THE USE OF MODAL LOGICS IN THE SECURITY PROTOCOLS ANALYSIS

Ing. Pavel OČENÁŠEK, Doctoral Degree Programme (3)
Dept. of Information Systems, FIT, BUT
E-mail: ocenaspa@fit.vutbr.cz

Mgr. Roman TRCHALÍK, Doctoral Degree Programme (3)
Dept. of Information Systems, FIT, BUT
E-mail: trchalik@fit.vutbr.cz

Supervised by: Prof. Miroslav Švéda

ABSTRACT

Traditionally, security protocols have been designed and verified using various techniques. Formal logics have been used to identify a number of flaws in protocols previously considered to be secure. The selection of proper modal logic is a crucial goal in the protocol analysis process. This paper gives a comparative study of modal logics, which are widely used in modeling of security protocols.

1 INTRODUCTION

Weaknesses in security protocols (SP) are hard to identify, as they can be the result of subtle design flaws. The formal verification of security protocols may be done in two ways. One possibility is to use a modal logic of authentication. The other possibility is to use general purpose formal methods. This paper provides a survey through the world of modal logics used in SP and gives a comparison of different variants of modal logics and their target areas of application.

2 MODAL LOGICS APPROACH

The general approach is based on the use of logics of belief and/or knowledge. Such logics involve a process of deductive reasoning. An attempt is made to derive the protocol goals by applying a set of axioms and inference rules to the assumptions and message exchanges of the protocol. Formal logics can be used to generate comparably short and simple proofs. Logics-based formal verification involves the following steps:

1. Formalization of protocol messages

2. Specification of initial assumptions

3. Specification of protocol goals

4. Application of logical postulates

A successfully verified protocol can be considered secure within the limitations of the logic. On the other hand, the results of a failed verification assist in the identification of missing initial assumptions and design-flaws of the protocol.

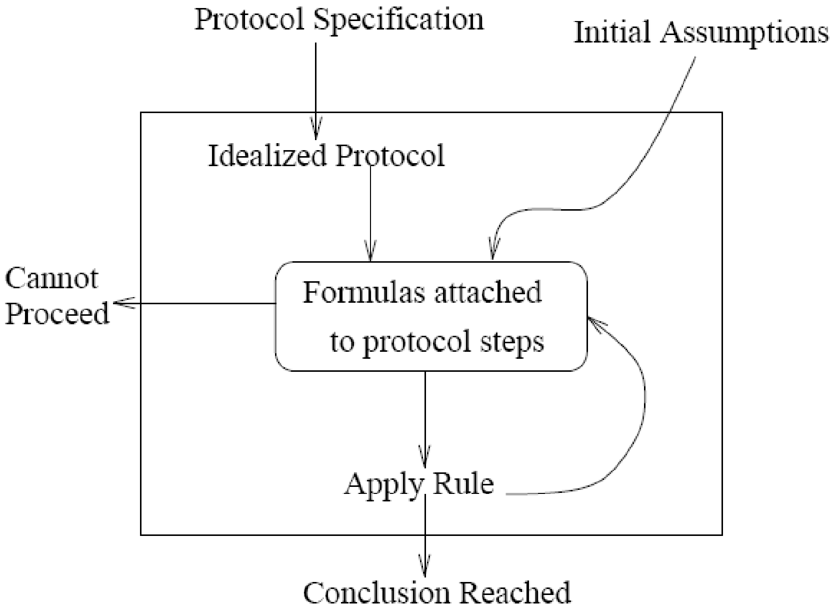


Fig. 1: *Protocol analysis with the modal logic.*

In the following chapters we give a more detailed description of selected modal logics and their use.

