar way, **(2)** models digital signatures. If an agent P believes that the verification key VK belongs to an agent Q, then P concludes, confronted with a message or hash encrypted with the corresponding signing key SK, that the message or hash originates from the agent Q. Regarding rule **(3)**, if an agent P believes that certain information is new, i.e. constructed during the current protocol run, and P furthermore believes that Q conveyed this information, then P concludes that the agent Q believes himself this information. (Underlying this is the overall assumption in BAN logic of honesty of agents; the participating agents' behavior is consistent with the particular protocol.) According to **(4)**, if an agent P sees an encrypted message and P possesses the decryption key then P can read the message itself.

Analysis of an authentication protocol using BAN-logic is comprised of four steps: (1) first, the initial beliefs of the participating agents are formulated; (2) the protocol security goals are stated; (3) the effect of the protocol's messages is formalized in BAN and (4), the final beliefs are shown to fulfil the goals.

3 Analyzing the GSM security protocol

The GSM protocol, Global System for Mobile communication, is a protocol for mobile telephony operating world-wide since the beginning of the 1990s. Abstracting away, in our analysis, from a lot of details we distinguish three parties involved in the protocol: the mobile station MS, which is the subscriber's interface to the network, connected via a radio link, the home location register HLR which is an intelligent database where a subscriber's specific parameters are stored, and the visitor location register VLR, a database that temporarily stores user specific parameters. The VLR's role is to support the HLR by handling most of the user-related queries.

If a mobile station MS wants the GSM service, it must prove its identity to the network. To this end, the mobile station sends a service request via the radio path to the VLR. The VLR then forwards this identity to the HLR, which generates a fresh random number RAND and computes a result RES by applying a one-way and collision free function $\mathtt{A}_3$ to a key K and RAND, i.e. $\mathtt{RES} = \mathtt{A}_3(\mathtt{K},\mathtt{RAND})$. The identity key K is a shared secret between the HLR and the MS, but not known to VLR. If MS is able to generate the same value RES when provided with RAND, it has proven its identity. The HLR also computes an encryption key $\mathtt{K}'$ that will ensure privacy of MS when it further communicates with the VLR. This is done using an algorithm $\mathtt{A}_8$, hence $\mathtt{K}' = \mathtt{A}_8(\mathtt{K},\mathtt{RAND})$. The HLR then sends the triple $\langle \mathtt{RAND}, \mathtt{RES}, \mathtt{K}' \rangle$ to the VLR. Upon receipt of this triple, the VLR forwards RAND to the MS and waits for an appropriate response from the MS. The mobile station can calculate RES from the key K and the challenge RAND using the algorithm $\mathtt{A}_3$. Then, MS uses RAND together with its key K and the algorithm $\mathtt{A}_8$ to calculate the session key $\mathtt{K}'$ and stores it locally.

We next give the crucial rules that we add to BAN. For the modeling we add various auxiliary functions and predicates like ch and comp. Free variables are assumed to be universally quantified.

(4) $\mathtt{VLR\ believes}\ [\mathtt{HLR\ said\ ch}(x,y,z) \rightarrow \mathtt{rand}(x) \wedge \mathtt{res}(y) \wedge \mathtt{skey}(z)]$

(5) $\mathtt{VLR\ believes}\ [(\mathtt{MS\ said\ res}(y) \wedge \mathtt{HLR\ said\ ch}(x,y,z) \wedge \mathtt{fresh}(x))$
$\rightarrow \mathtt{MS\ possesses}\ z]$

(6) $P\ \mathtt{believes}\ (P\ \mathtt{possesses}\ K \wedge Q\ \mathtt{possesses}\ K)$
$\rightarrow P\ \mathtt{believes\ secret}(K,P,Q)$

(7) $\mathtt{MS\ believes}\ [(\mathtt{VLR\ said\ rand}(x)) \wedge \mathtt{fresh}(x) \wedge$
$\mathtt{ikey}(K) \wedge \mathtt{a}_8(A) \wedge K' = \mathtt{comp}(A,x,K) \wedge$
$\mathtt{secret}(\langle K,A \rangle, \mathtt{MS}, \mathtt{HLR}) \rightarrow \mathtt{VLR\ possesses}\ K']$

Rule **(4)** says that the VLR knows the meaning of the three values in the challenge triplet $\mathtt{ch}(x,y,z)$ received from the HLR. Our rule **(5)** expresses that the session key is calculated by the mobile station at the same time the calculation of the res-value takes place. So at the moment that the right fresh res-value is received from the mobile station, the VLR knows that the mobile station is in possession of the right session key. Rule **(6)** is an obvious, but tricky rule. If some agent R possesses the key K as well, it is no longer justified to consider K a secret between P and Q. Finally, concerning rule **(7)**, when the mobile station receives a random challenge, it computes the corresponding response value and session key using the appropriate secret A8-algorithm, the random value and the secret identity key K that it shares with the HLR. It is implicit in the GSM authentication protocol that the mobile station trusts the network on the matter of the random value, i.e. MS assumes that it originates from the HLR. Furthermore, it is assumed that at the moment that the MS receives a random value from the VLR, the MS concludes that the VLR possesses the corresponding session key.

