

# Modelling the Relative Strength of Security Protocols

Ho Chung  
University of Southern California  
941 W. 37th Place  
Los Angeles, USA  
hochung@usc.edu

Clifford Neuman  
University of Southern California  
941 W. 37th Place  
Los Angeles, USA  
bcn@isi.edu

## ABSTRACT

In this paper, we present a way to think about the relative strength of security protocols using SoS, a lattice-theoretic representation of security strength. In particular, we discuss how the model can be used, present the TLS protocol as a compelling real world example, show how it is modeled, and then explain how lattice-theoretic properties can be used to evaluate security protocols.

## Categories and Subject Descriptors

H.1 [Models and Principles]: Miscellaneous; D.4.8 [Software Engineering]: Performance—Measurements, Modeling and prediction

## General Terms

Security

## Keywords

Cryptographic protocols, Relative strength, Lattice, Comparison

## 1. INTRODUCTION

In this paper, we present a security model for measuring the *relative strength of security* in cryptographic protocols. Existing research has focused on proving the *correctness* of the security protocols assuming perfect cryptography [12, 17, 1, 10, 19, 20, 4]. Very little work has been done on determining the relative strength of security protocols. One possible reason is that it is difficult to quantify. The following are two naive examples where Bob authenticates Alice based on a shared secret  $k_{AB}$  in both protocols. Protocol 1 and 2 have the same objective, unilateral authentication, and have very similar designs. Although both protocols have security flaws, we are interested in the relative strength of the two protocols, irrespective of their correctness.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

QoP'06 October 30–November 3, 2006, Alexandria, Virginia, USA  
Copyright 2006 ACM 1-59593-518-5/06/0010 ...\$5.00.

Protocol 1.

$$Alice \xrightarrow{IamAlice} Bob \quad (1)$$

$$Alice \xleftarrow{r_B} Bob \quad (2)$$

$$Alice \xrightarrow{[r_B]_{k_{AB}}} Bob \quad (3)$$

Protocol 2.

$$Alice \xrightarrow{IamAlice} Bob \quad (4)$$

$$Alice \xleftarrow{[r_B]_{k_{AB}}} Bob \quad (5)$$

$$Alice \xrightarrow{r_B} Bob \quad (6)$$

Suppose an attacker, Carol, does not have the secret,  $k_{AB}$ . It is easy to notice that both protocols suffer from several known attacks such as impersonation attack, replay attack, and DoS attack [14]. For example, Carol can impersonate as Alice to Bob in step 1, 2, 4, and 5, and she can also impersonate as Bob to Alice in step 1, 2, 3, 4, and 6. If Carol can replay the message in step 5, she can impersonate as Bob to Alice. Step 5 may be potentially vulnerable to DoS attack in a resource-constrained environment. Overall how do we measure or order the strength with which the authentication goal is met? Can we formalize this analysis of informal reasoning, and compare the relative strength of security?

Section 2 discuss related work. In Sect. 3, we describe a model for *SoS* (Strength of Security) that can be used for analysis and measurement for the relative strength of protocols. In Sect. 4, we discuss the feasibility of our approach in measuring the relative strength of protocols using SoS model.

## 2. RELATED WORK

We discuss approaches used in the analysis of security protocols. The formal methods such as BAN logic [4], CSP [20], and Strand space [10] prove the correctness of cryptographic protocols. The limitations of BAN logic are that the approach focuses on authentication only, and the method claimed some protocols are correct, but found to be flawed later [15]. The CSP approach in which the agents are modelled as processes who can exchange messages via specific channels has a state-space explosion problem. The strand space method is successful at analyzing cryptographic protocols at the symbolic level. Another related work is soft constraints for security analysis used to provide a qualitative or quantitative value to security properties in a protocol [3]. The notion of security levels belongs to a finite total order, whereas we

are also interested in the *incomparable* nature of security properties as well.