2.1 BAN

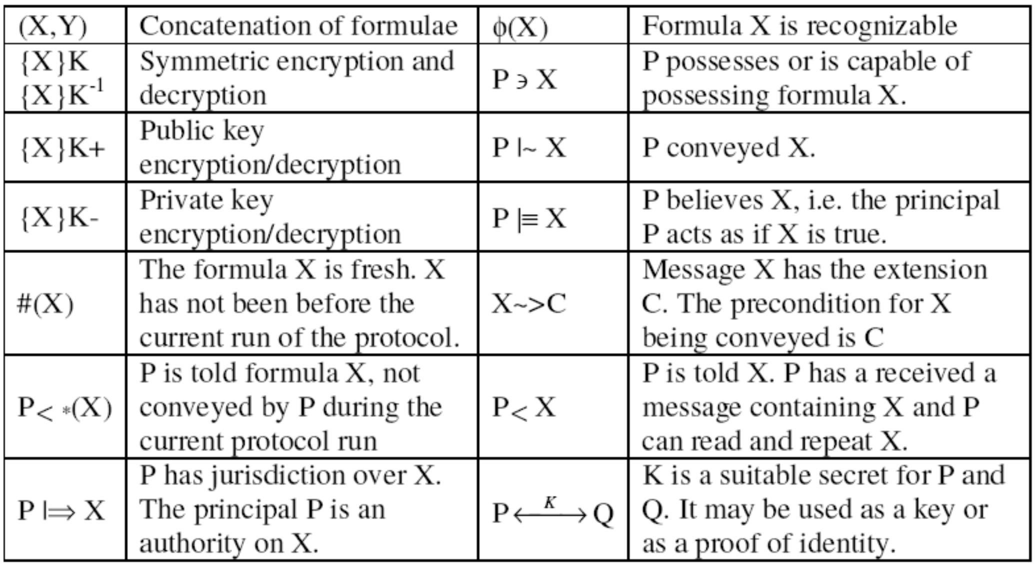
The most well-known modal logic is the Burrow, Abadi and Needham (BAN) logic [4] [1] [2]. In order to verify a protocol using BAN logic, a set of hypotheses describing the set of initial beliefs is laid down, and each step of the protocol is translated into BAN facts. The BAN logic has been used successfully for the verification of many protocols. But the BAN logic also has limitations, which have been the subject of many research activities.

The BAN-formalism is built on three sorts of objects: the subjects involved in a security protocol, the encryption/decryption and signing/verification keys that the subjects possess, and the messages exchanged between subjects. 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 content M. For a key pair $<EK, DK>$ 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 $<SK, VK>$ of a private signing key SK and a public verification key VK it holds $\{\{H\}_{SK}\}_{VK} = H$ for any hash value H. Hash values are obtained by aplying one-way collision-free hash-function. Proving the hash value $H(m)$ of a message m is a mean to demonstrate that m is known, without revealing it.

An important limitation on BAN is the type of protocols to which it can be applied. Diffie-Hellman protocols underly much of modern authenticated key distribution

2.2 GNY

The extension of BAN logic is the logic introduced by Gong, Needham and Yahalom [5], usually referred to as the GNY logic. The following table shows the constructs of GNY.



(X,Y)	Concatenation of formulae	$\phi(X)$	Formula X is recognizable
$\{X\}K$ $\{X\}K^{-1}$	Symmetric encryption and decryption	$P \ni X$	P possesses or is capable of possessing formula X.
$\{X\}K+$	Public key encryption/decryption	$P \mid\sim X$	P conveyed X.
$\{X\}K-$	Private key encryption/decryption	$P \mid\equiv X$	P believes X, i.e. the principal P acts as if X is true.
$\#(X)$	The formula X is fresh. X has not been before the current run of the protocol.	$X \sim> C$	Message X has the extension C. The precondition for X being conveyed is C
$P < *(X)$	P is told formula X, not conveyed by P during the current protocol run	$P < X$	P is told X. P has a received a message containing X and P can read and repeat X.
$P \mid\Rightarrow X$	P has jurisdiction over X. The principal P is an authority on X.	$P \xleftrightarrow{K} Q$	K is a suitable secret for P and Q. It may be used as a key or as a proof of identity.

Tab. 1: *Constructs of GNY logic.*

In particular, the GNY logic does not assume that redundancy exists in encrypted messages. Instead, it introduces the *notion of recognizability* to represent the fact that a subject expects certain formats in the messages it receives.

2.3 SG

Another one - SG logic [7] - is a revised and extended version of the formalism introduced by Guergens in 1996. By using the notion of message-types, the message property *not_said* is defined. In general, this formalism is capable of detecting the possibility of certain reflection and interleaving attacks on security protocols using a symmetric algorithm. By applying SG logic we are able to show the well-known weakness in the Neuman-Stubblebine protocol formally.

2.4 SVO

The SVO logic [6] uses the notation already introduced for BAN, with the following main additions:

$\neg\varphi$: Negations of formulae are added to the language

P says X : *X* is a message *P* said recently. Like BAN's "*P said X*" but *P* must have said *X* since the beginning of current epoch.

P has X : *X* is a message *P* can see. This includes messages:

- initially available to *P*,

- received by *P*,
- freshly generated by *P*, and
- constructible by *P* from the above.

