Comments on a password authentication and update scheme based on elliptic curve cryptography

Debiao He

School of Mathematics and Statistics, Wuhan University, Wuhan, People's Republic of China

Email: hedebiao@163.com

Abstract*:* The security of a password authentication and update scheme based on elliptic curve cryptography proposed by Islam et al. [S.K. Hafizul Islam, G.P. Biswas, Design of improved password authentication and update scheme based on elliptic curve cryptography, *Mathematical and Computer Modelling* (2011), doi:10.1016/j.mcm.2011.07.001] is analyzed. Three kinds of attacks are presented in different scenarios.

Key words*: Password authentication, Elliptic curve cryptography, offline password guessing attack, stolen-verifier attack and privileged insider attack*

1. Introduction

Secured password authentication and update of passwords are two essential requirements for remote login over unreliable networks. Recently, Islam et al. proposed a password authentication and update scheme based on elliptic curve cryptography to satisfy the above two requirements. They claimed that their scheme could withstand various attacks. In this letter, however, some security loopholes of their scheme will be pointed out and the corresponding attacks will be described.

The organization of the letter is sketched as follows. The Section 2 gives a brief review of Islam et al.'s scheme. The security flaws of Islam et al.'s scheme are shown in Section 3. Finally, we give some conclusions in Section 4.

2. Islam et al.'s scheme

In this section, we will briefly review Islam et al.'s scheme. Their scheme consists of four phases: Registration phase, Password authentication phase, Password change phase and Session key distribution phase. In order to facilitate future references, frequently used notations are listed below with their descriptions.

- ID_A: Identity of the client A;
- pw_A: Secret password of the client A.
- d_S: Secret key of the server S.
- U_S: Public key of the server S, where $U_S = d_S \cdot G$.
- U_A: Password-verifier of the client A, where $U_A = pw_A \cdot G$.
- K_x: Secret key computed either using $K = pw_A \cdot U_S = (K_x, K_y)$ or $K = d_S \cdot U_A = (K_x, K_y)$.
- $E_{K_x}(\cdot)$: Symmetric encryption (AES) with K_x.
- G: Bases point of the elliptic group of order n such that $n \cdot G = O$, where n is a large prime number.
- $H(\cdot)$: A collision-resistant one-way secure hash function.
- r_A / r_S: Random numbers chosen by the client/server from $[1, n-1]$ respectively.
- $+/-$: Elliptic curve point addition/subtraction.
- $e(\cdot)$: Bilinear mapping on elliptic curves.

2.1. Registration phase

In this phase, everyone who wants to register at the server should submit his identity and password-verifier to the server. The detail of the phase is described as follows.

1) The client A chooses his identity ID_A, password pw_A, computes the password-verifier $U_A = pw_A \cdot G$ and sends ID_A and U_A to the server S.

2) After receiving ID_A and U_A, S stores ID_A, U_A, and a $status-bit$ in a write protected file, where the $status-bit$ indicates the status of the client, i.e., when the client is logged-in to the server the status-bit is set to one, otherwise it is set to zero.

2.2. Password authentication phase

Once the client A wants to login to the server S, as shown in Fig. 1, he will perform the following login steps.

1) The client A inputs his identity ID_A and the password pw_A into the terminal. The client selects a random number $r_A \in [1, n-1]$, computes $R_A = r_A \cdot U_S$, $W_A = (r_A \cdot pw_A) \cdot G$ and $K = pw_A \cdot U_S = (K_x, K_y)$. Then A sends the message $M_1 = \{ID_A, E_{K_x}(ID_A, R_A, W_A)\}$ to S.

2) Upon receiving the message M_1, S computes $K = d_S \cdot U_A = (K_x, K_y)$, gets ID'_A, R'_A, W'_A by decrypting $E_{K_x}(ID_A, R_A, W_A)$ using K_x. Then the server checks whether the equations $ID_A = ID'_A$ and $e(R'_A, U_A) = e(W'_A, U_S)$ hold. If one of the equations does not hold, S stops the session. Otherwise, S selects a random number $r_S \in [1, n-1]$, computes $W_S = r_S \cdot U_S$ and sends $M_2 = \{W'_A + W_S, H(W_S)\}$ to A.

3) Upon receiving the message M_2, A computes $W'_S = (W'_A + W_S) - W_A$ and checks whether the equation $H(W'_S) = H(W_S)$ holds. If the equation does not hold, A stops the session. Otherwise, A sends the message $M_3 = \{H(W_A, W'_S)\}$ to S.

4) Upon receiving the message M_3, S checks whether the equation $H(W_A, W'_S) = H(W'_A, W_S)$ holds. If the equation does not hold, A rejects the client login request. Otherwise, A granted the client's login request.

