Formal methods for the Analysis and Design of Cryptographic Protocols: A state-of-the-art review

S.Gritzalis

Department of Informatics
University of Athens
TYPA Buildings, Athens GR-15771, Greece
tel.: +30-1-7291885, fax: +30-1-7219561
Department of Informatics
Technological Educational Institute (T.E.I.) of Athens
Ag.Spiridonos St. Aegaleo GR-12210, Greece
tel.: +30-1- 5910974, fax.: +30-1-5910975
email: sgritz@teia.ariadne-t.gr

N.Nikitakos

Hellenic Naval Academy
PO Box 80318, Piraeus 18510, Greece
tel.: +30-1-4625993, fax: +30-1-4181768
email:nikitas@naxos.esd.ntua.gr
Department of Informatics
Technological Educational Institute (T.E.I.) of Athens
Ag.Spiridonos St. Aegaleo GR-12210, Greece
tel.: +30-1- 5910974, fax.: +30-1-5910975

P.Georgiadis

Department of Informatics
University of Athens
TYPA Buildings, Athens GR-15771, Greece
tel.: +30-1-7291885, fax: +30-1-7219561
email: georgiad@di.uoa.ariadne-t.gr

Abstract

A state-of-the-art review is presented concerning formal methods for the design and analysis of cryptographic protocols over open networks and distributed systems. The most commonly followed approaches to the applications of related formal methods are reviewed, followed by the examination of robustness principles and application limitations as rules of thumb.

Communications and Multimedia Security Vol. 3 S. Katsikas (Ed.)
©IFIP 1997 Published by Chapman & Hall

Finally the modern trends for the use of formal methods in the design of new cryptographic protocols are discussed.

Keywords
Cryptographic Protocols, Protocol Analysis Tools, Open Networks and Distributed Systems Security.

1 INTRODUCTION

A protocol is a set of rules and conventions that define the communication framework between two or more parties. These parties are end-users, processes or computing systems, which will generically refer to as principals. In cryptographic protocols part of at least one message is encrypted. Cryptographic protocols are intended to establish secure communication over potential insecure open networks and distributed systems. These protocols use encryption and decryption techniques to achieve goals such as authentication of principals and services, integrity, secrecy, origin, destination, order, and timeliness of the messages, and the distribution of cryptographic keys. Unfortunately these open networks and distributed systems may contain a number of hostile intruders who may try to subvert the protocol design goals.

Given such requirements, it is not surprising that there have been several examples (Denning, 1981) (Needham, 1978) (Needham, 1987) of cryptographic protocols that were published, believed to be sound, and later shown to have security flaws. In the last few years, researchers have developed several different means for detecting protocol failures. As in the analysis of conventional communication protocols, there have been two kinds of techniques applied to this problem: attempting to construct possible attacks, using algebraic properties of the algorithms in the protocols; and attempting to construct inferences using specialised logics based on a notion of knowledge and belief, that protocol participants can confidently reach desired conclusions.

Attack-construction tools construct probable attack sets based on the protocol's algorithms algebraic properties. These methods (Dolev, 1983) (Kemmerer, 1989) (Meadows, 1992) (Millen, 1995) (Sidhu, 1986), (Varadharajan, 1989) are targeted towards ensuring authentication, correctness or security properties and are not dependent on the correctness of a proposed logic. Their main disadvantage lies mainly in the big number of possible events that must be examined.

Inference-construction tools are utilising modal logics similar to those that have been developed for the analysis of the evolution of knowledge and belief in distributed systems. These methods (Burrows, 1990) (Gong, 1990) (Syverson, 1994) are widely used. A number of specific problems associated with them (Brackin, 1996a) (Kessler, 1994) (Syverson, 1991) (Syverson, 1993) (Gritzalis, 1996), range from their inability to analyse zero knowledge protocols or to detect parallel session multi-role flaws and mainly to the difficulty of transforming messages and prepositions to idealized messages.

In this paper we give a review of the state-of-the-art in the application of formal methods to the design, and analysis of cryptographic protocols and we will attempt to outline some of the major trends of research in this specific area.

The remainder of this paper is organised as follows: In Section 2 and Section 3 we describe the two most commonly followed approaches to the applications of formal methods to cryptographic protocol analysis, as mentioned before. In Section 4 some helpful principles and limitations for the design of effective and reliable cryptographic protocols are presented. In Section 5 modern specification languages for automatically analysing cryptographic protocols are presented. In Section 6 the recent trends for the use of formal methods in the design of new cryptographic protocols are discussed. Section 7 concludes the paper.

2 ATTACK CONSTRUCTION TOOLS METHODS

Attack construction tools can be distinguished into three categories based on their theoretical foundation:

- methods based on general purpose validation languages and tools
- algebraic simplification theoretic model methods.
- expert system, scenario based methods

Accordingly, we will describe the basic features for every method.