### 3. SOS MODEL

#### 3.1 Notation

- $k$  A *secret key* in a symmetric algorithm
- $k_R$  *Public key* of the receiver  $R$  in an asymmetric algorithm
- $k_S^{-1}$  *Private key* of the sender  $S$  in an asymmetric algorithm
- $\langle \cdot \rangle$  The *send* operation over an insecure channel
- $m$  An *unprotected* message  $m$
- $[m]_k$  Message  $m$  is encrypted with a *symmetric* encryption algorithm  $[\cdot]$  using secret key  $k$
- $\{m\}_{k_R}$  Message  $m$  is encrypted with an *asymmetric* algorithm  $\{\cdot\}$  using public key  $k_R$
- $\{m\}_{k_S^{-1}}$  Message  $m$  is *digitally signed* with an *asymmetric* algorithm  $\{\cdot\}$  using private key  $k_S^{-1}$
- $m_1 \diamond m_2$  A message is composed of  $m_1$  and  $m_2$ , and does not consider permutation of the message components of a concatenated message.
- $m \mapsto m'$  Message  $m'$  is *derived* from message  $m$ , i.e.,  $m' = f(m)$  where  $f$  is a function
- $\wedge$  The logical AND operator
- $\vee$  The logical OR operator
- $\alpha_j \implies \alpha_{j'}$  If property  $\alpha_j$  exists, then property  $\alpha_{j'}$  exists as well, where  $j \neq j'$
- $\alpha_j \geq \alpha_{j'}$  Property  $\alpha_j$  is cryptographically *stronger* than or *equal* to property  $\alpha_{j'}$ , where  $j \neq j'$
- $\alpha_j \parallel \alpha_{j'}$  Property  $\alpha_j$  and property  $\alpha_{j'}$  are *incomparable*, where  $j \neq j'$

#### 3.2 Overview

Designing a security model requires us to understand a way to think about security in a manner that gives us a complete view of the relationship between the various components of strength of security. The SoS model (Strength of Security) is a holistic security framework which allows us to view the order-relationship of strength of security in multiple dimensions. Each dimension represents a different facet of the security. Here, we are not only interested in *stronger than* ( $\geq$ ) relationship, but also the *incomparable* ( $\parallel$ ) relationship as well.

For analysis of cryptographic protocols, we use a simple SoS model having three dimensions to demonstrate the feasibility of our approach: the 1<sup>st</sup> dimension represents the strength of cryptography, the 2<sup>nd</sup> dimension is security properties (e.g. confidentiality, authentication, randomness, privacy, integrity, and etc.), and the 3<sup>rd</sup> dimension represents the environment of the protocol (e.g. the capabilities of an

attacker, the applications using the protocol, and multiple protocol interactions [5]).

In the 1<sup>st</sup> dimension, SoS model assumes a non-ideal cryptography where there are a number of properties that need to be specified within that dimension to be evaluated. However, for the purpose of illustration here we assume an ideal cryptography, leaving out the first dimension and we are going to look at other dimensions.

In the 2<sup>nd</sup> dimension, we only discuss security properties of SoS such as confidentiality, authentication, and randomness due to space limitation. The rest is addressed in [6].

*Confidentiality* has several notions. For example, non-malleability implies indistinguishability under any type of attack [2]. That is non-malleability is considered as a stronger notion of security for encryption than indistinguishability under chosen-plaintext or non-adaptive chosen-ciphertext attacks, and being equivalent to indistinguishability under adaptive chosen-ciphertext attacks.

However, we will adopt a simple definition. If a message in a protocol is encrypted, we define that there is no explicit flow of input message given the output message, and we call it *No Explicit Flow (cfd.ex)*. A stronger notion of confidentiality compared to *No Explicit Flow* would be requiring unintended information other than the message itself to be undisclosed, and we call it *No Implicit Flow (cfd.im)*.

*Authentication* has many meanings [1, 16, 11]. In [16], G. Lowe introduces four different levels of authentication which include the notion of *matching history* [8]. We represent Lowe's authentication in SoS framework which is shown in Fig. 1.<sup>1</sup> Unlike Lowe's view on authentication as total ordering, we view it as having a partial order. Unilateral (*aut\_1*), mutual authentication (*aut\_2*), and data origin authentication (*aut\_org*) are additional definitions [18].

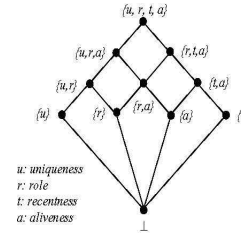


Figure 1: Lowe's authentication viewed in SoS

If a message in a protocol is random, and the message is not repeated over the lifetime of the protocol, then we define the message as having *Long Randomness (unq\_lr)* property (e.g.  $[m \diamond r]_k$ ). Note that  $\diamond$  operator gives a composition of security properties in the basic message types. If a message in a protocol is random and the message is not repeated only during the session of the protocol, then the message has *Short Randomness (unq\_sr)* property (e.g.  $[m \diamond i]_k$  where  $i$  is a predictable counter which gets reset at every session).

