

# Modelling the Relative Strength of Security Protocols

**Ho Chung**

Computer Science  
University of Southern California  
hochung@usc.edu

**Clifford Neuman**

Computer Science  
University of Southern California  
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.

## 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 [3, 5, 9, 10, 11, 13, 19]. Very little work has been done on determining the relative strength of security protocols [4]. 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.

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 [7]. 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 is 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

In this section, we discuss approaches used in the analysis of security protocols. The formal methods such as BAN logic [19], CSP [13], 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 claim 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 seems to be 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 [4]. 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 composed of  $m_1$  and  $m_2$ , and the *order* of  $m_1$  and  $m_2$  is unimportant
- $\wedge$  The logical AND operator
- $\vee$  The logical OR operator
- $\nabla$  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, and integrity), 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 [6]).

In the 1<sup>st</sup> dimension, we assume non-ideal cryptography. In the 2<sup>nd</sup> dimension, we only discuss security properties of

SoS such as confidentiality, authentication, and randomness due to space limitation.

*Confidentiality* has several notions [8]. For example, non-malleability implies indistinguishability under any type of attack [8]. 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* (*cf<sub>d</sub>.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* (*cf<sub>d</sub>.im*).

*Authentication* has many meanings [9, 16, 17]. In [16], G. Lowe introduces four different levels of authentication which include the notion of *matching history* [18]. We represent Lowe’s authentication in SoS framework 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 [12].

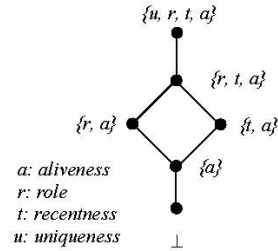


Figure 1: Lowe’s authentication viewed in SoS

If a message in a protocol is random for the lifetime of the protocol, then we define the message as having *Long Randomness* (*unq\_rn*) property (e.g.  $[m \diamond r]_k$ ). If a message in a protocol is random for 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 [20]. The attack list below are only non-algorithmic attacks. Other styles of attack, such as EM radiation and fluctuations in power consumption are outside the scope.

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 [3, 14].

$$cf_{d\_ex} \implies atk\_rb \tag{7}$$

<sup>1</sup>Note: we have ignored the agreement on *data* values since it is out of scope according to our definition of authentication.

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.

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

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  $[x]_k$ , asymmetric encryption  $\{x\}_{k_R}$ , asymmetric signature  $\{x\}_{k_S^{-1}}$ , asymmetric encryption of signature  $\{\{x\}_{k_S^{-1}}\}_{k_R}$ , and hashing  $[x]$ .

Given the basic cryptographic operators and the basic data types, we now define the *basic message types* by means of applying the operators on the data types. For example, several basic message types are  $m$ ,  $r$ ,  $t$ ,  $s$ , and  $id$  (unprotected),  $[m]_k$ ,  $[r]_k$ ,  $[t]_k$ , and  $[s]_k$ , and  $[id]_k$  (symmetric encryption-based message types),  $\{m\}_k$ ,  $\{r\}_k$ ,  $\{t\}_k$ ,  $\{s\}_k$ ,  $\{id\}_k$ ,  $\{m\}_{k_S^{-1}}$ ,  $\{r\}_{k_S^{-1}}$ ,  $\{t\}_{k_S^{-1}}$ ,  $\{s\}_{k_S^{-1}}$ ,  $\{id\}_{k_S^{-1}}$ ,  $\{\{m\}_{k_S^{-1}}\}_{k_R}$ ,  $\{\{r\}_{k_S^{-1}}\}_{k_R}$ ,  $\{\{s\}_{k_S^{-1}}\}_{k_R}$ , and  $\{\{id\}_{k_S^{-1}}\}_{k_R}$  (asymmetric-based message types), and  $[m]$ ,  $[r]$ ,  $[t]$ ,  $[s]$ , and  $[id]$  (hash-based message types). We do not discuss strength and weaknesses associated with each basic message types due to space constraints, but the details will appear in [2].

### 3.4 Methodology

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

**Define goals.** First, identify *goal(s)* for the protocol. Then, in each step of the protocol, identify *task* (or *subgoal*). 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 of step 1 is to claim the initiator’s identity to the responder, step 2 is to send a random challenge by the responder to the initiator. Finally, the goal of step 3 is to send a response to the responder by the initiator.

**Formulate protocol.** Simplify the protocol by capturing the key components of the protocol and *formulate* the protocol using the *basic message types*. We also indicate the *intensions* of each message types.

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 addition, we emphasize the *role* of each basic message types, instead of the role of agents participating in the protocol. In the 3<sup>rd</sup> 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 [9]. Since we are interested in authentication, we will use Lowe’s authentication in Fig. 1 in the 2<sup>nd</sup> dimension, and use the threat model abovementioned in the 3<sup>rd</sup> dimension.

**Perform measurement.** Given a set of exposed assumptions and goals, 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  $A_1 \leq A_2$ .<sup>2</sup>

In short,  $P_1$  does not satisfy *aliveness* because Alice has to completely trust the agent at the other end of the communication. However,  $P_2$  guarantees *aliveness*. In other words, the authentication poset representing  $P_1$  is embeddable in  $P_2$ , thus,  $P_1 \geq P_2$ . For the 3<sup>rd</sup> dimension, we can easily identify the impersonation attack associated with the basic message types in  $P_1$  and  $P_2$ . Likewise, we can state  $P_1 \geq P_2$  with respect to *atk\_im* in the 3<sup>rd</sup> dimension.

