Proceedings of 14th Youth Conference on Communication

Scientific Research

Formal Analysis of Authenticated Key Distribution Protocol Using Extended SVO Logic

Liu Zhimeng, Fan Hui, Feng Yanli, Zhao Yanli
School of Computer Science and Technology', Shandong Institute of Business and Technology, Yantai, China, 264005
liu_zhimeng@126.com

Abstract: Some new notions and approaches of SVO logic are introduced, which make it has some ability to analyze some authenticated key distribution protocols, and these new notions and axioms can be used to verifying the validity of certificate and the verity of its owners,. In the procedure of our formal derivation of security goals, some conclusions have been derived that Aydos et al.'s protocol can not resist attacks forward security and unkown key-share attack.

Keywords: authentication protocol; key distribution; formal analysis; SVO logic;

1 Introduction

An authentication protocol is an exchange of messages having a specific form for authentication of principals using cryptographic algorithms. They typically have additional goals such as the distribution of session keys. Security protocols may have any number of intended purposes, such as non-repudiation voting anonymous etc. We will focus on authenticated establishment of session keys, which is typically necessary for the running of security protocols for most other purposes. Recently, Aydos et al. proposed a efficient mutual authentication and key agreement protocols (MAKAP) [1] based on elliptic curve cryptography (ECC) [2] for wireless communication, which can establish a secure communication between a low user and a powerful network server.

Burrows, Abadi, and Needham developed BAN logic, which quickly become the most, widely used and widely discussed formal method for the analysis of authentication protocols, particularly authenticated key distribution protocols. There is fact that the BAN logic has not ability to reason about some features of both protocols and attacks on protocols. Its successor SVO logic [3-4] was presented by Syverson and van Oorschot. Though SVO logic has been widely used in the analysis of authenticated key distribution protocols for its simplicity, we find that it is weak to analyze security of key agreed in an authentication protocol based on certificate, such as forward secrecy property, and verify validity of a participant by certificate created by the Certification Authority (CA), and the attack procedures are deduced in this paper.

2 Aydos et al.' Protocol

2.1 User and Server Initialization

The server selects his secret key d_s and computes its public key $Q_s = d_s \times P$. Next, the server sends his public to the CA. Upon the received message, the CA signs a unique identity ID_s and an expiration dates t_s, and computes $R_s = k_s \times P$, $r_s = R_s.x$, $e_s = h(Q_s.x, ID_s, t_s)$, and $s_s = k_s^{-1} e_s + d_{ca} r_s$ mod n, where k_s is a random number. Then the CA returns Q_{ca}, ID_s, (r_s, t_s), t_s to the server. Finally, the server computes $e_s = h(Q_s.x, ID_s, t_s)$, and stores $< Q_s, Q_{ca}, ID_s, (r_s, t_s), e_s, t_s >$

Similarly, the user performs the same steps above and stores $< Q_u, Q_{ca}, ID_u, (r_u, t_u), e_u, t_u >$.

2.2 Mutual Authentication Phase

The mutual authentication phase is executed in real time, i.e., whenever a service is requested by the user or server. Firstly the initiating party, user sends its public key Q_u to the server which is initiated party. Then the server generates random number g_s and sends its public Q_s, and g_s to the user. Finally, the user and server compute $d_u \times Q_s$ and $d_s \times Q_u$, respectively, to agree on a mutual key $Q_K.x$(x coordinate of the point Q_K). Secondly, the user generates a random g_u, uses a symmetric key encryption algorithm E to encrypt its certificate $\{e_u, (r_u, t_u), g_u, g_s, t_u\}$ with the mutually agreed key $Q_K.x$ to obtain C0, and sends C0 to the server. The server decrypt C0 using a decryption algorithm D with the mutually agreed key and checks for the presence of t g_s and the validity of t_u. If both tests are valid then the server encrypts $\{(r_s, t_s), t_s, e_s, g_u\}$ to obtain

---

Foundation item: Supported by National Natural Science Foundation of China (No. 60773053).

C1 and sends C1 to the user. The user checks for the presence of g_u and the validity of t_s. Both parties verify each other's certificate. If invalid, they abort the protocol, otherwise they derive a unique session key K_m by computing the hash on $Q_K.x$, g_u and g_s in the end of the protocol.

3 SOV Logic

In this section, some new notions and approaches are introduced to SVO logic.

