Logics for Cryptographic Protocols - Virtues and Limitations

V.D. Gligor, R. Kailar, S. Stubblebine, and L. Gong*

Department of Electrical Engineering
University of Maryland, College Park, MD 20742

Abstract

In this note we offer a perspective on the virtues and limitations of several logics for cryptographic protocols focusing primarily on the logics of authentication. We emphasize the scope limitations of these logics rather than their virtues because (1) we consider their virtues to be better understood and accepted than their limitations, and (2) we hope to stimulate further research that will expand their scope.

1 Introduction

Advances in the formal analysis of authentication protocols, based primarily on the logic of authentication of Burrows, Abadi, and Needham [2,4], have stimulated interest in the development of new logics for analysis of other aspects of cryptographic protocols, such as message-meaning recognition [9,10] and message secrecy or privacy [6], not just for authentication. The use of these logics can be hampered, to some extent, by the difficulty of delimiting their usefulness in practical applications. The limited application scope of these logics can be easily misunderstood, thereby raising unreasonable expectations despite warnings to the contrary from the authors of these logics (e.g., [4]) and significant debate (e.g., [5,15]); at the same time, some important advantages of these logics are ignored, possibly due to insufficient application experience.

In this note, we offer a perspective on the virtues and limitations of the logics presented in references [2,4,9,10], and more recently in [1], based on preliminary attempts to use them in the analysis of practical cryptographic protocols. (Our focus on these logics is motivated exclusively by the fact that they are the best-known logics for cryptographic-protocol analysis to date.) We revisit some of the assumptions presented in the original articles on these logics to illustrate their impact. We also illustrate some scope limitations of these logics not reviewed before. We devote substantially more space to the presentation of limitations than to that of virtues because we consider the virtues of these logics to be better understood and accepted than their limitations. We expect that future research will enlarge the scope of these logics and enhance their applicability.

*Li Gong is with ORA Corporation, 301A Dates Drive, Ithaca, NY 14850. He is also a Visiting Scientist with the Department of Computer Science, Cornell University, Ithaca, NY 14853.

2 Virtues

The logics of cryptographic protocols in general, and those of authentication in particular, have four important virtues. First, they help formalize reasoning about useful abstract properties of cryptographic protocols; second, they force designers to make explicit security assumptions that might otherwise be buried in implementation details; third, they lead to stable belief formulas; [1] and fourth, they achieve a reasonably well-defined set of analysis goals.

2.1 Formal Reasoning

In the security area, formal reasoning is important because it aids early discovery of design flaws, thereby helping to prevent serious security breaches. Whenever the initial assumptions used in the design of a program or protocol are satisfied, the security properties under consideration can be verified only if there are no design errors within that program or protocol. Thus, significantly more assurance can be provided for the security-sensitive areas of system design than that made possible by testing.

The importance of formalized reasoning in the design of distributed system security cannot be overestimated given the susceptibility of these systems to attacks not present in centralized systems. Intruder attacks in distributed systems surpass the effectiveness of those in

[1] An exception is the reformulated logic of authentication presented in [1].

centralized systems because, among other reasons, administrative security measures can seldomly be used to compensate for security weaknesses. For example, system isolation through administratively imposed "air gaps" is clearly not useful in distributed systems. Consequently, threat models of intruder behavior must assume intruder access to and generation of messages exchanged in a distributed system, including those containing secrets, and must account for the possibility that the intruder can be, in fact, a legitimate user of the distributed system.

That formal reasoning considerations apply equally well to centralized systems, has led the naive to believe that the primary difference between formal reasoning in distributed- and centralized-system security is merely that of the complexity of cryptographic algorithms used in encryption/decryption operations. An important contribution to understanding the fallacy of such arguments has been provided Burrows, Abadi, and Needham in [2,4], where it is conclusively demonstrated that serious design errors can arise in authentication protocols regardless of the type, or strength, of the underlying cryptographic algorithm and the message-integrity measures used.