We have omitted rules concerning freshness here. Also, space does not allow us to formulate the initial beliefs of the three partners. Instead, we formulate the authentication question and the question concerning the agreement on the encryption key. The VLR must check whether a mobile station with a given identifier is really registered in the system, and whether the mobile station has the identity it claims. This is done by comparing the res-values received from the mobile station at one side and from the HLR at the other side. If the two values match then the VLR knows that it is indeed talking to the mobile station. The first authentication question is as follows:

(Q1) $\exists x : \texttt{fresh}(x) \wedge \texttt{VLR believes res}(x) \wedge \texttt{MS said res}(x)$

The next question that we discuss concerns the agreement on the session key. At the end of the protocol the VLR must know that it shares the right session key with the mobile station. The mobile station must believe the same statement. This is modeled as follows:

(Q2) $\exists z : \texttt{MS believes } (\texttt{fresh}(z) \wedge \texttt{secret}(z, \texttt{VLR}, \texttt{MS})) \wedge$
$\texttt{VLR believes } (\texttt{fresh}(z) \wedge \texttt{secret}(z, \texttt{VLR}, \texttt{MS}))$

In our full paper [2], we specify a set of initial beliefs of the three parties involved, translate the effect of the messages into BAN, and prove, amongst others, **(Q1)** and **(Q2)**.

4 Analyzing the SET protocol

In this section we describe the SET-protocol (see [13]). SET stands for *Secure Electronic Transactions* and is designed for credit card payments over the Internet. By combining both symmetric and public-key cryptographic techniques SET provides several security services at an adequate performance. In particular, SET provides message integrity together with blinding of information, such that (i) no banking information of the customer is revealed to the merchant, and (ii) no purchase information of the customer is disclosed to the bank. For this latter services SET deploys so-called dual signatures.

Let us first explain standard digital signatures. Each agent participating in a run of SET holds a pair of keys $\langle SK, VK \rangle$ consisting of a *signing key* and a *verification key*. The signing key is private, the verification key is public. If Alice wants to sign a message M, she first computes the hash value $\texttt{hash}(M)$ of M, encrypts this value with her signing key SK and attaches this signature $\{\texttt{hash}(M)\}_{SK}$ to the message M, thus yielding $(M, \{\texttt{hash}(M)\}_{SK})$. When Bob receives a signed message (M, S), presumably coming from Alice, he first computes the hash value $\texttt{hash}(M)$ of M, then he retrieves the corresponding verification key VK of Alice, decrypts the signature S with the key VK, and checks whether the resulting value $\{S\}_{VK}$ equals the earlier value $\texttt{hash}(M)$. Note, since $\langle SK, VK \rangle$ is a key pair with private key SK and public key VK, it holds that $\{\{X\}_{SK}\}_{VK} = X$. So, if the two values agree, Bob concludes that Alice signed the message as she is the only agent in possession of the signing key that is needed to produce the signature for the hash value of the message. The BAN rule that we have used to model this is

(8) $P \texttt{ sees } (M, S) \wedge P \texttt{ believes } VK \texttt{ belongs\_to } Q \wedge$
$\texttt{hash}(M) = \{S\}_{VK} \rightarrow P \texttt{ believes } Q \texttt{ said } M.$

Next we discuss dual signatures. SET makes use of so-called dual signatures to blind some information for one agent while showing the same information to another one, but in the mean time providing the two agents with a reference to the information. Sending each part of the information separately to the intended parties does not work, because the two parts of information are related and this relationship is important for the remainder of the protocol. For example, a purchase request from a card holder C contains both order data and payment data. The payment data is intended for the bank B and must be hidden from the merchant M. Dually, it is no business of the bank to get to know the actual goods that the customer buys. However, at some point in time after the particular purchase request, the merchant M want to capture his money for the purchase from the bank B. For this M and B need to refer to the same procurement.