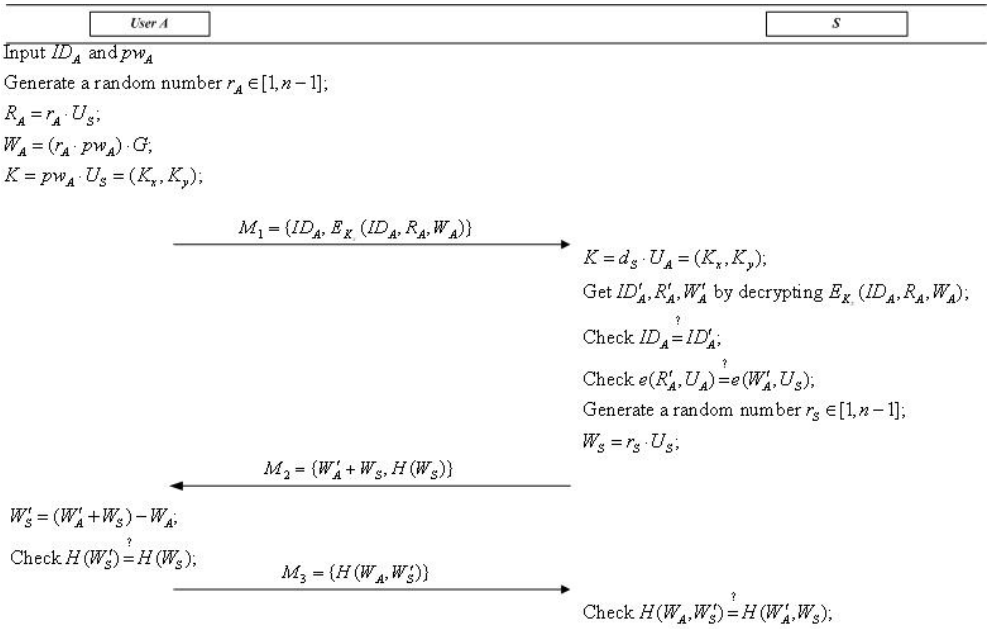

Fig. 1. Password authentication phase

2.3. Password change phase

This phase will be invoked if the client wants to change his password from pw_A to pw'_A.

The client A and the server S first execute steps 1) and 2) in subsection 2.2. Then the following steps will be executed.

3) Upon receiving the message M_2, A computes $W'_S = (W'_A + W_S) - W_A$ and checks whether the equation $H(W'_S) = H(W_S)$ holds. If the equation does not hold, A stops the session. Otherwise, A inputs the new password pw'_A, computes $U'_A = pw'_A \cdot G$ and sends the message $M_3 = \{ID_A, H(W_A, W'_S), W_A + U'_A, H(W'_S, U'_A)\}$ to S.

4) Upon receiving the message M_3, S checks whether the equations $H(W_A, W'_S) = H(W'_A, W_S)$ and $H(W'_S, U'_A) = H(W_S, (W_A + U'_A) - W'_A)$ holds. If the equation does not hold, A rejects the client's request. Otherwise, A accepts the request and replaces U_A with $W_A + U'_A) - W'$.

2.4. Session key distribution phase

Once the client A wants to login to the server S and generate a session key for future communication, the phase will be executed.

The phase is similar with the password authentication phase described in subsection 2.2. The following two steps are added to 3) and 4) separately to generate the session key.

- The client computes the final session key as $SK = (r_A \cdot pw_A) \cdot W_S = (r_A \cdot r_S \cdot pw_A \cdot d_S) \cdot G$
- The server computes $SK = (r_S \cdot d_S) \cdot W_A = (r_A \cdot r_S \cdot pw_A \cdot d_S) \cdot G$.

3. Cryptanalysis of Islam et al.'s scheme

We assume that an attacker $\mathscr{A}$ has total control over the communication channel between the client A and the remote server S, which means that he can insert, delete, or alter any messages in the channel. We shall prove that Islam et al.'s scheme is vulnerable to offline password guessing attacks, stolen-verifier

attack and privileged insider attack. A more detailed description of attacks is as follows.

3.1 Offline password guessing attack

In password authentication and update schemes that the client is allowed to choose his password, the client tends to choose a password that can be easily remembered for his convenience. However, these easy-to-remember passwords are potentially vulnerable to password guessing attack, in which an adversary can try to guess the client's password and then verify his guess. In general, the password guessing attack can be classified into online password guessing attack and offline password guessing attack. The adversary tries to use guessed passwords iteratively to pass the verification of the server in an online manner in online password guessing attack. While in offline password attack, the adversary intercepts some password-related messages exchanged between the client and the server, and then iteratively guesses the client's password and verifies whether his guess is correct or not in an offline manner. Online password guessing attacks can be easily thwarted by limiting the number of continuous login attempts within a short period. In an offline password guessing attack, since there is no need for the server to participate in the verification, the server cannot easily notice the attack.