2.1 Methods based on general purpose validation languages and tools

These methods analyse a cryptographic protocol as any other program whose correctness they are trying to prove. This is done by specifying the protocol: as a finite-state machine (Sidhu, 1986), (Varadharajan, 1989), using predicate calculus (Kemmerer, 1989), or within a process algebra (Roscoe, 1995), (Lowe 1996).

Some researchers (Sidhu, 1986), (Varadharajan, 1989) map the protocol to a finite-state machine. The analysis method proposed by (Sidhu, 1986), verifies the basic properties of a number of protocols, detects basic flaws, but can not detect flaws due to the re-use of old messages as no temporal assumptions are used. The method proposed by (Varadharajan, 1989) also verifies the basic properties of a number of protocols, but exhibits a number of problems as the number of states increases. In addition, in order to deal with flaws related to the re-use of old messages the author proposes to incorporate into the analysis data from the session key message contents.

Another approach proposed in (Kemmerer, 1989) is based on predicate calculus extensions. This method is using the specification language Ina Jo and the Formal Development Methodology. Ina Jo (Scheid, 1988) is a non-procedural assertion language that is an extension of first-order predicate calculus. Formal specifications written in Ina Jo specify definitions, initial conditions, transforms, axioms, and criteria. Criteria are used to specify critical requirements for a secure state. Ina Jo formal specifications can then be executed and verified by related tools, such as Inatest. This approach has been successful in locating both active and passive attack flaws, since in both cases the intruder is a separate entity in the model's mathematical framework.

A more recent approach is based on modelling the communicating principals and the intruder as Communicating Sequential Processes (CSP). The proposed method can be used to formalise messages, traces, intruders, and nonce challenges. The Failures Divergence's Refinement checker (FDR) tool is a general purpose tool that can be used to determine whether an implementation refines a specification. In the case of protocol authentication, checking for refinement amounts to testing whether each trace of the implementation is also a trace of the specification. At first it has been used to analyse many sorts of systems, including distributed databases and communications protocols (Roscoe, 1993), but recently (Roscoe, 1995), (Lowe, 1996) it has been used to analyse security protocols as well.

Another theorem prover the Higher Order Logic (HOL) (Gordon, 1993) has been used by Snekkenes in (Snekkenes, 1995) for stating and proving properties of cryptographic protocols. In (Lichota, 1996) a tool named Convince is being developed to facilitate the modelling and analysis of cryptographic protocols. Convince uses a HOL theorem prover with automated support. This tool implements the Brackin version of belief logic, referred to as BGNY. Through HOL constructs, BGNY provides significant extensions to GNY: it allows goals to be specified at different protocol steps, not just after the protocol has completed. It also allows use of multiple algorithms for symmetric and public-key encryption and hashing, use of message authentication codes, computed values as keys, and use of key-exchange functions. The HOL BGNY theory contains the rules used in trying to prove that a protocol's initial conditions and messages indeed establish the protocol's goals.

All the above approaches described above have shown a good performance discovering attacks caused by lack of explicitness in the protocol messages. In spite of that they suffer from the fact that the state space under exploration can be very large. Additionally it remains questionable the effectiveness of these tools, in cases where the systems are of very large scale.

Although these methods have been judged as an important contribution to the field, research has turned into more specialised directions. The driving force behind this turn is the desire to use cryptography domain specific reasoning knowledge.

2.2 Algebraic simplification theoretic model methods

In these methods a protocol is modelled with a collection of rules for transforming and reducing algebraic expressions representing messages. Important methods in this category have been proposed by (Dolev, 1983), and (Meadows, 1992).

(Dolev, 1983) is the basic model for the state-machine approach. In this model the network is assumed to be under the control of an intruder who can read all traffic, create alter and destroy messages, and perform any operation that is available to legitimate user of the system. It is assumed that initially the intruder does not know any secret information such as encryption keys, belonging to honest users of the system.

Since the intruder can prevent any message from reaching its destination, and since he can also create messages of his own, it may be treated any message sent by an honest user as a message received from the intruder. In this way the system becomes a machine used by the intruder to generate words. These words obey certain rewrite rules, such as the fact that encryption and decryption with the same key cancel each other out. In this system the intruder perform as a term-rewriting system manipulator. The final task of the intruder is to discover a word that is meant to be secret. In this way the robustness of a protocol concerning security becomes a word problem in a term-rewriting system. This observation was used to develop several algorithms to analyse restricted classes of protocols in terms of their properties as term-rewriting system. In (Dolev, 1983) two models were developed, namely the cascade protocols, in which the users can apply cryptographic operations in several layers to form messages and the name-stamp protocols in which the users are allowed to append, delete, and check names encrypted together with the plaintext. A name-stamp protocol can also contain layers of encryption.

The Dolev-Yao model has the main drawback that it does not allow principals to remember state information from one state to the next, and it can only be utilised to detect failures of secrecy. So it was obvious that it was required an augmented approach.