A related contribution of the work reported in references [2,4,10] is the demonstration of the utility of many-sorted modal logics in cryptographic-protocol analysis. This suggests (1) that either new analysis tools based on modal logic should be designed for use in the design of cryptographic protocols, or (2) that, if the axioms and inference rules of these logics can be "interpreted" in models of existent tools without loss of reasoning power for specific protocols, existing tools could be used profitably for the analysis of (at least, some) cryptographic protocols. An example of the latter approach for some of the inference rules of the BAN logic [2,4] is provided in [8] where an informal "interpretation" of these rules within the Ina-Jo language is illustrated. Although the generality of this approach remains an open issue, it appears that the simplicity of the BAN logic inference rules makes such interpretation possible at least for some protocols. If this approach proves to be general, the practical usefulness of these logics could be significantly enhanced. (An interesting exercise would be to attempt an interpretation of the more complex rules of logic presented in [10] and of the streamlined version of the logic of authentication presented in [1] within the computation models of existing tools.)

2.2 Explicit Security Assumptions

The use of logics for cryptographic protocols force designers to state explicitly the security assumptions made in protocol specifications. For example, the BAN logic requires the designer to state the trust assumptions in terms of the secrecy of the shared key (i.e., the shared key is good if it can be discovered only by the two parties involved in mutual authentication and by other parties trusted by either of them). This helps provide subsequent communication privacy, even though secrecy of shared keys may not be necessary for authentication. Similarly, the BAN logic also makes explicit the role of a distinguished third party that is trusted to generate quality keys in authentication protocols. Without asserting the jurisdiction of this third party over key generation, the quality of an encryption key cannot be easily ascertained.

Although the reformulated logic of authentication presented in [1] removes key secrecy as a requirement of key sharing for authentication, the key-secrecy assumption is important in cryptographic protocols wherever communication privacy is essential. Thus, the assumption of key secrecy must be made explicitly whenever necessary. In contrast to the key-secrecy requirement, which may only hold in some protocols, the requirement for ascertaining the freshness of message contents is present in all cryptographic protocols. Reasoning with the logics for cryptographic protocols forces designers to determine whether the exclusive use of timestamps as nonces is an adequate design decision in specific environments of protocol use. For example, in environments where time monotonicity cannot be guaranteed (e.g., in some public workstations), the adequacy of using timestamps as nonces is questionable.

It must be noted that the need for explicit security assumptions, or for specific security policies, does not imply that these assumptions or policies must be coupled with specific inference rules of a logic. Instead, the logic must ensure that, whenever the stated assumptions or policies are required but not satisfied by a protocol, the use of the logic's inference rules would fail to achieve desired, but unsupported, conclusions.

2.3 Stability of Belief Formulas

With the notable exception of the logic presented in reference [1], all other logics referred to in this paper lead to stable beliefs. This means belief formulas remain true for the duration of the protocol run after they become true. This is both a virtue and a potential limitation. It is a virtue because stable formulas make it easier to provide formal semantics for a logic and to prove its soundness and completeness with respect to that semantics. It is a potential limitation because belief negation may occasionally be useful. For example, abandoning stability enables one to write axioms using implication, which results in a more elegant, if not more powerful, semantics [1]. Furthermore, it may sometimes be desirable to negate beliefs. For example, assume that the vulnerability of a key increases with the number of messages in which it is used for encryption or checksum computation. To decrease security exposures, it might be required to impose a key-lifetime limit, which would be determined either by a lifetime function or by a real-time, key-change message. A key whose lifetime has expired could be used by the two parties to generate the next key bilaterally, without using a third-party authentication protocol. In such cases key beliefs may be negated during an application-protocol run since the old keys must be destroyed once the new ones are generated. Whether stability of formulas become a compelling limitation in the application of the logics to system analysis remains to be determined by practical experience.

2.4 Well-Defined Goals

The logics of cryptographic protocols achieve reasonably well-defined goals. For example, they help verify whether various parties obtain beliefs about the quality of keys in those instances where key jurisdiction is required. They explicitly tie the evolution of beliefs to message contents, number of messages and message rounds, helping determine the minimum number of messages needed to achieve a certain set of beliefs. For example, the use of the BAN logic shows that the Otway-Rees protocol [16] fails to achieve the same set of beliefs as that of the Needham-Schroeder(shared-key) protocol [14]. Thus, the Otway-Rees protocol, which requires only four messages, cannot be considered more "efficient" than the Needham-Schroeder protocol, which requires five messages. These logics can also help eliminate some protocol and message-field redundancy [2,4].

The logics of authentication are able to achieve their primary mission, namely the demonstration of mutual authentication for two parties and the freshness of the cryptographic key shared by the two parties. The incontestable importance of this can only be appreciated in the context of the large numbers of protocols that fail to satisfy these simple goals in the face of typical intruder attacks.