To see this at work, suppose that an agent P has signing key SK, verification key VK and wants to send some information I to an agent Q. Additionally, Q needs a way to link I to some other information J that will not be revealed to him. To realize this, the agent P sends Q the message

$$(I, \mathtt{hash}(J), \{\mathtt{hash}(\mathtt{hash}(I) \mid \mathtt{hash}(J))\}_{SK}),$$

where the so-called dual hash $\mathtt{hash}(\mathtt{hash}(I) \mid \mathtt{hash}(J))$ is obtained by hashing the juxtaposition of $\mathtt{hash}(I)$ and $\mathtt{hash}(J)$. By encryption of the dual hash with her private signing key, agent P testifies that both the information I and the hash $\mathtt{hash}(J)$ originate from her. The agent Q can verify this by using the corresponding public verification key of P, similar to the case of standard digital signatures as described above. However, Q does not know anything about the information J. Now, if yet some other agent R obtains from P the message

$$(\mathtt{hash}(I), J, \{\mathtt{hash}(\mathtt{hash}(I) \mid \mathtt{hash}(J))\}_{SK}),$$

she knows that the message indeed has been sent by P, but has no clue about the information I. The crucial point is that agents Q and R both are aware that $I\&\mathtt{hash}(J)$ and $\mathtt{hash}(I)\&J$, respectively, are related. Our relevant BAN rule is as follows:

(9) P $\mathtt{sees}$ $(I, G, \{H\}_{SK}) \wedge P$ $\mathtt{believes}$ $\mathtt{belongs\_to}$ $(VK, Q) \wedge$
$\mathtt{hash}(\mathtt{hash}(I)|G) = H \rightarrow P$ $\mathtt{believes}$ $(Q$ $\mathtt{said}$ $I \wedge \mathtt{dhash}_1(I, G, H))$

Here, $\mathtt{dhash}_1$ is a predicate stating that the hash of I concatenated with G will hash to H. There is also a similar rule for a message $(F, J, \{H\}_{SK})$ for a hash value F, a message J and dual hash H using an auxiliary predicate $\mathtt{dhash}_2$.

Besides user authentication, during a run of SET, the card holder, the merchant and the payment gateway have to agree on a purchase and the corresponding cost. So after the run of the protocol, the three participants must store consistent information about the purchase. For example, the three parties must store the same purchase amount. This is achieved when the three parties agree on the linkage between the order data and the payment instructions. As example, for P, this is modeled in security requirement **(Q3)** as follows:

(Q3) P $\mathtt{believes}$ $\big(\mathtt{pay\_instr}(C, \mathtt{PData}) \wedge (\exists \mathtt{OData} : \mathtt{link}_2(\mathtt{OData}, \mathtt{PData}) \wedge$
M $\mathtt{believes}$ $\mathtt{link}_1(\mathtt{OData}, \mathtt{PData}) \wedge$
C $\mathtt{believes}$ $\mathtt{link}_1(\mathtt{OData}, \mathtt{PData}) \wedge \mathtt{link}_2(\mathtt{OData}, \mathtt{PData}))\big)$

The payment gateway P cannot see the order data, but believes in the existence of an order $\mathtt{OData}$ generated by the card holder which is linked to the card holder's payment instructions. Therefore, the well-known Barcan-formula P $\texttt{believes}$ $\exists x : \phi \rightarrow \exists x : P$ $\texttt{believes}$ ϕ is not valid in our framework. Furthermore, P has to trust this link, and has to know that both the merchant and the card holder believe in the same link. Similar rules apply to M and C.

In [2] we present the translation of the protocol in BAN. In order to give the reader a flavor of this, let us consider the third transaction of the protocol, in which the card holder sends its purchase request to the merchant.

(PReq) $C \rightarrow M$:
$(\texttt{OData}, \texttt{hash}(\texttt{PData}), \{\texttt{hash}(\texttt{hash}(\texttt{OData}) \mid \texttt{hash}(\texttt{PData}))\}_{SKC}),$
$\{\texttt{PData}, \texttt{hash}(\texttt{OData}), \{\texttt{hash}(\texttt{hash}(\texttt{OData}) \mid \texttt{hash}(\texttt{PData}))\}_{SKC}\}_K,$
$\{K\}_{EKP},$
$\texttt{sigCert}(C, \mathit{SKC})$

That is, card holder C sends to merchant M the following: (1) a message containing the order data and the hash of the payment data as clear text, but dually signed with the signing key of agent C; (2) a message containing the hash value of the order data and the payment data, dually signed with the signing key of C and encrypted with a random session key K; (3) the session key K encrypted by the payment gateway's encryption key *EKP*, so that only the payment gateway, hence not the merchant, can learn this session key and, consequently, the payment data encrypted with the key K; (4) a public-key certificate $\texttt{sigCert}(C, \mathit{SKC})$ for the cardholder's signing key which includes the corresponding verification key.