In the 3<sup>rd</sup> dimension, we adopt the threat model proposed by Dolev and Yao [9]. The attack list below are only non-algorithmic attacks. Note that the list is not meant to be complete, and other styles of attack, such as EM radiation and fluctuations in power consumption are outside the scope. The proofs on the logical relation between the

<sup>1</sup>We've defined agreement on *data* values under integrity [6].

properties as given in equations 7 to 9 are left as our future work.

If a message in a protocol is not vulnerable to *ciphersuite* or *protocol version rollback* attacks, then we define the message as having *atk\_rb* property. Often these types of attacks are targeted against the plaintext transmitted over an insecure channel [12, 21].

$$cfd\_ex \implies atk\_rb \quad (7)$$

If a message in a protocol is not vulnerable to an attack involving use of information from a previous protocol execution, then we define the message as having *atk\_re* property.

$$cfd\_ex \wedge (unq\_lr \vee unq\_sr) \implies atk\_re \quad (8)$$

If a message is not vulnerable to an impersonation attack, then we define that the message has *atk\_im* property.

$$(atk\_rb \wedge atk\_re) \implies atk\_im \quad (9)$$

### 3.3 Basic Message Types

First, we define six *basic data types*: non-predictable nonce  $r$ , predictable timestamp  $t$ , predictable sequence number  $s$ , non-predictable cryptographic key  $k$ , predictable identity of an agent  $id$ , and plaintext message  $m$ . Next, we define five *basic cryptographic operations*: symmetric encryption  $[\cdot]_k$ , asymmetric encryption  $\{\cdot\}_{k_R}$ , asymmetric signature  $\{\cdot\}_{k_S^{-1}}$ , asymmetric encryption of signature  $\{\{\cdot\}_{k_S^{-1}}\}_{k_R}$ , and hashing  $[\cdot]$ .

Given the basic cryptographic operators and the basic data types, we define the *basic message types* by means of applying the operators on the data types. For example, several basic message types are  $m, r, t, s, id$  (unprotected),  $[m]_k, [r]_k, [t]_k, [s]_k, [id]_k$  (symmetric encryption-based message types),  $\{m\}_k, \{r\}_k, \{t\}_k, \{s\}_k, \{id\}_k, \{m\}_k^{-1}, \{r\}_k^{-1}, \{t\}_k^{-1}, \{s\}_k^{-1}, \{id\}_k^{-1}, \{\{m\}_{k_S^{-1}}\}_{k_R}, \{\{r\}_{k_S^{-1}}\}_{k_R}, \{\{t\}_{k_S^{-1}}\}_{k_R}, \{\{s\}_{k_S^{-1}}\}_{k_R}, \{\{id\}_{k_S^{-1}}\}_{k_R}$  (asymmetric-based message types), and  $[m], [r], [t], [s], [id]$  (hash-based message types). For example, the strength associated with  $[r]_k$  are *cfd\_im*, *cfd\_ex*, *aut\_1*, *unq\_lr*, *unq\_sr*, *atk\_im*, *atk\_rb*, *atk\_re*, and etc. For the details, the reader may refer to [6].

We give a methodology for the assessment of protocols (e.g.  $P_1$  and  $P_2$ ) against the measurement lattice.

**Define goals.** Identify the *goal* in the protocol. Then, identify the *subgoal* of messages in each step of the protocol. For example, the goal of  $P_1$  and  $P_2$  is that Bob wants to authenticate Alice based on a shared secret  $k_{AB}$ . The subgoal in step 1 is to claim the initiator’s identity  $A$  to the responder  $B$ , in step 2 is to send a random challenge  $r$  by  $B$  to  $A$ . Finally, in step 3 is to send a response  $r'$  (i.e.  $r \mapsto r'$ ) to  $B$  by  $A$ .

**Formulate protocol.** Simplify the protocol by capturing the key components of the protocol, and formulate the protocol using the basic message types. This is called the *idealization* [6]. The idealization occurs when we add a security property (e.g. timeliness) to a basic message type representing a certificate in the protocol, or when we abstract the message type using  $\mapsto$  operator (e.g.  $(r_1, r_2, m \mapsto r_3)$ ). In addition, we indicate the *intensions* of each message types according to the subgoals.

Security protocol  $P$  is represented as a sequence of a set of basic message types, i.e.,  $X_1, X_2, \dots, X_n$ , where  $X_i$  is the set of basic message types in the  $i$ -th step of the protocol, and  $n$  is the maximum number of steps in the protocol.

For example, step 1, 2, and 3 are  $\langle id \rangle$ ,  $\langle r \rangle$ , and  $\langle [r_B]_{k_{AB}} \rangle$ , respectively. In the  $3^{rd}$  step in  $P_1$  when Alice sends  $\langle [r_A]_{k_{AB}} \rangle$  to Bob, the intension of the specification is that “Alice (*initiator*) sends (*action*) a random number (*object* with a certain cryptographic property, i.e. randomness) to Bob (*responder*)”.