To achieve their goals, however, all logics for cryptographic-protocol analysis require protocol "idealization." That is, a protocol description must be provided in a suitable form for analysis. In and of itself, this requirement is neither new nor novel. Formal verification methods require idealized formal versions of (abstract) programs instead of informal specifications. As with all other formal methods, the analysis of a protocol is as good as its "idealized" abstract specification. However, the idealization step for cryptographic protocols is somewhat more complex than that necessary for writing formal specifications. A reason for this is that techniques for the modular composition of idealized specifications are currently unavailable.

The users of logics for cryptographic protocols should be aware of the pitfalls of the idealization process. Seemingly trivial omissions of protocol message contents may render an otherwise correct proof incorrect with respect to some of its assumptions. For example, the idealized versions of the Needham-Schroeder (shared-key) and Kerberos [13] protocols exclude the first message of the informal protocol specifications [2-4]. The first message of the Needham-Schroeder (shared-key) protocol contains a cleartext nonce (which is unnecessary in the public-key version of that protocol). Similarly, the first message of the Kerberos protocol includes a cleartext timestamp, which is inadvertently dropped from the informal specification of the protocol. This timestamp is considered to be monotonic and, therefore, can act as a nonce [12,13]. Excluding the nonce and the timestamp along with the first message of the these protocols could render their analysis incorrect because, as explained in reference [14], the absence of a nonce in the first message enables an intruder to force the reuse of an old key between the two parties interested in mutual authentication.

The GNY logic [10] uses protocol parsing to reduce dependency on the 'man in the loop' in the idealization process. Although GNY neither provides guarantees for the correctness of the parser nor completely eliminates the need for manual interpretation, it aids the designer in producing the idealized version of a protocol.

3 Scope Limitations

The primary mission of the logics of authentication, namely that of verifying whether a protocol delivers mutual authentication between two parties and distributes fresh keys of known quality, helps delimit the scope of these logics fairly precisely. Any use of these logics beyond this point can be hazardous and, therefore, must be carefully scrutinized. The operational assumptions required by the use of all logics for cryptographic-protocol analysis - not just those for authentication - also help delimit the scope of these logics. Any use of these logics in environments where operational assumptions, such as those of message integrity, are not satisfied could yield undesirable results. All limitations presented here are limitations of scope rather than of logics in general.

3.1 Level of Abstraction versus Properties of Cryptographic Mechanisms

The level of abstraction at which the logics of cryptographic protocols are intended to function is significantly higher than that necessary to determine whether message-integrity assumptions are satisfied within a given protocol. The use of the BAN logic assumes message integrity in the sense that an encrypted (or signed) message cannot be altered or pieced together from smaller encrypted (or signed) messages. The use of the GNY logic assumes message integrity also in the sense that every bit of ciphertext must depend on all bits of the cleartext and the key in such a way that any change to the cleartext causes a random change in the ciphertext and vice-versa. The purpose of revisiting these assumptions is to illustrate their implications, not to question their validity or soundness.

Recall that the DES Cipher Block Chaining (CBC) mode can provide data privacy and integrity at the block level, but not necessarily across multiple blocks [19]. Let us consider a hypothetical implementation of the Otway-Rees protocol with DES CBC. (Note that this deliberately naive implementation is recommended neither by the protocol authors nor by us.) The Otway Rees protocol can be summarized as follows:

Message 1. $A \rightarrow B : M, A, B, \{N_a, M, A, B\}^{K_{as}}$

Message 2. $B \rightarrow S : M, A, B, \{N_a, M, A, B\}^{K_{as}}, \{N_b, M, A, B\}^{K_{bs}}$

Message 3. $S \rightarrow B : M, \{N_a, K_{ab}\}^{K_{as}}, \{N_b, K_{ab}\}^{K_{bs}}$

Message 4. $B \rightarrow A : M, \{N_a, K_{ab}\}^{K_{as}}$

Suppose that some principal C conducts an active attack with the following scenario:

1. Principal C intercepts *Message 1* and creates the string: $M, \{N_a, M_1\}^{K_{as}}$, where M_1 represents the resulting cleartext when M is truncated at the end of a cipher block. M_1 has the same size as that of K_{ab}. (Fixed cipher-block lengths allows us to determine easily cipher block boundaries.) Note that since neither principal B nor server S expects a message from A, Messages 2 and 3 are effectively eliminated from the protocol.