3.1 Extension of SOV Logic

First, we presented a extended Axiom, as follows:

A0. (P believes $\varphi \wedge P$ believes ψ) $\equiv$ (P believes $(\varphi \wedge \psi)$).

Hash function is the one-way function, a hash function accepts a variable-size message as input and produces a fixed-size output referred to as a hash code which can be used to provide message authentication which is a mechanism used to verify the integrity of a message, and assures that data received are exactly as sent by and that the purported identity of the sender is valid, In a authentication protocol based on ECC, a certificate of entity A is denoted as:

$$Cert_A = < I_A, m, Sig_{CA}(H(I_A, m))>$$

Where I_A means the identification information of A and m is a message to be signed by CA over the concatenation of public key Q_A of A, I_A and expiration date t_A of $Cert_A$. Because SVO logic does not include the axioms which can be used to verifying the validity of certificate, we extend two axioms which make it has great capabilities in analyzing trusted third-party based authentication and key agreement protocols, as follows:

Certificate and Subject Verification: Key and Hash code are used to deduce the validity of the sender's identity.

A1: $PK_\sigma(\mathrm{CA}, Q_{CA}) \wedge A \lhd * \wedge SV(Sig, Q_{CA}, H) \supset CV(Cert, Q_{CA}, *)$ and A send $A \ni Cert_A$

A2: $CV(Cert_A, Q_{CA}, *) \wedge (H = H(x_A)) \supset Cert_A$, where $x_A = <Q_A, ID_A, t_A>$

Recall that $PK\sigma(C_A, Q_{CA})$ says that Q_{CA} is the public signature verification key for CA, and $SV(Sig, Q_{CA}, H)$ says that given signed H, applying Q_{CA} to it as a signature verification key verifies * as the message signed with d_{CA} (private key of CA according to Q_{CA}), received a certificate, and verification of Sig included in unknown message * could be verified using CV's public key of signature, then the unknown message * is a vilification certificate of an honest principal. If the received hash code equal to the result of hashed code of the concatenation of the public key, the temporary identity I_A created by CA, and the certification expiration date t_A recomputed by message receiver, means that the certification belong to A, and A send the message that he has $Cert_A$.

3.2 Genetic Formal Goals

G1 Far-end operative: P believes Q says X

G2 Targeted entity authentication: P believes Q says $F(X, N_P)$

G3 Secure key establishment: P believes $P \leftarrow K- \rightarrow Q$

G4 Key confirmation: P believes $P \leftarrow K+ \rightarrow Q$

G5 Key freshness: P believes fresh (K)

G6 Mutual understanding of shared key: P believes (Q says $Q \leftarrow K- \rightarrow P$).

3.3 Formal Analysis of the Protocol

3.3.1 Initial Assumptions

The first step in analyzing the protocol is to set out the assumptions that we make based on the protocol specification. And these assumptions will serve as premises, which will be used together with the axioms and the rules of the logic to derive conclusions. All assumptions of entity B that we make based on the protocol specification as follows:

1) $A \mid\equiv \{PK_\sigma(CA, Q_{ca}), PK_\sigma(A, Q_A), PK_\sigma(B, Q_B)\}$, $A \mid\equiv \{SV(X, Q_{CA}, Y), CV(X, Q_{CA}, Y)\}$
2) $A \mid\equiv \#(g_A)$, $A \mid\equiv \{A \ni g_A, PK_\delta(A, g_A)\}$, $A \mid\equiv B \ni Q_A$
3) $A \mid\equiv A \lhd (Q_B, g_B, \{C_1^B\}Q_{k.x}, \{x\}_{Km})$
4) $A \mid\equiv (A \mid\approx g_A \wedge A \lhd (Q_B, g_B)) \supset A \mid\equiv \{B \lhd Q_A \wedge B \mid\equiv PK_\sigma(A, Q_A \rightarrow Q_K) \wedge B \mid\equiv PK_\sigma(B, Q_B \rightarrow Q_K)\} \wedge (A \mid\equiv B \mid\approx (Q_B, g_B) \supset A \mid\equiv PK_\delta(B, Q_B \rightarrow Q_K))$
5) $A \mid\equiv SV((r_B, s_B), Q_{CA}, e_B) \supset A \mid\equiv Cert_B \supset A \mid\equiv PK_\delta(B, g_B \rightarrow K_m) \wedge A \mid\equiv (B \mid\equiv (PK_\delta(A, g_A \rightarrow K_m) \wedge \#(g_A)) \wedge A \mid\sim (A \ni g_A))$
6) $E \ni \{d_S, d_U, C'_0, C'_1, Q'_R, H, T(\#E, a, b, P, n, h)\}$

3.3.2 Forward Secrecy

The forward secrecy property is that if secret keys including dS and dU of S and U respectively are compromised, the session keys used in the past should not be recovered. Assume that are known to an adversary E, and E has all the information exchanged between S and U.