**Add dimension.** Add a set of properties to be considered when evaluating the relative strength of protocol. A security protocol has specific objectives, and it is characterized by a set of security properties [1]. Therefore, *consistent property assignments* within a protocol are important. It is up to the security analyst to decide which set of security properties to include for that dimension. For example, we use Lowe’s authentication in Fig. 1 in the  $2^{nd}$  dimension, instead of a simple unilateral/mutual authentication since Lowe’s model has more information when represented in SoS model. In the future, we will investigate how to arrive at the set of properties in a consistent manner, and their assignments to message types and protocol goals.

**Perform measurement.** Given a set of exposed goals (and/or assumptions), measure the relative strength of the protocols along each dimension in SoS model. We state that if a partially ordered set (poset)  $A_1$ , a set of properties associated with  $P_1$ , is *substitutable* into  $A_2$ , a set of properties associated with  $P_2$ , then  $P_1 \leq P_2$ .<sup>2</sup>

In the  $2^{nd}$  dimension, both  $P_1$  and  $P_2$  achieves an *aliveness* guarantee of Alice to Bob in step 2, 3, 5, and 6. In the  $3^{rd}$  dimension, we identify that an attacker can impersonate as Alice to Bob in step 1, 2, 4, and 5. However, if we take potential weaknesses of the random number generator into account,  $P_2$  can be broken in step 6 while  $P_1$  cannot. Thus,  $P_1 \geq P_2$ . Note that the aliveness of Bob to Alice is not considered, since both  $P_1$  and  $P_2$  never meant to achieve mutual authentication or any aliveness of Bob.

## 4. RELATIVE STRENGTH ANALYSIS

Due to space constraints, we only briefly sketch the relative strength analysis of TLS 1.0 and SSL 2.0 handshake protocols, and the details are refer to [6]. The similar analysis applies to the alternative protocol versions.

By studying the TLS specification we learn that the TLS handshake protocol has the following four goals [7]: (i) exchange cryptographic preferences (e.g. *ClientHello* ( $M1$ ) and *ServerHello* ( $M2$ ) messages), (ii) exchange TLS version number (e.g.  $M1$  and  $M2$ ), (iii) entity authentication (e.g. *ServerCertificate* ( $M3$ ), *ServerKeyExchange* ( $M4$ ), and *CertificateRequest* ( $M5$ )), and (iv) exchange secret to be used in at the record layer protocol (e.g. *ClientKeyExchange* ( $M8$ )).

After representing the TLS handshake protocol using the basic message types and *intensions*, it is easy to see that the first two messages of type  $\langle m \diamond r \rangle$  are vulnerable against rollback attacks. The rollback attack exploits the lack of protection on any messages to result in a least common denominator security. The server authentication by client is achieved due to  $M3$ ,  $M4$  and  $M5$ .<sup>3</sup> Examining

<sup>2</sup>If  $\alpha$  and  $\beta$  are *order types* of properties, and  $\alpha$  is *embeddable* in  $\beta$ . Also, let  $\gamma$  be an order type of the subset of  $\beta$ , which is order-isomorphic to  $\alpha$ . If the mapping of the maximal elements (resp. minimal elements) of  $\alpha$  into  $\beta$  doesn’t violate the order-relation with their parents (resp. children) in  $\beta$  when  $\gamma$  is replaced by  $\alpha$ , then  $\alpha$  is *substitutable* in  $\beta$ .

<sup>3</sup>Message 4 is represented as  $\langle m \diamond \{\{r_3\}\}_{k_S^{-1}} \rangle$ , where

*ClientKeyExchange* ( $M8$ ) and *CertificateVerify* ( $M9$ ) along the 3<sup>rd</sup> dimension,  $M8$  containing *premaster* is vulnerable against replay attack, and allow the attacker to present compromised *premaster* (e.g. old *premaster*) previously encrypted by the client using server's public key,  $k_S$ . Although the attacker cannot complete the protocol, the message type,  $\{r_3\}_{k_S}$ , lacks integrity, and the weakness may lead to discovery of vulnerabilities discussed in [21].