In order to compute the effects of this message on the beliefs of M, one first has to know the beliefs of M prior to performing the transaction **(PReq)**. For example, one belief concerns the possession of the verification key of the root Certification Authority for the public-key certificate for C's signing key. It is clear that the analysis makes such an assumption explicit. Although the SET protocol is rather detailed, in [2] we were able to give a formal proof of **(Q3)** as mentioned, as well as for other authentication goals.

5 Future research

We see the following lines of research triggered by the experiences reported in this paper. First, the formidable proofs in [2] call for tool-support for the verification of non-trivial security goals. Apart from automated translation, our experiences reveal the need for flexible computer-assistance to incorporate logical extensions and to guide proof construction in the adapted framework.

Second, it is well known that the semantics of BAN logic is under debate (cf. [1, 15, 16, 3]). At present, we cannot claim that our rules are sound or complete. On the one hand, this questions the impact of the derived results (what does it mean that some string has been derived?), but on the other hand, strengthens the call for an adequate model. To our opinion the various proposed semantics for BAN hardly allow for changes in the logic. This raises the question for a model which easily treats extensions as described in this paper. In the protocols considered not only information is exchanged, but also requests. Therefore, one might want to stress the goals of an agent as part of his mental state. However, the framework should not only reflect the state of the agent, but also the *dynamics*: agents learn from the messages they receive, thus calling for a language that deals with knowledge together with epistemic actions like learning and updating. To this aim we try in a current project to exploit the knowledge games semantics of [5] in the setting of authentication logics.

References

1. M. Abadi and M. Tuttle, A Semantics for a Logic of Authentication, in *Proceedings of the ACM Symposium on Principles of Distributed Computing,* p201–216, 1991.
2. N. Agray, *Formal Verification of Security Protocols Using BAN Logic,* Master Thesis, Utrecht University and KPN Research Leidschendam. To appear, 2001.
3. A. Bleeker and L. Meertens, A Semantics for BAN Logic, *Proc. DIMACS workshop on Design and Formal Verification of Protocols,* 1997, `http://dimacs.rutgers.edu/Workshops/Security/program2/program.html`.
4. M. Burrows, M. Abadi and R. Needham, A Logic of Authentication, *ACM Transactions on Computer Systems,* vol. 8, p18–36, 1990.
5. H. van Ditmarsch, *Knowledge Games.* PhD. thesis, Groningen, 2000, available at `http://tcw2.ppsw.rug.nl/~hans`.
6. R. Fagin, J.Y. Halpern, Y. Moses and M.Y. Vardi, *Reasoning About Knowledge,* MIT Press, 1995.
7. M. Fasli, On Commitments, Roles and Obligations. This volume.
8. M. Fitting and R. Mendelsohn, *First order modal logic,* Kluwer, 1998.
9. L. Gong, R. Needham and R. Yahalom, Reasoning about Belief in Cryptographic Protocol Analysis, *Proc. IEEE Symp. on Research in Security and Privacy,* p234–248, 1990.
10. H. Gunnar, *GSM Networks: Protocols, Terminology, and Implementations,* Artech House 1999.
11. J.Y. Halpern and L.D. Zuck, A Little Knowledge Goes a Long Way: Simple Knowledge-Based Derivations and Correctness Proofs for a Family of Protocols, *Proc. 6th ACM Symp. on Principles of Distributed Computing,* 1987, p268–280.
12. J.-J.Ch. Meyer and W. van der Hoek, *Epistemic Logic for AI and Computer Science,* Cambridge University Press, 1995.
13. The SET Standard Book 2: Programmer's Guide, version 1.0, SETCO 1997, `www.setco.org`.
14. F. Stulp and R. Verbrugge, A knowledge-based algorithm for the Internet protocol TCP, to appear in the *Bulletin of Economic Research,* 2001. Also at `http://tcw2.ppsw.rug.nl/prepublications`
15. P. Syverson, The Use of Logic in the Analysis of Cryptographic Protocols, in *Proc. IEEE Symp. on Research in Security and Privacy,* 1991.
16. G. Wedel and V. Kessler, Formal Semantics for Authentication Logics, *Proc. ESORICS'96,* p219–241, 1996.