2. Principal C relays to A: $M, \{N_a, M_1\}^{K_{as}}$. The protocol effectively becomes:

Message I. $A \rightarrow C : M, A, B, \{N_a, M, A, B\}^{K_{as}}$

Message II. $C \rightarrow A : M, \{N_a, M_1\}^{K_{as}}$

Since M is unencrypted, C knows the value of M and, thus, can masquerade as B for a complete session. (Other similar message-splicing attacks can be imagined for other naive protocol implementations.) Because the BAN and GNY logics assume that the message integrity is preserved (i.e., encrypted message portions cannot be spliced to obtain composite messages), they do not detect vulnerabilities of a protocol to message-splicing attacks. These attacks can be countered by using cryptographic checksums that capture the required (assumed) message-integrity properties. In some protocols, a simple rearrangement of the message fields is sufficient to counter splicing attacks thereby avoiding use of application-level checksum algorithms in addition to those already used by encryption. This suggests that the analysis of message integrity measures can be a significant exercise in its own right.

3.2 Jurisdiction Trust

The logics of authentication help establish, among other properties, the first and second level beliefs; i.e., beliefs of the form

$$A \;\; \textbf{believes} \; F$$
$$A \;\; \textbf{believes} \;\; B \;\; \textbf{believes} \; F.$$

The derivation of first-level beliefs in BAN logic is dependent on the notion of jurisdiction; i.e., the beliefs of the form A **believes** F (where F is some formula such as $A \overset{K_{ab}}{\longleftrightarrow} B$) require that A receives a message containing F, encrypted with a key which A shares with a principal that is trusted with the truth of statement F. Jurisdiction implies that a certain party of a protocol has designated authority over some statement F, such as that regarding the quality of a shared key, thereby assuring the validity of that statement. (Of course, jurisdiction is a form of trust in the sense that assurance evidence must be obtained by means unrelated to the logics to justify or rationalize assertion of jurisdiction).

The dependency of the first-level beliefs on jurisdiction has the consequence that protocols which do not make use of jurisdiction, but nevertheless can establish first-level beliefs in key sharing, cannot be easily analyzed. For example, in the unmodified Yahalom protocol [2,4], a party can make use of a key before it derives any belief in the quality of that key. Although the first-level beliefs in key sharing can be derived by the two communicating parties of this protocol, the application of the BAN logic cannot yield first-level beliefs without the assumption that the party which conveys a message including a key has jurisdiction over the freshness of that message. In short, the BAN logic cannot handle easily protocols that allow use of uncertified keys. The simple modification of the Yahalom protocol presented in references [2,4], which establishes message freshness without changing the protocol's message count, enables the derivation of first-level beliefs (and produces a certified-key protocol).

In other protocols, which may assume different types of key jurisdiction than those of BAN, only second-level beliefs can be derived using BAN, even though these protocols can establish first-level beliefs [8]. For example, a principal may be trusted to *read* and *convey* a statement about a key issued by another principal who has jurisdiction over that statement (key), even though the principal may not have jurisdiction to issue the statement itself. The recipient of this statement(key) should be able to infer that the statement (key) is authentic and that it is shared with the conveying principal. However, in such cases, analysis using the BAN logic would yield only weaker beliefs (second-level beliefs) since this type of trust assumption does not satisfy the definition of jurisdiction required to derive first-level beliefs using BAN logic.

Furthermore, in protocols that assume different types of key jurisdiction than those of BAN, belief ordering is not always preserved; e.g., a principal may incorrectly obtain a belief that another principal has derived its first-level belief [11]. The significance of deriving only second-level beliefs and of lack of belief ordering can be debated. For example, it might be possible to transform a given protocol, which does not exhibit key jurisdiction as defined in BAN, into another protocol that exhibits key jurisdiction and satisfies the same goals as those of the original protocol in the same number of messages. If this could be done for all protocols that do not have jurisdiction, then perhaps the dependency of first-level key beliefs on key jurisdiction will not be a practical limitation. It is worth noting, however, that in the logics presented in references [1,9,10], the derivation of the first-level beliefs is not explicitly dependent on key jurisdiction.

3.3 Honesty

The BAN and GNY logics assume implicitly honesty. This means that whenever a party states a formula (e.g., a shared key) it believes, or it has derived belief, in that formula. For example, a party asserting the freshness of a message it is assumed to believe in that assertion [2, 4, 10]. Gong introduces an explicit "eligibility" rule that deals with infeasible protocol specifications [9], which also prevents dishonesty; e.g., it prevents the forwarding of a logical formula not believed by the forwarding party. Note that such a rule does not necessarily prevent some parties from cheating at run time.

