A New Method to Security Electronic Commerce Protocol

Yang Jie
School of Computer Science and Engineering, SCUT
yjclear@scut.edu.cn

Mo Hai-guang
School of Computer Science and Engineering, SCUT
scutmo@126.com

Long Yu-heng
SCSE of SCUT
kewelllong@21cn.com

Abstract

We use semi-regular entities to denote the entities different from penetrates that participate in the protocol on behalf of themselves such as customers in e-commerce.

1. Introduction

Electronic transactions are protected by e-commerce protocols categorized as foundation-protocol to establish “secure tunnel” used by high layer protocols and application-protocols to complete electronic transactions through the “secure tunnel” established by the former. Formal methods are used to verify e-commerce protocols. It is called as honest assumption that all entities (except penetrates) comply with the analyzed protocol strictly. In our new method is a new conception: semi-honest entity (semi-regular strand) based on strand space model. Besides regular strand and penetrate strand, semi-regular strand is also redefined for dishonest participant. We also change protocol spaces and analyze TLS and IKE. Finally we find an attack on the IKE protocol using our method.

2. One-Way Hash Function

The original strand space model contains no description about hash function so it is added as a unary operator: hash: A \rightarrow A, where A is a set of terms including all messages potential generated by a protocol. Accordingly, a new penetrate trace is added to the original set of penetrate traces: H: <-g, +hash(g)>

Definition 2.1 \( S \subseteq A, W[S] \) is the smallest set that satisfies the following two conditions:

1. \( \forall g \in S, \text{then } g \notin W[S] \)
2. \( \forall g \in W[S], h \in A, \text{then } gh, hg \in W[S] \)

W [S] is equal to I-[S]. It denotes the domain of hash function that contains any element of the set S.

Definition 2.2 \( S \subseteq A, W[S] \) is defined in Definition 2.1. H [S] is the smallest set that satisfies the following conditions: \( \forall g \in W[S], \text{then } \text{hash}(g) \in H[S] \). So hash is a bijection between W [S] and H [S]. We write \( H [\{g\}] \) and \( W [\{g\}] \) as \( W [g] \) and \( H [g] \) respectively for concision in the rest of this paper.

Theorem 2.1 Suppose \( S \subseteq A, x \in H [S], K \) is the set of all keys. Whenever a penetrate node p is an entry point for IK[x] (written as I[x] for short), then p belongs to penetrate strand H. In other words, x cannot originate on any penetrator node, except for H nodes.

3. Diffie-Hellman Exchange and Semi-regular Strand with Node

Definitions 3.1 In a D-H exchange, the data exchanged between the two entities are g and h, and then we write the result of the exchange as dh (g, h).

Definition 3.2 An entity X, if X knows a message m, then we say m \( \in \text{KNOWN}_X \).

Proposition 3.1 Suppose an entity X and a D-H exchange with exchanged data g and h, if g, h \( \in \text{KNOWN}_X \), and at least one is originated by X, then dh (g, h) \( \in \text{KNOWN}_X \).

Definition 3.3 An arbitrary strand with some limitations is a semi-regular strand. The node on a semi-regular strand is a semi-regular node.

Definition 3.4 Suppose s is a strand, s is normal if \( \forall x \in LS \), no node in s is an entry point for I[x].
4. The Transport Layer Security Protocol

The Transport Layer Security (TLS) protocol [3] is a version of the Secure Sockets Layer (SSL) protocol. **Definition 4.1** Suppose M1=INiPi, M2= NrSidPr, M3={PMS} kr{hash(Nr, R, PMS)} ki-1{Finished_I} ksi, M4={Finished_R} ksr

1. **Init** [I, R, Sid, Ni, Nr, Pi, Pr, PMS] is the set of strands, whose trace is (+M1, -M2, +M3, -M4).
2. **Resp** [I, R, Sid, Ni, Nr, Pi, Pr, PMS] is the set of strands, whose trace is (-M1, +M2, -M3, +M4)
3. **Init’** [I, R, Sid, Ni, Nr, Pi, Pr, PMS] is the set of strands, whose trace is (…, +M1, …, -M2, …, +M3, …, -M4, …).
4. **Resp’** [I, R, Sid, Ni, Nr, Pi, Pr, PMS] is the set of strands, whose trace is (…, -M1, …, +M2, …, -M3, …, +M4, …).

