Comments on a Secure Authentication Scheme for IoT and Cloud Servers

Wei-Liang Tai[1] and Ya-Fen Chang[2]
(Corresponding author: Ya-Fen Chang)
Department of Information Communications, Chinese Culture University[1]
55, Hwa-Kang Road, Yang-Ming-Shan, Taipei, Taiwan
Department of Computer Science and Information Engineering, National Taichung University of Science and Technology[2]
No. 129, Sec. 3, Sanmin Rd., North Dist., Taichung, Taiwan
(Email: cyf@nutc.edu.tw)
(Received May 22, 2016; revised and accepted Aug. 5 & Sept. 2, 2016)

Abstract

Recently, Kalra and Sood proposed an authentication scheme based on Elliptic Curve Cryptography (ECC) to have embedded devices and cloud servers communicate securely using HTTP cookies. After analyzing their scheme, it is found that there are five issues that are not properly addressed. In this paper, the details and further discussions are given.

Keywords: Cloud Computing; ECC; Elliptic Curve Cryptography; IoT

1 Introduction

In 2015, Kalra and Sood proposed an ECC-based authentication scheme [7]. They claimed that their scheme could ensure the security of communications between embedded devices and cloud servers. In their scheme, HTTP cookies are used for mutual authentication, and a session key will be negotiated by the embedded device and the cloud server to protect communications. This technique makes Kalra and Sood's authentication scheme different from other authentication schemes [1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12].

However, after analyzing Kalra and Sood's scheme, it is found that there are five issues that are not properly addressed.

1) An embedded system is a machine instead of a real person. Allowing an embedded system to register at the cloud server might damage the cloud server.
2) An embedded system cannot be authenticated by the cloud server because an important parameter is not issued by the cloud server.
3) Some computational operations do not comply with the definitions of ECC.
4) The session key can be computed by neither the embedded device nor the cloud server.
5) When updating or setting a cookie, the request is not verified.

Because of the above five issues, Kalra and Sood's authentication scheme cannot ensure the security of communications between embedded devices and cloud servers as claimed. The rest of this paper is organized as follows. Section 2 briefly reviews Kalra and Sood's authentication scheme for IoT and cloud servers. Section 3 implicitly shows the found issues and makes further discussions. Some conclusions are drawn in Section 4.

2 Review of Kalra and Sood's Authentication Scheme for IoT and Cloud Servers

Kalra and Sood's authentication scheme is composed of three phases, registration phase, pre-computation and login phase, and authentication phase. The notations used in Kalra and Sood's scheme are listed in Table 1. Before all phases, the cloud server S chooses an elliptic curve equation $y^2 = x^3 + ax + b$ in Z_p, where $a, b \in Z_p$ and $4a^3 + 27b^2 \bmod p \neq 0$. The order of this elliptic curve is a prime n, where $n > 2^{160}$, and O is an infinite point such that $n \times G = O$. And, the server S chooses X as its private key. The details are as follows.

2.1 Registration Phase

When a new embedded device D_i wants to access S, it needs to register at S at first. The details of this phase are as follows:

Step 1: D_i sends its unique identity ID_i to S as a registration request.

Table 1: Notations used in Kalra and Sood's authentication scheme

Symbol	Definition
D_i	An embedded device
S	The cloud server
ID_i	D_i's identity
P_i	D_i's password
R_i	A random number generated by S for D_i
N_1, N_2	Random numbers generated for ECC
$H()$	One-way hash function
X	S's private key
Z_p	A finite field
p	A prime greater than 2^{160}
G	A generator point of prime order n
CK	Cookie
EXP_TIME	CK's expiration time
$\|$	A concatenation operator
$\oplus$	An XOR operator

Step 2: After getting the registration request, S generates D_i's dedicated password P_i and a unique random number R_i. S computes cookie $CK = H(R_i \parallel X \parallel \text{EXP\_TIME} \parallel ID_i)$, $CK' = CK \times G, T_i = R_i \oplus H(X)$, and $A_i = H(R_i \oplus H(X) \oplus P_i \oplus CK')$, where CK' is an ECC point and is stored in D_i as the cookie information. Then, S stores $ID_i, A'_i = A_i \times G, T_i$, and the cookie expiration time EXP_TIME for D_i. When the cookie expires, the expiration time will be updated to EXP_TIME$'$, and the cookie CK will be updated to $H(R_i \parallel X \parallel \text{EXP\_TIME}' \parallel ID_i)$.

Step 3: S sends CK' to D_i.

2.2 Pre-computation and Login Phase

Step 1: Before each login, D_i chooses a random number N_1, computes the corresponding ECC point $P_1 = N_1 \times G$, and stores the information in its memory.