Trivially, we find that the hello messages in the SSL 2.0 protocol (e.g. assuming no session-identifier) [13] are also vulnerable against rollback attack. Thus, the first two goals have failed. We also observe that the protocol allows an attacker to launch the man-in-the-middle attack such that the server believes it has shared a session key with the client [6], but in reality, the server is sharing the key with the attacker. This is due to the weaknesses associated with the basic message types in the protocol such as the lack of *atk\_rb*, and *atk\_im*. Although the attack does not allow the attacker to have the client to believe that it is sharing the key with the server but actually the client is sharing with the attacker, this may pose a security threat. Consequently, the final goal - exchange of secret, has failed. In short, the model states that the TLS handshake protocol is relatively stronger than that of SSL 2.0 only in the sense and to the extent that the protocol achieves its goals.

## 5. CONCLUSION

In this paper, we have presented a way to think about the relative strength of security protocols using SoS model, where certain known rankings in several dimensions are used to model a range of security systems using lattice-based structure.

In the past, security protocol analysis has focused on formal methods and provable security, especially on the correctness of security protocols under a set of assumptions. Relative security is interested in security systems (e.g. security protocols) whose strength are difficult to quantify under varying sets of assumptions. We have shown how to order the relative strength of these systems in situations where the correctness approach may not help us. Therefore, the relative strength is an important area of research.

## 6. REFERENCES

- [1] G. Bella. *Inductive Verification of Cryptographic Protocols*. PhD thesis, Clare College University of Cambridge, 2000.
- [2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption schemes. *Advances in Cryptology-CRYPTO '98*, 1462:162–177, 1998.
- [3] S. Bistarelli, G. Bella, and S. Foley. Soft constraints for security. In *First International Workshop on Views On Designing Complex Architectures (VODCA)*, September 2004.
- [4] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. In *Proceeding of the Royal Society of London*, 1989.
- [5] R. Canetti, C. Meadows, and P. Syverson. Environmental requirements for authentication  $(r_1, r_2, m) \mapsto r_3$ . Ideally, the strength of the message type should be the same as before the *idealization* process on that message type.
- protocols. In *Proceeding of the International Symposium on Software Security*, pages 339–355. Springer-Verlag, 2002.
- [6] H. Chung and C. Neuman. Modelling the relative strength of security protocols. Technical Report 06-882, University of Southern California, Computer Science Department, August 2006.
- [7] T. Dierks. The TLS protocol, version 1.0. *RFC 2246*, January 1999.
- [8] W. Diffie, P. C. van Oorschot, and M. J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2(2):107–125, June 1992.
- [9] D. Dolev and A. C. Yao. On the security of public key protocols. In *Proceeding of the IEEE 22nd Annual Symposium on Foundations of Computer Science*, pages 350–357, 1981.
- [10] F. T. Fábrega, J. Herzog, and J. D. Guttman. Strand spaces: Why is a security protocol correct? In *Proceeding of the 16th IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 1998.
- [11] D. Gollmann. What do we mean by entity authentication? In *Proceeding of the 1995 IEEE Symposium on Security and Privacy*, page p46, 1996.
- [12] C. He and J. Mitchell. Security analysis and improvements for IEEE 802.11i. In *Proceeding of the 12th Annual Network and Distributed System Security Symposium (NDSS'05)*, 2005.
- [13] K. E. Hickman. The SSL 2.0 protocol. <http://wp.netscape.com/eng/security/SSL2.html>, January 1995.
- [14] C. Kaufman, R. Perlman, and M. Speciner. Network security: Private communication in a public world. In *Prentice Hall PTR, 2nd Edition*, pages 257–264, 2002.
- [15] G. Lowe. Breaking and fixing the needham-schroeder public-key protocol using FDR. In *Proceeding of the 2nd International Workshop on Tools and Algorithms for Construction and Analysis of Systems*, volume 1055, pages 147–166. Lecture Notes in Computer Science, 1996.
- [16] G. Lowe. A hierarchy of authentication specifications. In *Proceeding of The 10th Computer Security Foundations Workshop*, 1996.
- [17] C. Meadows. Formal methods for cryptographic protocol analysis: Emerging issues and trends. *IEEE Journal on Selected Areas in Communication*, 21(1):44–54, January 2003.
- [18] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
- [19] J. Mitchell, V. Shmatikov, and U. Stern. Finite-state analysis of SSL 3.0. In *7th USENIX Security Symposium*, 1998.
- [20] S. Schneider. Verifying authentication protocols with CSP. In *Proceeding of the 10th IEEE Computer Security Foundations Workshop*, pages 3–17. IEEE Computer Society Press, 1997.
- [21] D. Wagner and B. Schneier. Analysis of the SSL 3.0 protocol. In *The 2nd USENIX workshop on Electronic Commerce*, pages 29–40. USENIX Press, 1996.