1) $E \ni (d_A, d_B) \wedge E \ni (Q_A, Q_B) \supset E \ni d_A \times Q_B = d_B \times Q_A = (d_A d_B) \times Q_A = Q_K \supset \mathrm{E} \ni Q_K.x$
2) $E \perp (B \rightarrow A\text{: } \{C_0\} Q_K.x) \wedge E \ni Q_K.x \supset E \ni C_0 = \{e_B, r_B, s_B, t_B, g'_A, g'_B\} \supset E \ni g'_B$， Similarly, $E \ni g'_A$
3) $E \ni Q_K.x \wedge E \ni g'_B \wedge E \ni g'_A \supset E \ni H(Q_K.x, g'_A, g'_B) = F(Q_K.x, g'_A, g'_B) = K'_m$.

Namely, Aydos et al's Protocol does not provide forward secrecy.

3.3.3 Attacks to Authentication

The derivation is of goal for B, which is the initiated party in the protocol. The goals we drive here are that B believes that the distributed key is good for talking with A, and B believes that the distributed key is fresh. We denote symbol $\perp$ that the adversary can intercept and capture all message exchaged between A and B.

1) $E \perp (A \rightarrow B\text{: } Q_A) \supset E \ni Q_A$
2) $E \rightarrow B\text{:} Q_E \supset E \ni Q_E$
3) $E \perp (B \rightarrow A\text{: } (Q_B \wedge g_B)) \supset E \ni (Q_B \wedge g_B) \supset E \ni d_E \times Q_B ,\ E \ni (d_E, Q_A) \supset E \ni d_E \times Q_A$
4) $A \triangleleft \{ Q_E, g_E \} \supset A \ni Q_E \wedge g_E,\ A \ni d_A \supset A \ni d_A \times Q \supset A \ni Q_{AK}.x,\ (Q_{AK} = d_A \times Q_E = d_E \times Q_A = d_B d_E \times P)$
5) $A \mid\equiv PK_\delta(B, Q_E) \wedge A \mid\equiv PK_\delta(A, d_A) \supset A \mid\equiv A \leftarrow Q_{AK}.x \rightarrow B$
6) $A \mid\equiv B \mid\approx (Q_E, g_E) \supset A \mid\equiv B \mid\approx (Q_E,\ g_E \times Q_A) \vdash A \mid\equiv B \mid\approx Q_{AK}.x$
7) $A \mid\equiv A \leftarrow Q_{AK}.x + \longrightarrow B$， by 4， 5 and 6
8) Similarly， $B \mid\equiv B \leftarrow Q_{BK}.x + \longrightarrow A$， $R_{BE} = (d_B \times Q_E = d_E \times Q_B = d_B d_E \times P)$， $E \mid\equiv E \leftarrow Q_{AK}.x + \longrightarrow A$， $E \mid\equiv E \leftarrow Q_{BK}.x + \longrightarrow B$。
9) $A \triangleleft C_1 \supset A \ni C_1 \wedge A \triangleleft (r_e, s_e),\ A \ni C_1 \wedge A \ni Q_{AK}.x \supset A \ni \{(r_e, s_e), t_e, e_e, g_A\}$
10) $A \mid\equiv PK_\delta(B, Q_E) \wedge A \triangleleft (r_e, s_e) \wedge \mathrm{SV}((r_e, s_e), Q_{CA}, e_e) \supset A \mid\equiv B \mid\sim e_e \supset A \mid\equiv B \mid\sim C_1$
11) $A \mid\equiv \# g_A \supset A \mid\equiv \# C_1$, and 10 can $A \mid\equiv B \mid\sim e_e \supset A \mid\equiv B \mid\approx C_1$， Similarly, $B \mid\equiv A \mid\approx C_0$， $E \mid\equiv B \mid\approx C_B \wedge A \mid\approx C_A$
12) $A \mid\equiv PK_\delta(B, g_E) \wedge PK_\delta(A, g_A) \supset A \mid\equiv E \leftarrow K_{AE} + \longrightarrow A$, where, $K_{AE} = H(Q_{AK}.x, g_A, g_E)$
13) $A \ni (Q_{AK}.x, g_A, g_E) \supset A \ni F(Q_{AK}.x, g_A, g_E) \supset A \ni K_{AE}$