Step 2: When D_i wants to login to S, D_i computes $P_2 = H(N_1 \times CK')$ and sends $\{P_1, P_2, ID_i\}$ to S.

2.3 Authentication Phase

After getting D_i's login request $\{P_1, P_2, ID_i\}$, authentication phase is executed as follows:

Step 1: S computes $R_i = T_i \oplus H(X), CK = H(R_i \parallel X \parallel \text{EXP\_TIME} \parallel ID_i)$, and $P'_2 = H(P_1 \times CK)$. Then S checks whether P_2 and P'_2 are equal or not. If they are equal, this phase proceeds.

Step 2: S chooses a random number N_2 and computes $P_3 = N_2 \times G$ and $P_4 = N_2 \times A'_i$. Then S sends $\{P_3, P_4, T_i\}$ to D_i.

Step 3: After receiving $\{P_3, P_4, T_i\}$, D_i computes $A_i = H(T_i \oplus P_i \oplus CK')$ and $P'_4 = P_3 \times A_i$. Then D_i checks whether P_4 and P'_4 are equal or not. If they are equal, this phase proceeds.

Step 4: D_i computes $V_i = H((N_1 \times CK') \parallel P'_4)$ and sends $\{V_i\}$ to S.

Step 5: After receiving $\{V_i\}$, S computes $V'_i = H((P_1 \times CK) || P_4)$ and checks whether V_i and V'_i are equal or not. If they are equal, D_i and S authenticate each other successfully, and they can obtain the session key $SK = H(X \parallel ID_i \parallel N_1 \parallel N_2)$.

3 The Found Five Issues and Further Discussions

In this section, the details of the found issues are given, and further discussions are made.

3.1 The Found Issues

After analyzing Kalra and Sood's authentication scheme, it is found that it cannot ensure the security of communications between embedded devices and cloud servers as claimed because of the following five issues.

Issue 1: An embedded system is a machine instead of a real person. Allowing an embedded system to register at the cloud server might damage the cloud server.

When allowing an embedded system to register at a server, it denotes that a machine even a robot can register at will. An attacker can easily mount a DoS (denial-of-service) attack by registering at the server with plenty of distinct device identities to get a number of cookies and accessing the cloud server with these registered identities to consume the system resources.

Issue 2: An embedded system cannot be authenticated by the cloud server because an important parameter is not issued by the cloud server.

In registration phase, after the cloud server gets D_i's registration request, S generates a dedicated password P_i and a unique random number R_i for D_i. Then S computes the corresponding parameters CK, CK', T_i, A_i, and A'_i, and S sends CK' to D_i. In authentication phase, D_i needs to compute $Ai = H(T_i \oplus P_i \oplus CK')$ and $P'_4 = P_3 \times A_i$ to authenticate the cloud server S, and D_i needs to compute $V_i = H((N_1 \times CK') \parallel P'_4)$ to have S authenticate it. However, D_i does not know P_i because P_i is chosen by S and is not issued to D_i in registration phase. That is, D_i is not capable of computing A_i, P'_4, and V_i, and the embedded system D_i will never be authenticated successfully.

Issue 3: Some computational operations do not comply with the definitions of ECC.

In ECC, a multiplication operation is defined as $B = \alpha \times Q$, where Q and B are ECC points and α is an integer. In Kalra and Sood's scheme, computing $P_2' = H(P_1 \times CK)$, $P_4' = P_3 \times A_i$, and $V_i' = H((P_1 \times CK) \parallel P_4)$ violates the definitions of ECC.

Issue 4: The session key can be computed by neither the embedded device nor the cloud server.

In Kalra and Sood's scheme, a session key $SK = H(X \parallel ID_i \parallel N_1 \parallel N_2)$ is negotiated after mutual authentication. According to Elliptic Curve Discrete Logarithm Problem (ECDLP), it is computationally infeasible to retrieve α when Q and B are known, where $B = \alpha \times Q$, Q and B are ECC points and α is an integer. Consequently, D_i only knows the random number N_1 generated by itself because it cannot retrieve N_2 from P_3, and S only knows the random number N_2 generated by itself because it cannot retrieve N_1 from P_1, where $P_3 = N_2 \times G$ and $P_1 = N_1 \times G$. Moreover, only S knows the private key X. That is, it is impossible for both D_i and S to obtain the session key $SK = H(X \parallel ID_i \parallel N_1 \parallel N_2)$.

Issue 5: When updating or setting a cookie, the request is not verified.

Kalra and Sood's scheme uses HTTP cookies for mutual authentication. But updating or setting a cookie is not verified such that an attacker can maliciously modify the cookie stored in the embedded device D_i to make it unable to be authenticated.