The NRL Protocol Analyzer (Meadows, 1992) is an interactive program written in Prolog that can be used to assist either in the verification of security properties of cryptographic protocols or in the detection of security flaws. The NRL model takes the same approach as the term-rewriting model of Dolev-Yao. The main difference between the two models is that the Dolev-Yao model treats a protocol as a machine for producing words, while NRL Analyzer treats a protocol as a machine for producing not only words, but beliefs, and events as well. In NRL each participant in the protocol possesses a set of beliefs. These beliefs are created or modified as the result of receiving messages made up of words, while messages are sent depending upon both beliefs and messages received. Events represent the state transitions in which new words are generated and beliefs are modified. Thus an intruder who controls the dissemination of messages can use the protocol to produce words, beliefs, and events.

As in the case of Interrogator, one uses the tool to find protocol security flaws by specifying an insecure state and attempting to construct a path to that state from an initial state. Unlike Interrogator, an unlimited number of protocol rounds are allowed in a single path, so that the state space is infinite. This allows the NRL Analyzer to discover attacks that rely on the intruder's ability to weave several different runs of a protocol together. Also, unlike the Interrogator, the emphasis is not only on finding paths to insecure states, but on proving that these states are unreachable. This is made possible by having the user prove that certain paths leading backwards from the insecure state go into infinite loops, never reaching an initial state.

The NRL Protocol Analyzer has been used to locate a series of previously unknown flaws in a number of protocols (Simmons, 1985), (Burns, 1990), and to demonstrate flaws that were already known (Kemmerer, 1994). The current implementation's main drawback is the paucity of reduction operators which are limited to conventional and public key encryption operators. Another source of difficulty in using NRL Protocol Analyzer is in generating the lemmas, stating that infinite classes of states are unreachable, that are to be proved. In (Meadows, 1996) an effective procedure is presented for making this task easier by automating the generation of lemmas involving the use of formal languages. In addition, as with most rule-rewrite systems, it is not clear how well the system scales as more complicated algorithms will need to be expressed using an ever increasing set of rules.

2.3 Expert system, scenario based methods

The method due to (Millen, 1987) (Millen, 1995), known as the Interrogator Model, is one of the earliest systems used a Dolev-Yao approach. The Interrogator is a software tool - Prolog program, that incorporates a state-transition model for protocols. While the abstract model includes the usual state variable for the intruder's set of known items, the search algorithms expressed recursively uses a state representation with no explicit mention of the known set.

The Interrogator also has an equation-solving facility for terms using encryption and other operators used in authentication protocols. This facility called "generalised narrowing" implements a multiple-theory approach which handles commutative operators like exclusive-

or and others, such as a limited form of finite-field exponentation to which prior narrowing algorithms do not apply.

Protocol participants are modelled as communicating state machines whose messáges to each other are intercepted by an intruder who can either destroy messages, modify them or let them pass through unmodified. Given a final state in which the intruder knows some word which should be secret, the Interrogator will try all possible ways of constructing a path by which that state can be reached. Finally, if it finds such a path, then it has identified a security flaw.

The Interrogator model has not uncovered previously unknown attacks in well-known protocols, but it has been able to reproduce a number of already known attacks, as mentioned in (Kemmerer, 1994).

3 INFERENCE CONSTRUCTION TOOLS METHODS

Some years ago a formal logic model for the analysis of knowledge and belief, called BAN logic (Burrows, 1990) has been developed, and became the most widely used formal method for the analysis of authentication protocols. It assumes that authentication is a function of integrity and freshness, and uses logical rules to trace both of those attributes through the protocol.

There are three main stages to the analysis of a protocol using BAN logic. The first step is to express the assumptions and goals as statements in a symbolic notation, so that the logic can proceed from a known state so as to be able to ascertain whether the goals are in fact reached. The second step is to transform the protocol steps also in formulas in symbolic notation. Lastly, a set of deduction rules called postulates are applied. The postulates should lead from the assumptions, via intermediate formulas, to the authentication goals.

BAN logic has been a success. It has found flaws in several protocols, including Needham-Schroeder (Needham, 1978), CCITT X.509 (CCITT, 1988). It has uncovered redundancies in many protocols, including Needham-Schroeder, Kerberos (Millen, 1987), Otway-Rees (Otway, 1987), CCITT X.509. Many published papers use BAN logic to make claims about their protocol's security (Pal, 1996), (Shieh, 1996).

However successful, critiques of BAN logic on various features have been published. Nessett, in (Nessett, 1990), criticised BAN logic about its claimed goals of authentication. He constructed a specific example, in order to demonstrate the BAN logic's failure to discover flaws which violate security in a basic sense. Snekkenes, in (Snekkenes, 1991) examined the BAN logic's limitation of providing partial correctness proofs. Syverson in (Syverson, 1991) described some confusions about the BAN logic's goals and further explained a problem of informality in the BAN logic's operational semantics. For this reason, in (Mao, 1993) measures to make BAN logic formal are proposed. The proposed formalisation is found to be desirable, not only for its potential in providing rigorous analysis of security protocols, but for its readiness for supporting a computer-aided fashion of analysis as well.