Although Islam et al. claimed that their scheme could withstand password guessing attack [1], an offline password attack method will be given here as a counter example. Assume the attacker $\mathscr{A}$ intercepts the message $M_1 = \{ID_A, E_{K_x}(ID_A, R_A, W_A)\}$ transmitted between the client A and the server S, where $R_A = r_A \cdot U_S$, $W_A = (r_A \cdot pw_A) \cdot G$ and $K = pw_A \cdot U_S = (K_x, K_y)$. Then the attacker $\mathscr{A}$ can successfully find out A's password pw_A by performing the following procedure.

1) $\mathscr{A}$ guess a password pw'_A and computes $K' = pw'_A \cdot U_S = (K'_x, K'_y)$.

2) $\mathscr{A}$ gets ID'_A, R'_A, W'_A by decrypting $E_{K_x}(ID_A, R_A, W_A)$ using key K'_x.

3) $\mathscr{A}$ checks whether ID'_A and ID_A are equal. If they are equal, $\mathscr{A}$ find the correct password. Otherwise, $\mathscr{A}$ repeats steps 1), 2) and 3) until finding the correct password.

From the above description, we know the adversary can get the password. Therefore, Islam et al.'s scheme is vulnerable to the offline password guessing attack.

3.2. Stolen-verifier attack

The stolen-verifier attack, which is described in [2], means that an outsider theft the password-verifier from the server's database and applies an off-line guessing attack on it to get the client's exact password and hence, he can impersonate a legitimate client. In Islam et al.'s scheme [1], the client A registers to the remote server S with identity ID_A, the password-verifier $U_A = pw_A \cdot G$ and S then stores the pair (ID_A, U_A) to the database. The outsider $\mathscr{A}$ can successfully find out A's password pw_A by performing the following procedure.

1) $\mathscr{A}$ steals (ID_A, U_A) from the S's database and tries to find out A's password pw_A by using an off-line password guessing attack on stolen U_A.

2) $\mathscr{A}$ guess a password pw'_A and computes $U'_A = pw'_A \cdot G$.

3) $\mathscr{A}$ checks whether U'_A and U_A are equal. If they are equal, $\mathscr{A}$ find the correct password. Otherwise, $\mathscr{A}$ repeats steps 1), 2) and 3) until finding the correct password.

From the above description, we know the adversary can get the password. Therefore, Islam et al.'s scheme is vulnerable to the stolen-verifier attack.

3.3. Privileged Insider attack

In a real environment, it is a common practice that many users use same passwords to access different applications or servers for their convenience of remembering long passwords and ease-of-use whenever required[3]. However, if the system manager or a privileged insider $\mathscr{A}$ of the server S knows the passwords of client A, he may try to impersonate A by accessing other servers where A could be a registered user. In the user registration phase of Islam et al.'s scheme, A sends his identity ID_A, the password-verifier $U_A = pw_A \cdot G$ to S. Although, the password pw_A is not directly transmitted to the system, the privileged-insider of S could get the password through the off-line password

guessing attack. The detail of the off-line password guessing attack is described as follows.

1) $\mathscr{A}$ guesses a password pw'_A and computes $U'_A = pw'_A \cdot G$.

2) $\mathscr{A}$ checks whether U'_A and U_A are equal. If they are equal, $\mathscr{A}$ find the correct password. Otherwise, $\mathscr{A}$ repeats steps 1) and 2) until finding the correct password.

From the above description, we know the adversary can get the password. Therefore, Islam et al.'s scheme is vulnerable to the privileged insider attack.

4. Conclusion

In [1], Islam et al. proposed a password authentication and update scheme based on elliptic curve cryptography and demonstrated its immunity against various attacks. However, after review of their scheme and analysis of its security, three kinds of attacks, i.e., offline password guessing attack, stolen-verifier attack and privileged insider attack are presented in different scenarios. The analyses show that the scheme is insecure for practical application.

Reference

[1]. S.K. Hafizul Islam, G.P. Biswas, Design of improved password authentication and update scheme based on elliptic curve cryptography, *Mathematical and Computer Modelling* (2011), doi:10.1016/j.mcm.2011.07.001

[2]. W.C. Ku, M.H. Chaing, S.T. Chang, Weaknesses of Yoon-Ryu-Yoo's hash-based password authentication scheme, *ACM Operating Systems Review*, 39(1) (2005) 85-89.

[3]. H.C.Hsiang, W.K. Shiha, Improvement of the secure dynamic ID based remote user authentication next term scheme for multi-server environment, Computer Standards & Interfaces, 31(6) (2009) 1118-1123.