3.2 Further Discussions

To remedy the found issues, some modifications should be made. First, a user instead of an embedded device can register at the cloud server to prevent an attacker from registering at the server with distinct device identities and consuming the system resources. Second, no matter who chooses the password P_i, both the embedded device D_i and the cloud server S need to know P_i, and the party choosing P_i should transmit P_i to the other via a secure channel. Third, P_2', P_4', and V_i' should be computed as $P_2' = H(CK \times P_1)$, $P_4' = A_i \times P_3$, and $V_i' = H((CK \times P_1) \parallel P_4)$, respectively. Forth, the session key SK can be $H(ID_i \parallel N_1N_2 \times G)$, where D_i computes $N_1 \times P_3 = N_1N_2 \times G$ and S computes $N_2 \times P_1 = N_1N_2 \times G$. Fifth, the path of setting the cookie should be dedicated to the embedded device to prevent an attacker from modifying the cookie stored in the embedded device.

4 Conclusions

After analyzing Kalra and Sood's scheme, it is found that five issues are not well addressed. In this paper, the details of these issues are shown with further discussions to remedy them. With these modifications, Kalra and Sood's scheme can be improved to ensure the security of communications between embedded devices and cloud servers.

Acknowledgments

This work was supported in part by Ministry of Science and Technology under the Grants MOST 104-2221-E-034-004-, MOST 104-2221-E-025-006-, and MOST 105-2221-E-034-014-.

References

[1] C. C. Chang, W. Y. Hsueh, and T. F. Cheng, "An advanced anonymous and biometrics-based multi-server authentication scheme using smart cards," *International Journal of Network Security*, vol. 18, no. 6, pp. 1010–1021, 2016.

[2] T. Y. Chang, W. P. Yang, and M. S. Hwang, "Simple authenticated key agreement and protected password change protocol," *Computers & Mathematics with Applications*, vol. 49, no. 5, pp. 703–714, 2001.

[3] Y. F. Chang, "Flexible access control over verifiable cloud computing services with provable security," *Informatica*, vol. 26, no. 2, pp. 181–198, 2015.

[4] Y. F. Chang, W. L. Tai, and H. C. Chang, "Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 3430–3440, 2014.

[5] M. S. Hwang, C. C. Lee, and Y. L. Tang, "An improvement of SPLICE/AS in WIDE against guessing attack," *International Journal of Informatica*, vol. 12, no. 2, pp. 297–302, 2001.

[6] Q. Jiang, J. Ma, G. Li, and L. Yang, "Robust two-factor authentication and key agreement preserving user privacy," *International Journal of Network Security*, vol. 16, no. 3, pp. 229–240, 2014.

[7] S. Kalra and S. K. Sood, "Secure authentication scheme for iot and cloud servers," *Pervasive and Mobile Computing*, vol. 24, pp. 210–223, 2015.

[8] C. W. Lin and M. S. Hwang C. S. Tsai, "A new strong-password authentication scheme using one-way hash functions," *International Journal of Computer and Systems Sciences*, vol. 45, no. 4, pp. 623–626, 2006.

[9] C. H. Ling, C. C. Lee, C. C. Yang, and M. S. Hwang, "A secure and efficient one-time password authentication scheme for WSN," *International Journal of Network Security*, vol. 19, no. 2, pp. 177–181, 2017.

[10] P. K. Thandra, J. Rajan, and S. A. V. S. Murty, "Cryptanalysis of an efficient password authentication scheme," *International Journal of Network Security*, vol. 18, no. 2, pp. 362–368, 2016.

[11] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password authentication schemes: Current status and

key issues," *International Journal of Network Security*, vol. 3, no. 2, pp. 101–115, 2006.

[12] J. Wei, W. Liu, and X. Hu, "Secure and efficient smart card based remote user password authentication scheme," *International Journal of Network Security*, vol. 18, no. 4, pp. 782–792, 2016.

Biography

Wei-Liang Tai biography. Wei-Liang Tai received the M.S. degree in computer science and information engineering from National Chung Cheng University, Taiwan, in 2004 and the Ph.D. degree in computer science and information engineering from National Chung Cheng University, Taiwan, in 2008. He is currently Associate Professor, Department of Information Communications, Chinese Culture University. His main interests are in information security and forensics and multimedia signal processing. He is currently an Editor of KSII Transactions on Internet and Information Systems.

Ya-Fen Chang biography. Ya-Fen Chang is a professor of Department of Computer Science and Information Engineering at National Taichung University of Science and Technology in Taiwan. She received her BS degree in computer science and information engineering from National Chiao Tung University and PhD degree in computer science and information engineering from National Chung Cheng University, Taiwan. Her current research interests include electronic commerce, information security, cryptography, mobile communications, image processing, and data hiding.