But the most criticised point in BAN logic is the idealization step because of its ambiguity and vagueness, particularly where a message is idealized into a formula containing information not present in the message itself. Active research field for the BAN logic and other BAN-like logics, is the design and development of an efficient method for authentication protocol idealization. This method will be based on rule-based techniques and could have as a result to refine a big step of protocol messages transformation into several smaller, simpler and easier to understand. Thus, it will reduce the possibility of error occurrence in the informal protocol idealization steps. Furthermore it will be increased the ease of diagnoses of lower-level design flaws, for achieving the development of robust cryptographic protocols. An attempt to this direction, which works well is (Mao, 1995), but it does not cover protocols using public-key algorithms and not includes theoretic proof of the soundness of the idealization rules.

Other logic systems have been published, some designed as extensions to BAN logic (Gong, 1990), (van Oorschot, 1993), (Kessler, 1994) and others based on BAN to correct perceived weaknesses (Snekkenes, 1991), (Mao, 1993).

In (Gong, 1990) a successful approach is proposed named GNY logic. GNY logic aims to analyse a protocol step-by-step, makes explicit any assumptions required, and draw conclusions about the final position it attains. This logic offers important advantages over BAN logic. GNY approach places a strong emphasis on the separation between the content and the meaning of messages, which increases consistency in the analysis and introduce the ability to reason at more than one level. In GNY principals can include in messages data which they do not believe in, but just possess. It is possible to express the ability of a recipient to identify the messages he expects. It allows us to determine that certain messages are not replays of a recipient's own previous messages in a session.

Despite all these, GNY logic addresses only authentication, is much more complicated and elaborated, there are many rules which have to be considered at each stage, and has some drawbacks and shortcomings (Anderson, 1992). In (Brackin, 1996a) a HOL theory formalising an extended version of the GNY is described, which used by software that automatically proves authentication properties of cryptographic protocols.

In (Syverson, 1994) a SvO logic is presented, that it designed to capture the features of extensions and variants of four logics, namely BAN, GNY, AT (Abadi, 1991) and vO (van Oorschot, 1993) in a single unified framework. They have also presented a model-theoretic semantics with respect to which it is sound. The SvO logic is simpler to use than any of those from it is derived, and more expressive than any of them.

A recent logic is proposed in (Kailar, 1995) for the analysis of communication protocols that require accountability, such as those for secure electronic transactions. This logic looks at what can be achieved without making any assumptions about freshness. A set of postulates which are applicable to the analysis of proofs in general and the proofs of accountability in particular are proposed.

More recently, another logic for the analysis of authentication protocols is proposed in (Wedel, 1996), and a formal semantics is given for proving its soundness. This logic can handle a wide variety of cryptographic mechanisms using a minimum of notation. In this approach, the elimination of the formulas out of the idealized messages leads to a clear distinction between the protocol itself and the assumptions about it.

4 ROBUSTNESS PRINCIPLES

A complement approach is to try to encapsulate relative experience of good and bad practice into rules of thumb. The robustness principles are very helpful, in that adherence to them would have contributed to the simplicity of protocols and avoided a considerable number of published confusions and mistakes.

In (Anderson, 1996) a number of principles is proposed. More complete analyses of desirable protocol properties and relevant limitations can be found in (Abadi, 1994) and in (Syverson, 1996). Some of them are mentioned below:

- be very clear about the security goals and assumptions
- be clear about the purpose of encryption (secrecy, authenticity, etc.). Do not assume that its use is synonymous with security
- be careful that your protocol does not make some unexamined assumption about the properties of the underlying cryptographic algorithm
- be sure to distinguish different protocol runs from each other
- sign before encrypting; if a signature is affixed to encrypted data, then one cannot assume that the signer has any knowledge of the data; a third party certainly cannot assume that the signature is authentic, so nonrepudiation is lost
- where the identity of a principal is essential to the meaning of a message, it should be mentioned explicitly in the message
- if timestamps are used as freshness guarantees by reference to absolute time, then the difference between local clocks at various machines must be less than the allowable age of a message deemed to be valid; furthermore, the time maintenance mechanism everywhere becomes part of the Trusted Computing Base.
- do not assume that a message you receive has only a particular form, even if you can check this.

It is remarkable that, in many cases, following one design principle will sometimes lead to violating another. This is almost expected, since we have to deal with general rules of thumb. But we should not infer from the fact that we meet even all the ones we have that the result is a good design.

A widely accepted logical deduction is that, both formal proofs and structured design rules at the beginning, middle, and end of designing a protocol, are complement to achieving effective and reliable cryptographic protocols.