14) $A \mid\equiv (A \leftarrow Q_{AK}.x \rightarrow B \wedge A \triangleleft \{C_1^B\} Q_{AK}.x) \supset A \mid\equiv B \ni Q_{AK}.x \supset A \mid\equiv B \ni g_A,\ A \mid\equiv B \ni (g_A \wedge g_E) \supset A \mid\equiv B \ni K_{AE}$
15) $A \mid\equiv \# g_A \supset A \mid\equiv \# K_{AE}$
16) $A \mid\equiv B \ni Q_{AK}.x \wedge A \mid\equiv (B \mid\sim C_1) \supset A \mid\equiv (B \mid\sim F(Q_{AK}.x, g_A, g_E)) \supset A \mid\equiv (B \mid\sim K_{AE})$
17) $A \mid\equiv (\# K_{AE} \wedge B \mid\sim K_{AE}) \supset A \mid\equiv B \mid\approx K_{AE} \equiv A \mid\equiv A \leftarrow K_{AE} + \longrightarrow B$, Similarly, $B \mid\equiv B \leftarrow K_{AE} + \longrightarrow A$, $E \mid\equiv E \leftarrow K_{AE} + \longrightarrow A$, $E \mid\equiv E \leftarrow K_{BE} + \longrightarrow B$
18) $A \mid\equiv B \mid\approx C_1 \wedge B \ni Q_{AK}.x \supset A \mid\equiv B \ni C_1 \supset A \mid\equiv B \ni K_{AE} \wedge B \mid\equiv B \leftarrow K_{AE} - \longrightarrow A$
19) $A \mid\equiv B \ni K_{AE} \wedge A \mid\equiv B \mid\equiv B \leftarrow K_{AE} - \longrightarrow A \wedge A \mid\equiv B \mid\approx K_{AE} \supset A \mid\equiv B \leftarrow K_{AE} + \longrightarrow A$
20) $A \mid\equiv B \mid\equiv \# g_B \supset A \nmid\equiv B \mid\equiv \# K_{AE}$， Similarly, $B \mid\equiv A \mid\equiv \# K_{BE}$, $E \mid\equiv A \mid\equiv \# K_{AE}$, $E \mid\equiv B \mid\equiv \# K_{BE}$

We can draw a conclusion that is K_{AE} is the agreed session key belong to user A and the adversary E, while K_{BE} is the agreed session key belong to user B and the adversary E. But both A and B think K_{AE} and K_{BE} are the session key agreed by both of them. That is the protocol can not resist unkown key-share attack.

4 Conclusions

In this paper, two axioms have been presented which we used together with the axioms and rules of the SVO logic to analyze the authentication protocol based on certificate, and we have derived two conclusions that Aydos et al.'s protocol can not resist attacks to forward security and unkown key-share attack. Moreover, their protocol does not provide mutual authentication.

References

[1] M. Aydos, Ç. K. Koç, Implementing network security protocols based on elliptic curve cryptography [A], Proceedings of the Fourth Symposium on Computer Networks[C], pages 130–139, Stanbul, Turkey, May 20–21, 1999.

[2] V. Miller, Uses of elliptic curves in Cryptography. H.C. Williams, (ed.), Advances in Cryptology-CRYPTO 85, Proceedings, Lecture Notes in Computer Science, No 218 (1985), 417-426, Springer-Verlag.

[3] P F. Syverson, P C. Van Oorschot, On unifying some cryptography protocol logics [A]. In: Proceedings of 1994 IEEE Computer Society Symposium on Security and Privacy[C]. Los A lam itos, CA, 1994:14–28.

[4] P F. Syverson, P C. Van Oorschot. A unified cryptography protocol logic [M]. TR: NRL Publication 5540–227, 1996.

[5] D. Dolev, A. Yao. On the security of public Key Protocols. IEEE Transactions on Information Theory, 29 (2):198–208, Mar, 1983.