In contrast with the eligibility rule, and citing the relativity of the notion of honesty (e.g., a protocol that is honest in one context may not be honest in other contexts), Abadi and Tuttle remove the honesty assumption from the axioms of the logic for authentication (without providing an alternate honesty axiom or rule [1]). We provide below an example of forwarding which, as suggested by Abadi and Tuttle in reference [1], illustrates the need for decoupling honesty from the inference rules of the logics of authentication. The example also shows that the lack of honesty can lead to the derivation of unordered beliefs in much the same way as the lack of key jurisdiction. This suggests that explicit honesty axioms or rules, such as the eligibility rule, would be useful for similar reasons as those justifying the explicit retention of key jurisdiction.

Example of Authentication Forwarding (without belief in the quality keys)

In this example we present a simple protocol in which the 'honesty' assumption of references [2,4] does not hold. Suppose that in a Kerberos V like system, a principal A requests a forwarded ticket-granting ticket ($TGT_{B/C}$) from the ticket-granting server (TGS), which is valid for use by two principals B and C. After obtaining both the $TGT_{B/C}$ and the response containing the session key K_{A-TGS} from the TGS, A forwards the ticket to B. Principal B can now forward the ticket to C, whenever it finds this necessary. Of course, A must pass the accompanying session key K_{A-TGS} to B, and B must pass this key on to C for, otherwise, the $TGT_{B/C}$ containing that key would be unusable by B, and later on by C. Note that B forwards the $TGT_{B/C}$ and session key to C though it does not necessarily have any belief in the quality of the session key.

Protocol Description

Message 1: $A \rightarrow TGS$: Request for forwarded TGT valid at B and C

Message 2: $TGS \rightarrow A$: $\{TGT_{B/C}\}^{K_{TGS}}, \{...K_{A-TGS}...\}^{K_A}$

Message 3: $A \rightarrow B$: $\{TGT_{B/C}\}^{K_{TGS}}, \{...K_{A-TGS}...\}^{K_{A-B}}$

Message 4: $B \rightarrow C$: $\{TGT_{B/C}\}^{K_{TGS}}, \{...K_{A-TGS}...\}^{K_{B-C}}$

The analysis of this protocol using the BAN logic would result in the following types of beliefs for the principals (assuming that the principals believe in the freshness of these messages and in the $TGS's$ jurisdiction over the $TGT_{B/C}$):

$$A \textbf{ believes } (B/C \overset{\mathrm{K_{A-TGS}}}{\longleftrightarrow} TGS)$$

$$B \textbf{ believes } A \textbf{ believes } (B/C \overset{\mathrm{K_{A-TGS}}}{\longleftrightarrow} TGS)$$

$$C \textbf{ believes } B \textbf{ believes } (B/C \overset{\mathrm{K_{A-TGS}}}{\longleftrightarrow} TGS)$$

Since B forwards the message containing the key to C without necessarily obtaining any belief about the quality of the key, the belief that C obtains is not justified unless B has belief in A's jurisdiction on the key. Moreover, C's derived belief does not preserve belief ordering.

Application of Gong's eligibility rule to the analysis of this protocol would allow the ticket forwarding in Message 4 since B could posses the $TGT_{B/C}$ (B can receive the actual bits in encrypted form in Message 3). However, this rule would not allow B to associate any of its beliefs with the ticket since B would merely be relaying a message. Thus, C would be prevented from inferring that B has any confidence in the ticket and that the ordering of beliefs is preserved.

3.4 Privacy Trust

Privacy trust can be defined in terms of two related security requirements: (1) key secrecy, and (2) knowledge of the identities of all parties sharing a secret key. First, two parties (e.g., a client and a server) that wish to communicate with each other privately may use a shared *secret* key, which helps ensure that the contents of communication remain private to the two parties and whoever else either of them trusts. Secret keys are typically shared on long terms between users and authentication servers or between authentication servers, and on short terms between processes that establish private communication sessions. The logics for cryptographic protocols support key secrecy in different ways. The BAN, GNY and G [9] logics assume that shared keys remain secret to the communicating parties and parties they trust. The reformulated BAN logic presented in [1] removes the secrecy property from the definition of shared keys since this property is unnecessary for authentication. (However, secrecy axioms may need to be introduced whenever necessary.)