5 DESIGN PROCESS INTEGRATION ASPECTS

It has become evident that it was difficult for the analysts other than the developers themselves of the previous mentioned techniques to apply them. The main reason for this difficulty is the fact that the protocols had to be re-specified for each technique, and it was not easy to transform the published description of the protocol into the required formal system. Some tools are designed as translators towards performing the transformation automatically. The input to any such translator still requires a formally-defined language, but it can be made similar to the message-oriented protocol descriptions that are typically published. The main idea is the designing of a single common protocol specification language, that could be used as the input format for any formal analysis technique.

A much promised and effective approach is provided in (Brackin, 1997a) (Brackin, 1997b) (Brackin, 1996b); a simple Interface Specification Language (ISL) is specified and an Automatic Authentication Protocol Analyzer (AAPA) is described that automatically; either proves that specific protocols have their specific desired properties, or identifies precisely where these proof attempts fail. The AAPA produces its proofs using BGNY protocol analysis belief logic, implemented in the HOL family of proof tools. The AAPA can be used either alone or as part of the Convince system, which consists of the AAPA together with a GUI that automatically creates ISL specifications from user-created graphical protocol representations. The AAPA does not detect all possible protocol failures, but a large family of common errors. It has strengths and weaknesses as well, and it is proposed as one of the most effective tools for aiding in the design process.

Recently in (Millen, 1997) another promised integrated tool named Common Authentication Protocol Specification Language (CASPL), partly inspired by ISL, is developing by J.Millen. CASPL is proposed as a single common protocol specification language that can be used as the input format for any formal analysis technique, such as Prolog state-search analysis tools (Millen, 1995), the NRL Protocol Analyzer (Meadows, 1992), model-checking with FDR (Lowe, 1996) and HOL (Brackin, 1996a). The main objectives of the CASPL design are usability, abstraction, completeness, extensibility, parsability, and scalability. This work is in progress, has not yet been completed and published since the end of February 1997, and it is described in a WWW site for suggestions, refinement and standardisation of the language definition.

6 THE NEXT STEP: USING FORMAL METHODS FOR PROTOCOL DESIGN

There is no doubt that designing secure cryptographic protocols is a very difficult process. Until recently, researchers orientated to use formal methods to the analysis of existing protocols. These methods have proved successful at discovering flaws with existing protocols, sometimes previously unrecognised ones. Despite that fact, there remains a great deal of doubt as to whether any of the existing techniques is sufficient to provide a proof that a given protocol is correct. This situation has a fair analogy in testing process for general purpose computer programs, where reliable testing techniques allow many bugs to be found, but will not provide at all a complete proof of correctness. In any situation, it would be a prudent and mature trend, methods to be designed and tools to be implemented for the correct design of cryptographic protocols in the first place. The incorporation of formal methods into design can be implemented heretofore in various methods.

One approach (Meadows, 1995) is to develop specific methodologies for design of protocols so that they will be more amenable to analysis by formal methods. In (Heintze, 1994) they develop a modular approach to designing cryptographic protocols. They design a family of tools for reasoning about protocol security and prove a composition theorem which allow them to state sufficient conditions on two secure protocols, such that they may be combined to form a new secure protocol. Moreover, defining the secret-security and the time-security notions, they gave counter-examples in order to show that when the conditions are not met, the new protocol may not be secure.

More recently, design principles have been put forth for producing protocols whose security is easy to evaluate. In (Gong, 1995) it is proposed a new approach to designing secure protocols that is centred on a novel notion of fail-stop protocols. The main idea came from the work in (Schlichting, 1983) where they proposed the concept of a fail-stop processor, which, when failing, stops completely before any effect is visible to the outside world. A fail-stop protocol, which automatically halts in response to any active attack that interferes with protocol execution, thus reducing protocol security analysis to that of passive attacks only (i.e., eavesdropping). After that it is much easier to conclude whether the secrecy assumption can be violated.

The suggesting proof methodology for a fail-stop protocol concludes three phases: The verification that the protocol is fail-stop, the validation of the secrecy assumption and the apply of BAN-like logics. This proposed methodology must apply BAN-like logics because even for a fail-stop protocol, the residue from its execution may be useful to an attacker, e.g. as in (Nessett, 1990). Another encouraging point for this methodology, is that the specifications of fail-stop protocols satisfy some of the main prudent engineering principles from (Abadi, 1994) and (Syverson, 1996). Accordingly, if we use the GNY logic to analyse a fail-stop protocol, we can reduce dramatically the complexity of this logic. In (Gong, 1995), it is mentioned that their investigation showed that many existing protocols proved to be fail-stop, so that the new notions are not too limiting.

An alternative approach (Meadows, 1995) is a layered one, in which an abstract model is used at the top layer, and every succeeding layer is proved to be an implementation of the layer above it, until a detailed specification is produced. Much of the existing work on requirements specifications (Syverson, 1993) has this specific flavour. For the application of BAN Logic (Carlsen, 1994) it has developed a parser that will translate members of a limited class of protocol specifications into BAN Logic.