## 4 RELATIVE STRENGTH ANALYSIS

Due to space constraints, we briefly sketch the relative strength analysis of TLS 1.0 and SSL 2.0 handshake protocols only.

The TLS specification [1] does not clearly list the requirements for the TLS handshake protocol. However, by studying the specification we learn that TLS handshake protocol has the following four goals: (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 TLS handshake protocol using the basic message types, it is easy to see that 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 (e.g. in  $M3$ ,  $M4$  and  $M5$ ). Examining *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 [14].

Trivially, we find that the hello messages in the SSL 2.0 protocol are also vulnerable against rollback attack. Thus,

<sup>2</sup>Suppose  $\alpha$  and  $\beta$  are *order types* of security 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  $\beta$  does not violate the order-relationship with their parents (resp. children) in  $\beta$  when  $\gamma$  is replaced by  $\alpha$ , then we say that  $\alpha$  is *substitutable* in  $\beta$ .

the first two goals stated 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 [2], but in reality, the server is sharing the key with the attacker. 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 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.

So far, security protocol analysis has to do with formal methods and provable security. The formal methods and provable security that can talk about correctness of security protocols under a set of assumptions, whereas the relative security is interested in those that are difficult to quantify in the sense that how to compare two systems that one is secure under one set of assumptions, and another that is secure under a different set of assumptions. Here we need to come up with how to order the importance of those assumptions. So, in this situation the correctness approach may not help us. Therefore, we believe that the relative strength is an important research area.

## REFERENCES

- [1] T. Dierks, *The TLS Protocol, version 1.0*, RFC 2246, January 1999.
- [2] Ho Chung, and Clifford Neuman, *Modelling the Relative Strength of Security Protocols.*, to appear in USC/ISI technical report, 2006.
- [3] Changhua He, and John Mitchell, *Security Analysis and Improvements for IEEE 802.11i*, In Proc. of the 12th Annual Network and Distributed System Security Symposium (NDSS'05), 2005.
- [4] Stefano Bistarelli, Giampalo Bella, and Simon Foley, Constraints for Security, First International Workshop on Views On Designing Complex Architectures (VODCA), Sept. '04. Springer Verlag Electronic Notes in Computer Science.
- [5] Catherine Meadows, *Formal Methods for Cryptographic Protocol Analysis: Emerging Issues and Trends*, IEEE Journal on Selected Areas in Communication, vol.21, No.1, pp.44-54, Jan. 2003.
- [6] Ran Canetti, Catherine Meadows, and Paul Syverson, *Environmental Requirements for Authentication Protocols*, Proceedings of the International Symposium on Software Security, Springer-Verlag LNCS 2609, pp.339-355, 2002.
- [7] Charlie Kaufman, Radia Perlman, Mike Speciner, *Network Security: Private Communication in a Public World*, Prentice Hall PTR; 2nd ed., 2002.
- [8] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, *Relations Among Notions of Security for Public-Key Encryption Schemes*, An extended abstract of this paper appears in Advances in Cryptology-CRYPTO '98, Lecture Notes in Computer Science Vol.1462, Springer-Verlag, 1998.
- [9] Giampaolo Bella, *Inductive Verification of Cryptographic Protocols*, Clare College University of Cambridge, PhD dissertation, 2000.
- [10] F.J. Thayer Fábrega, J.C. Herzog, and J. D. Guttman, *Strand Spaces: Why is a Security Protocol Correct?*, In Proceedings of the 17th IEEE Symposium on Security and Privacy, IEEE Computer Society Press, 1998.
- [11] John Mitchell, Vitaly Shmatikov, and Ulrich Stern, *Finite-State Analysis of SSL 3.0*, 7th USENIX Security Symposium, 1998.
- [12] Alfred Menezes, Paul van Oorschot, and Scott Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
- [13] S. Schneider, *Verifying Authentication Protocols with CSP*. In Proc. of the 10th IEEE Computer Security Foundations Workshop, pp.3-17, IEEE Computer Society Press, 1997.
- [14] David Wagner, and Bruce Schneier, *Analysis of the SSL 3.0 protocol*, The 2nd USENIX workshop on Electronic Commerce Proceedings, USENIX Press, pp.29-40, 1996.
- [15] Gavin Lowe, *Breaking and Fixing the Needham-Schroeder Public-Key Protocol using FDR*, Proc. of the 2nd International Workshop on Tools and Algorithms for Construction and Analysis of Systems, Lecture Notes in Computer Science; Vol.1055, pp. 147-166, 1996.
- [16] Gavin Lowe, *A Hierarchy of Authentication Specifications*, Proc. of The 10th Computer Security Foundations Workshop, 1996.
- [17] Dieter Gollmann, *What do we mean by Entity Authentication?*, Proc. of the 1996 IEEE Symposium on Security and Privacy, pp. 46, 1996.
- [18] Whitfield Diffie, Paul C. van Oorschot and Michael J. Wiener *Authentication and Authenticated Key Exchanges*, Designs, Codes and Cryptography, Volume 2, Issue 2, June 1992, pp. 107-125.
- [19] M. Burrows, M. Abadi, and R. Needham, *A logic of authentication*, Proc. of the Royal Society of London, 426, 1989.
- [20] D. Dolev and Andrew C. Yao, *On the Security of Public Key Protocols*, Proc. of the IEEE 22nd Annual Symposium on Foundations of Computer Science, pp.350-357, 1981.