Second, *the identity of all parties* sharing the secret key should be known to the communicating parties. These identities are known at the outset of a cryptographic protocol explicitly or implicitly, as part of the initial trust relations assumed, or can be derived during the evolution of the protocol.

In typical intra-domain authentication protocols, the identity of any party who is entrusted with the shared key by either of the communicating parties is implicitly assumed to be known at the outset of the protocol. For instance, if party A of the Kerberos forwarding example of Section 3.3, forwards its identity and the shared secret key (K_{A-TGS}) to party B, and party B forwards $A's$ identity and the shared secret key to C, parties B and C are implicitly assumed to be known to other parties (e.g., to the TGS) as A. (Note that Kerberos forwarding requires a party such as C to know the identifier of party whose identity it takes on – not just that of the forwarding party B. That is, C must know A's identifier, since $C's$ authenticators for various services would have to include $A's$ identifier whenever C takes on $A's$ identity [12].)

However, making implicit assumptions of principal identities does not mean that all parties can discover the identity of all other parties sharing a key during a protocol run. For instance, if the forwarded $TGT_{B/C}$ of the example in Section 3.3 would be valid at D also (i.e., if A would request $TGT_{B/C/D}$ from the TGS), and if C would pass $TGT_{B/C/D}$) on to D, the identity of B would not necessarily be known to D. Thus, D would not know all parties who share the session key K_{A-TGS} with it. The logics discussed here either cannot easily analyze identity-forwarding protocols or, if they allow such protocol analysis, do not help detect gaps of identity knowledge in these protocols.

In typical inter-domain authentication protocols, all intermediate authentication servers of the domains traversed between the two communicating parties either know the shared (secret) key used by those parties [7] or can discover it after some work [12]. Thus, both parties must trust each intermediate authentication server with maintaining the secrecy of the shared key. This type of trust differs from the key-jurisdiction trust discussed in Section 3.2 in the sense that the quality of the shared key need not be at issue in the case of the intermediate authentication servers. Nevertheless, as in the case of key-jurisdiction trust, the trust relationships between both communicating parties and each intermediate authentication server must be explicitly defined in the idealization step.

The idealization of the trust relationships in inter-domain authentication is non-trivial because, in general, inter-domain authentication protocols impose a certain structure on the path of domains and authentication servers to be traversed. This structure is defined by various security policies. For example, these policies may define trust hierarchies of domains and authentication servers [7,12]. More general policies may introduce "peer links" between disjoint hierarchies of domains and authentication servers [17]. Domain traversals across peer links must be restricted according to a specific policy, which may differ substantially from those based on trust hierarchies. Typically, domain-traversal policies require that communicating parties be able both to identify the traversed domains and to discover the structure of the traversed path of domains (e.g., domain ordering). The idealization step of the logics discussed in this note needs to be enhanced to allow the definition of the trust relationships and authentication paths using explicit policies. A preliminary attempt to expand the scope of the logics for authentication to inter-domain authentication is reported in reference [11].

3.5 Formal Semantics

The need for formal semantics appears for at least three practical reasons. First, a formal semantics helps separate implicit assumptions from axioms and inference rules by using properties required to demonstrate the soundness of a logic [1]. For example, a formal semantics could elucidate the implicit assumptions of privacy trust in logics of authentication, could cause the separation of jurisdiction trust from the derivation of first-order beliefs and, as shown in reference [1], could cause the separation of the honesty assumption from message freshness beliefs. Second, a formal semantics helps delimit the scope limitations of a logic by elucidating what may *not* be derived using the logic, as opposed to the logic itself, which determines what can be derived with the logic. (Thus, formal semantics for all the logics for cryptographic protocols, could limit the need for papers such as this one.) Third, it is often easier and more natural to reason in the semantics during the analysis of a protocol than in the logic itself [18]. To take advantage of this, however, the logic must be sound (but need not always be complete) with respect to the semantics.

The definition of the GNY and G logics do not include formal semantics. Several limitations of the formal semantics of the BAN logic are explained and remedied in [1].