Another way (Boyd, 1994) is a technique to designing key exchange protocols which are guaranteed to be correct in the sense that a specified security criterion will not be violated if protocol principals act correctly. This technique is developed from basic cryptographic properties that can be expected to be held by a variety of cryptographic algorithms. Protocols can be developed abstractly and any particular type of algorithm that possesses the required property can then be used in a concrete implementation.

Furthermore, in (Gollmann, 1996) it is suggested that the design of authentication protocols has proven to be error prone, partly due to a language problem. The objectives of entity authentication are usually given in terms of human encounters while we actually implement message passing protocols. Within that paper, it is proposed various translations of the high level objectives into a language appropriate for communication protocols.

7 CONCLUSIONS

An overview of the modern trends in the application of formal methods for the analysis and design of cryptographic protocols is presented. All these models at the various levels of abstraction have their areas of usefulness. It is efficient to use the more abstract models at earlier points in the design stage, when implementation details have not been yet decided upon. In (Meadows, 1995) it is proposed for a protocol designer, as a first stage, to use a BAN-like logic to determine what the role of each message of a protocol should be. Then the designer may use a state-based tool, such as the NRL Protocol Analyzer or the Interrogator model, when attempting to determine what the structure of messages should be. Accordingly, when the implementation, in terms of formatting messages, encryption schemes, and integrity mechanisms, is in question, it would be very useful to use specific models to determine how these implementation decisions affect the security of the cryptographic protocol. For this specific purpose a useful operations model, presented in (Stubblebine, 1992), defines message integrity in terms of a condition which must be satisfied in every state of a protocol run.

Within this paper, some areas were emerged where further research is needed. A complement direction for research work is to investigate the potential integration of methods like NRL Protocol Analyzer and the Interrogator model, within the methodology of fail-stop protocols, in cases that there exist protocols like (Gong, 1993) which have requirements that conflict with some of the particular fail-stop requirements.

Concluding the review and surveying the recent research trends, it is obvious that the research community should work towards designing effective tools that take easy-to-write specifications of protocols and expected properties and quickly, in minutes, perform formal analyses checking for failures of these protocols to achieve their desired properties. AAPA and CAPSL seems to be the most promised approaches to bridge the gap between the typical information presentations of protocols given in research papers and the precise characterisations required to conduct formal analysis. Furthermore the researchers should work towards developing more effective techniques to design protocols that are guaranteed to be reliable and correct in the first place.

Acknowledgements
The authors would like to thank Dr. D.Spinellis, University of the Aegean Greece, and Dr. D.Gritzalis, Athens University of Economics and Business, Greece, for their helpful and insightful comments.

8 REFERENCES

Abadi M., Needham R., (1994) Prudent Engineering Practice for Cryptographic protocols, *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, pp. 122-136.

Abadi M., Tuttle M., (1991) A Semantics for a Logic of Authentication, *Proceedings of the Tenth ACM Symposium on Principles of Distributed Computing*, pp. 201-216, ACM Press.

Anderson R., (1992) A Second Generation Wallet, *ESORICS '92 Proceedings of the Second European Symposium on Research in Computer Security*, Springer Verlag, pp. 411-418.

Anderson R., Needham R., (1996) Programming Satan's Computer, *Lecture Notes in Computer Science LNCS 1000*, Springer Verlag, pp. 426-440.

Boyd C., Mao W., (1994) Designing Secure Key Exchange Protocols, *ESORICS '94, Proceedings of the Third European Symposium on Research in Computer Security*, Springer Verlag, pp. 93-105.

Brackin S., (1997a) An Interface Specification Language for Automatically Analyzing Cryptographic Protocols, *Proceedings of the 1997 Symposium on Network and Distributed System Security*, pp. 40-51, IEEE Computer Society Press.

Brackin S., (1997b) Automatic Formal Analyses of Cryptographic Protocols, private communication.

Brackin S., (1996a) A HOL Extension of GNY for Automatically Analyzing Cryptographic Protocols, *Proceedings of the 1996 IEEE Computer Security Foundations Workshop IX*, pp. 62-76, IEEE Computer Society Press.

Brackin S., (1996b) Automatic Formal Analyses of Cryptographic Protocols, *Proceedings of the 19th National Conference on Information Systems Security*, Baltimore, MD, IEEE.

Burns J., Mitchell C., (1990) A Security Scheme for Resource Sharing over a Network, *Computers and Security*, Vol. 19, pp. 67-76.

Burrows M., Abadi M., Needham R., (1990) A Logic of Authentication, *ACM Transactions on Computer Systems*, 8(1), pp. 18-36.

Carlsen U., (1994) Generating Formal Cryptographic Protocol Specifications, *Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 137-146, IEEE Computer Society Press.

CCITT (1988) *CCITT X.509*, The Directory - An Authentication framework, CCITT.

Denning D., Sacco G., (1981) Timestamps in Key Distribution Protocols, *Communications of the ACM*, Vol. 24, No. 8, pp. 533-536.