**Proposition 4.1** If s ∈ Initi [I, R, Sid, Ni, Nr, Pi, Pr, PMS], Ni originates on <s,1> and PMS originates on <s,3>.

**Proposition 4.2** If s ∈ Resp [I, R, Sid, Ni, Nr, Pi, Pr, PMS], then Nr originates on <s, 2>.

**Theorem 4.1** Suppose C is a bundle in TLS I,R’, kr-1 is known by R only. PMS is uniquely originating. Any strand in Resp’ is normal. If s ∈ Initi [I, R, Sid, Ni, Nr, Pi, Pr, PMS] has C-Height 4, then Finished_R is originated by R in a strand t ∈ Resp’ [I, R, Sid, Ni, Nr, Pi, Pr, PMS].

**Theorem 4.2** Suppose C a bundle in IKEAP I,R; ki -1 known by I only and kr -1 is known by R only; PMS uniquely originating in C and originated by I; any strand in Initi is normal. If s ∈ Resp [I, R, Sid, Ni, Nr, Pi, Pr, PMS] has C-Height 4, Finished_I is originated by I in a strand t ∈ Initi [I, R, Sid, Ni, Nr, Pi, Pr, PMS].

5. The Internet Key Exchange Protocol

**Definition 5.1** Suppose M1= CiSAiKEi{I}kr{Ni}kr, M2= CiCrSArKEr{R}kr{Nr}krHASH_R, M3= CiCrHASH_I

1. **Init** [I, R, Ci, Cr, SAi, SAr, KEi, KEr, Ni, Nr] is the set of strands, whose trace is (+M1, -M2, +M3).
2. **Resp** [I, R, Ci, Cr, SAi, SAr, KEi, KEr, Ni, Nr] is the set of strands, whose trace is (-M1, +M2, -M3)
3. **Init’** [I, R, Ci, Cr, SAi, SAr, KEi, KEr, Ni, Nr] is the set of strands, whose trace is (…, +M1, …, -M2, …, +M3, …).
4. **Resp’** [I, R, Ci, Cr, SAi, SAr, KEi, KEr, Ni, Nr] is the set of strands, whose trace is (…, -M1, …, +M2, …, -M3, …).

**Proposition 5.1** If s ∈ Initi [I, R, Ci, Cr, SAi, SAr, KEi, KEr, Ni, Nr], Ni and KEi originate on <s,1>.

**Proposition 5.2** If s ∈ Resp [I, R, Ci, Cr, SAi, SAr, KEi, KEr, Ni, Nr], Ni and KEr originate on <s,2>.

**Theorem 5.1** Suppose C is a bundle in IKEAP I,R’, Ni∉ KNOWNp. If s ∈ Initi [I, R, Ci, Cr, SAi, SAr, KEi, KEr, Ni, Nr] has C-Height 3, HASH_R is originated by R in a strand t ∈ Resp’ [I, R, Ci, Cr, SAi, *, KEi, KEr, Ni, Nr].

**Theorem 5.2** Suppose C is a bundle in IKEAP I,R, Nr∉ KNOWNp. If s ∈ Resp [I, R, Ci, Cr, SAi, SAr, KEi, KEr, Ni, Nr] has C-Height 3, HASH_I is originated by I in a strand t ∈ Initi [I, R, Ci, Cr, SAi, *, KEi, KEr, Ni, Nr].

Then we find an attack on the IKEAP protocol as:

I → R: Ci, SAi, KEi, {I}kr, {Ni}kr;
R → V: Ci, SAi, KEi, {R}kv, {Ni}kv
V → R: Ci, Cr, SA, KE, {V}kr, {Nr}kr, HASH_V;
R → I: Ci, Cr, KEr, {R}ki, {Nr}ki, HASH_R
I → R: Ci, Cr, HASH_I;    R → V: Ci, Cr, HASH_R

6. Conclusions

Nothing can guarantee honesty even the honest entities in conventional analysis. Semi-honest entity likes a black box. The outer actions are identical to some specification and we know nothing about the inner actions. It is just a special case in Definition 3.3. To illustrate our method, we give TLS and IKE instances with hash function and D-H exchange added to the model. The spaces TLS I,R’, and IKEAP I,R’ are used to prove guarantee for initiator and similar for responder. Analysis leads to a deeper insight into TLS, and the exact conditions ensuring the protocol’s security. An attack that hides in the original strand space model on the IKE protocol is found, too. This work is supported by the G03-E5041450 and SRP (G03-Y1060140, Y1060150).

References