Scope Limitations-Summary

Property		BAN90	GNY90	G91	AT91
Message Integrity		Required	Required	Required	Required
Semantics		Formal	Operational	Operational	Improved Formal
Protocol Idealization		Required	Required/ Parser provided	Required/ Parser provided	Required
Jurisdiction Trust	Explicit Coupling with First-level beliefs	Coupled	Not coupled	Not coupled	Not coupled
	Belief ordering in the absence of Jurisdiction	No	No	Yes	Yes
Honesty	Explicit Coupling with Inference rules	Coupled	Coupled	Not Coupled (Specific rule included)	Removed(no specific axiom or rule included)
	Belief ordering in the absence of honesty	No	No	Yes	Yes
Privacy Trust	Key Secrecy (for Comm. Privacy)	Assumed	Assumed	Assumed	Not assumed(no specific axiom or rule included)
	(Ordered) Accumulation of principal identities	Not Supported	Not Supported	Not Supported	Not Supported

Acknowledgements

Whatever clarity this paper has is due in no small measure to comments received from Martin Abadi and Roger Needham. Discussions with John McLean, Catherine Meadows, and Paul Syverson helped clarify some issues of formal semantics.

References

[1] M.Abadi and M.Tuttle, "A Semantics for a logic for Authentication (Extended Abstract)" to appear in Proc. of the ACM Symposium of Principles of Distributed Computing, January 1991.

[2] M.Burrows, M.Abadi, and R.Needham, " A Logic of Authentication," in Proceedings of the 12th ACM Symposium on Operating Systems Principles, Litchfield Park, Arizona, December, 1989. Published in *ACM Operating Systems Review,* Vol.23, No.5, December 1989; a full version appears in DEC System Research Center report No.39, Palo Alto, California, February, 1989.

[3] M.Burrows, M.Abadi, R.M.Needham, " Logic of Authentication," Proceedings of the Royal Society of London A, Vol.426, 1989.

[4] M.Burrows, M.Abadi, R.M.Needham, "Logic of Authentication," *ACM Transactions on Computer Systems,* Vol.8, No.1, February, 1990; a full version appears as a DEC System Research Center report No. 39, (revised) February 1990.

[5] M.Burrows, M.Abadi, R.M.Needham, "Rejoinder to Nessett," *Operating Systems Review,* vol. 24, no. 2, 1990.

[6] P. Bieber, "A Logic of Communication in Hostile Environment," Proceedings of The Computer Security Foundations Workshop III, Franconia, N.H., June 1990.

[7] A.D. Birrell, B. W. Lampson, R. M. Needham and M.D. Schroeder " A Global Authentication Service without Global Trust," in the Proceedings of the IEEE Symposium on Security and Privacy, Oakland, California, 1986.

[8] P.-C. Cheng and V.D. Gligor, " On the formal Specification and verification of a Multiparty Session Protocol," Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, May 7-9, 1990, Oakland, California.

[9] L. Gong, "Cryptographic Protocols for Distributed Systems," PhD dissertation, University of Cambridge, April, 1990. Also see "Handling Infeasible Specifications of Cryptographic Protocols" in these Proceedings.

[10] L. Gong, R. Needham, and R. Yahalom, "Reasoning about Belief in Cryptographic Protocols," in Proceedings of the IEEE 1990 Symposium on Security and Privacy, Oakland, California, May, 1990.

[11] R. Kailar and V.D. Gligor, "On Belief Evolution in Authentication Protocols," these Proceedings.

[12] J.Kohl, C.Neuman, J. Steiner, "Kerberos Version 5 RFC, draft 2," MIT Project Athena, November 1989.

[13] S.P.Miller, C.Neuman, J.I.Schiller, and J.H.Saltzer. " Kerberos Authentication and Authorization System," Project Athena Technical Plan, Section E.2.1, MIT, July 1987.

[14] R.M.Needham and M.D.Schroeder. "Using Encryption for Authentication in Large Networks of Computers," *CACM* Vol.21, No. 12, December 1978.

[15] D.M. Nessett, "A Critique of the Burrows, Abadi, and Needham Logic," *Operating Systems Review,* vol. 24, no. 2, 1990.

[16] D.Otway and O.Rees, " Efficient and Timely Mutual Authentication," *Operating System Review* Vol.21, No.1, January 1978.

[17] J.Pato, "DCE Authorization Services" - Privilege Server, Cooperative Computing Division, Hewlett-Packard Co., Chelmsford, MA 01824, December, 1990.

[18] P.Syverson, "The Use of Logic in the Analysis of Cryptographic Protocols," Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, California, May 1991.

[19] V.L.Voydock, S.T.Kent. "Security Mechanisms in High-level Network Protocols," *Computing Surveys*, Vol 15. No.2, 1983.