Dolev D., Yao A., (1983) On the Security of Public Key Protocols, *IEEE Transactions on Information Theory*, 29(2), pp. 198-208.

Gollmann D., (1996) What do we mean by Entity Authentication, *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pp. 46-54, IEEE Computer Society Press.

Gong L., Lomas T., Needham R., Saltzer J., (1993) Protecting Poorly Chosen Secrets from Guessing Attacks, *IEEE Journal on Selected Areas in Communications*, Vol. 11, No. 5, pp. 648-656.

Gong L., Needham R., Yahalom R., (1990) Reasoning about Belief in Cryptographic Protocols, *Proceedings of the 1990 IEEE Symposium on Security and Privacy*, pp. 234-248, IEEE Computer Society Press.

Gong L., Syverson P., (1995) Fail-Stop Protocols: An Approach to Designing Secure Protocols, *pre-Proceedings of DCCA-5 Fifth International Working Conference on Dependable Computing for Critical Applications*, pp. 45-55.

Gordon M., Melham T., (1993) *Introduction to HOL: A Theorem Proving Environment for Higher Order Logic*, Cambridge University Press, Cambridge, UK.

Gritzalis S. (1996) The BAN logic for the analysis of authentication protocols in distributed systems: A review, *In Proceedings of the 1st meeting of the IKAROS human network for the Security, Quality, and Reliability in Information & Communication Technologies* (in Greek).

Heintze N., Tygar J., (1994) A Model for Secure Protocols and their Compositions, *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, pp. 2-13, IEEE Computer Society Press.

Kailar R., (1995) Reasoning about Accountability in Protocols for Electronic Commerce, *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, pp. 236-250, IEEE Computer Society Press.

Kemmerer R., (1989) Analyzing encryption protocols using formal verification techniques, *IEEE Journal on Selected Areas in Communications*, 7(4), pp. 448-457.

Kemmerer R., Meadows C., and Millen J. (1994) Three Systems for Cryptographic Protocol Analysis, *Journal of Cryprology* (7), pp. 79-130.

Kessler V.;, Wedel G., (1994) AUTLOG-An advanced Logic of Authentication, *Proceedings of the 1994 IEEE Computer Security Foundations Workshop VII*, pp. 90-99.

Lichota R., Hammonds G., Brackin S., (1996) Verifying the Correctness of Cryptographic Protocols using Convince, *Proceedings of the 12th IEEE Computer Security Applications Conference*, pp. 117-128, IEEE Computer Society Press.

Lowe D., (1996) Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR, *In Proceedings of TACAS*, Springer Verlag, pp. 147-166.

Mao W., (1995) An Augmentation of BAN-like Logics, *Proceedings of the 1995 IEEE Computer Security Foundations Workshop VIII*, pp. 44-56, IEEE Computer Society Press.

Mao W., Boyd C., (1993) Towards formal analysis of security protocols, *Proceedings of the 1993 IEEE Computer Security Foundations Workshop VI*, pp. 147-158, IEEE Computer Society Press.

Meadows C., (1992) Applying Formal Methods to the Analysis of a Key-Management Protocol, *Journal of Computer Security*, vol. 1, pp. 5-35.

Meadows C., (1995) Formal Verification of Cryptographic Protocols: A Survey, *Advances in Cryptology, ASIACRYPT '94, Proceedings*, Springer Verlag, pp. 133-150.

Meadows C., (1996) Language Generation and Verification in the NRL Protocol Analyzer, *Proceedings of the 1996 IEEE Computer Security Foundation Workshop IX*, pp. 48-61, IEEE Computer Societ Press.

Millen J., (1997) Common Authentication Protocol Specification Language, *http://www.mitre.org/research/capsl.*

Millen J., (1995) The Interrogator Model, *Proceedings of the 1995 IEEE Symposium on Security and Privacy*, pp. 251-260, IEEE Computer Society Press.

Millen J., Clark S., Freedman S., (1987) The Interrogator: Protocol Security Analysis, *IEEE Transactions on Software Engineering*, Vol. 13, No. 2.

Millen J., Neuman C., Schiller J., Saltzer J., (1987) Kerberos Authentication and Authorization system, *Project Athena Technical Plan*, Section E.2.1. M.I.T., MA.

Needham R., Schroeder M., (1978) Using Encryption for Authentication in large networks of computers, *Communications of the ACM*, 21(12), pp. 993-999.

Needham R., Schroeder M., (1987) Authentication revisited, *Operating Systems Review*, Vol. 21, No 1, pp. 7.

Nesset D., (1990) A Critique of the BAN Logic, *ACM Operating Systems Review*, Vol. 24, No. 2, pp. 35-38.

Neuman B., Stubblebine S., (1993) A Note on the Use of Timestamps as Nonces, *ACM Operating Systems Review*, 27(2), pp. 10-14.

Oorschot van P. C., (1993) Extending Cryptographic Logics of Belief to Key Agreement Protocols, *Proceeedings of the First ACM Conference on Computer and Communications Security*, pp. 232-243.