Comparison of Protocol Analysis Steps:

BAN Analysis

1. Idealize the protocol.
2. Write assumptions about initial state.
3. Annotate protocol. For each message *“P → Q: M”* of the idealized protocol, assert *“Q received M”*
4. Use the logic to derive the beliefs held by protocol principals.

SVO Analysis

1. Write assumptions about initial state.
2. Annotate protocol. For each message *“P → Q: M”* of the (not idealized) protocol, assert *“Q received M”*
3. Assert comprehensions of received messages.
4. Assert interpretations of comprehended messages.
5. Use the logic to derive beliefs held by protocol principals.

Another variation of this logic - SVD logic has many more rules than SVO logic, However, the SVD logic is not well suited for automation with theorem provers such as Isabelle, neither for proving negative results.

2.5 CKT5

Bieber [3] extends the epistemic logic of Hintikka. This logic of communication in a hostile environment, called CKT5, allows a user to describe the states of knowledge and ignorance associated with the communication via encrypted messages. Bieber also extends the logic of knowledge and time, KT5 of Sato [6] with operators that relate directly to the sending and receiving messages.

The CKT5 specification given in [3] allows each honest principal participating in a protocol to play exactly one role. This restriction could cause, that attacks that rely on having the same principal act both as initiator and a responder, for example, are missed. There were later done some corrections of this limitation by upgrading the one-to-one relation between roles and principal to many-to-one correspondence. Therefore, a given principal was now associated with a set of roles, an entity also known as a *multi-role*.

Clearly, if the protocol at hand is constrained in such a way that every honest principal can play at most one role, then no multi-role flaws can be uncovered. Even in this limited settings, the CKT5 as a specification language does not prevent the possibility of all attack.

3 CONCLUSIONS

There are some more modal logics used for security protocols verification [6] [7]. This paper deals just with those best known and widely used. The description of all the logics in detail is beyond the scope of this paper.. In general we can say that, the basic constructs for logic verification were outlined with foundation of BAN logic [4]. And, since there are essentially expansions, e.g. SVO logic encompasses BAN itself as well. GNY [5] and AT logic add to and reformulate BAN to better reason about the same class of protocols. Another, VO logic adds rules to reason about key-agreement protocols.

The selection of proper modal logic for security protocol verification is the crucial goal in the protocol analysis process. The main contribution of this paper consists in the comparison of various logics, their target area of use and description of specific advantages.

ACKNOWLEDGEMENTS

This paper was created as a part of the research which has been supported by the Grant Agency of the Czech Republic through the following grants: GACR 102/05/0723: A Framework for Formal Specifications and Prototyping of Information System's Network Applications, GACR 102/05/0467: Architectures of Embedded Systems Networks, GACR 102/05/H050: Integrated approach to education of PhD students in the area of parallel and distributed systems, and by the Czech Ministry of Education in frame of MSM 0021630503 Research Intention MIKROSYN: New Trends in Microelectronic Systems and Nanotechnologies.

REFERENCES

[1] Abadi, M., Tuttle, N.: A Semantic for a Logic of Authentication, In: Proceedings of the ACM Symposium on Principles of Distributed Computing, 2001, pp. 201-216

[2] Agray, N., van der Hoek, W., de Vink, E.: On BAN Logics for Industrial Security Protocols, In: CCEMAS, p. 8

[3] Bieber, P.: A logic of communication in a hostile environment. In: Proceedings of the Computer Security Foundation Workshop III, 1990, pp. 14-22

[4] Burrows, M., Abadi, M., Needham, R.: A Logic of Authentication, ACM Transactions on Computer Systems, vol. 8, 1990, pp. 18-36

[5] Gong, L., Needham, R., Yahalom, R.: Reasoning about belief in cryptographic protocols. IEEE Computer Society Synopsis on Research in Security and Privacy, 1990, pp. 234-248

[6] Rubin, A. D., Honeyman, P.: Formal methods for the analysis of authentication protocols, Technical Report 93--7, Center for Information Technology Integration, Department of Electrical Engineering and Computer Science, University of Michigan, 1993, p. 35

[7] Gurgens S.: SG Logic - A Formal Analysis Technique for Authentication Protocols, In: Security Protocols, vol. 1316 of LNCS, Springer-Verlag, 1997.

[8] Syverson, P., van Oorschot, P.: On Unifying Some Cryptographic Protocols Logics, In: Proceedings of the 13th IEEE Symposium on Security and Privacy. IEEE Comp. Society Press, 1994