Otway D., Rees O., (1987) Efficient and timely mutual authentication, *ACM Operating Systems Review*, 21(1), pp. 8-10.

Pal G. (1996) Verification of the iKP family of secure electronic payment protocols, *http://web.mit.edu/gnpal/www/ikp/verify_ikp.html.*

Roscoe, A.W., (1993) Developing and verifying protocols in CSP, *Proceedings of Mierlo workshop on protocols*, TU Eidhoven.

Roscoe, A.W., (1995) Modelling and verifying key-exchange protocols using CSP & FDR, *Proceedings of the 1995 IEEE Computer Security Foundations Workshop IIX*, pp. 98-107, IEEE Computer Society Press.

Satyanarayanan M., (1989) Integrating Security in a large distributed system, *ACM Transactions on Computer Systems*, 7(3), pp. 247-280.

Scheid J., Holtsberg S., (1988) *Ina Jo Specification Language Reference Manual*, System Development Group, Unisys Corporation, CA.

Schlichting R. D., Schneider F. B., (1983) Fail-Stop Processors: An Approach to Designing Fault-Tolerant Computing Systems, *ACM Transactions on Computing Systems*, Vol. 2, No. 2, pp. 222-238.

Shieh, S.P., Yang, W.H., (1996) An Authentication and Key Distribution System for Open Network Systems, *ACM Operating Systems Review*, Vol. 30, No. 2, pp. 32-41.

Sidhu D., (1986) Authentication Protocols for Computer Networks, *Computer Networks and ISDN Systems*, 11, pp. 297-310.

Simmons G. (1985) How to Selectively Broadcast a Secret, *Proceedings of the 1985 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press.

Stubblebine S., Gligor V., (1992) On Message Integrity in Cryptographic Protocols, *Proceedings of the 1992 IEEE Symposium on Security and Privacy*, pp. 85-104, IEEE Computer Society Press.

Snekkenes E., (1991) Exploring the BAN approach to Protocol Analysis, *Proceedings of the IEEE Computer Security Foundations Workshop IV*, pp. 171-181, IEEE CS Press.

Snekkenes E., (1995) *Formal Specification and Analysis of Cryptographic Protocols*, Ph.D. Thesis, University of Oslo, Norway.

Syverson P., (1991) The Use of Logic in the Analysis of Cryptographic Protocols, *Proceedings of the 1991 IEEE Computer Security Symposium on Security and Privacy*, pp. 156-170, IEEE Computer Society Press.

Syverson P., (1993) On Key Distribution Protocols for Repeated Authentication, *ACM Operating Systems Review*, 27(4), pp. 24-30.

Syverson P., (1996) Limitations on Design Principles for Public Key Protocols, *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pp. 62-72, IEEE Computer Society Press.

Syverson P., Meadows C., (1993) A Logical Language for Specifying Cryptographic Protocol Requirements, *Proceedings of the 1993 IEEE Computer Security Symposium on Security and Privacy*, pp. 165-177, IEEE Computer Society Press.

Syverson P., P.C. van Oorschot (1994) On Unifying some Cryptographic Protocol Logics, *Proceedings of the 1994 IEEE Computer Security Foundations Workshop VII*, pp. 14-29, IEEE Computer Society Press.

Varadharajan V., (1989) Verification of Network Security Protocols, *Computers and Security*, Vol. 8, pp. 693-708.

Wedel G., Kessler V., (1996) Formal Semantics for Authentication Logics, *ESORICS '96 Proceedings of the Fourth European Symposium on Research in Computer Security*, Springer Verlag, pp. 219-241.

BIOGRAPHIES

Stefanos Gritzalis holds a BSc in Physics and an MSc in Electronic Automation, both from the University of Athens, Greece. He is also pursuing a PhD degree on Distributed Systems Security, with the Department of Informatics of the University of Athens, Greece. Currently, he is an Assistant Professor with the Department of Informatics of the Technological Educational Institute (TEI) of Athens, Greece. His research interests include Distributed Systems, Computer Systems Security, and Operating Systems.

Nikitas Nikitakos holds a B.Sc. in Naval Engineering from Hellenic Naval Academy, an M.Sc. in Applied Mathematics and an M.Sc. in Electrical Engineering both from the Naval Postgraduate School, California, USA, and a Ph.D. in Electrical and Computer Engineering from the National Technical University of Athens, Greece. Currently he is an active Commander in Hellenic Navy. His research interests include Radar and Sonar Theory and Techniques, Computer Systems Security, and Communication Systems.

Panagiotis Georgiadis holds a B.Sc. in Physics from the University of Athens, Greece, an M.Sc. in Computer Science from the Warwick University, UK, and a Ph.D. in Computer Science from the University of Athens, Greece. Currently he is an Associate Professor with the Department of Informatics, University of Athens, Greece. His research interests include Distributed Systems, Simulation, Operating Systems, and Computer Systems Security.