Cryptanalysis of Chang-Chang’s EC-PAKA protocol for wireless mobile networks

Hae-Soon Ahn, Member, IEEE, and Eun-Jun Yoon, Member, IEEE

Abstract—With the rapid development of wireless mobile communication, applications for mobile devices must focus on network security. In 2008, Chang-Chang proposed security improvements on the Lu et al.’s elliptic curve authentication key agreement protocol for wireless mobile networks. However, this paper shows that Chang-Chang’s improved protocol is still vulnerable to off-line password guessing attacks unlike their claims.

II. REVIEW OF CHANG-CHANG’S EC-PAKA PROTOCOL

This section briefly reviews Chang-Chang’s EC-PAKA protocol [5]. The following notations are used throughout this paper.

- $A$, Bob($B$): two communication users;
- $E$: an elliptic curve defined over a finite field $A$ with large group order;
- $n$: a secure large prime;
- $P$: a point in $E$ with large order $n$;
- $D$: a uniformly distributed dictionary of size |$D$|;
- $S$: a low-entropy password shared between Alice and Bob, which is randomly chosen from $D$;
- $t$: the value $t$ is derived from the password $S$ in a predetermined way, which is uniformly distributed in $Z_n^*$;
- $H(\cdot)$: a secure one-way hash function;
- $||$: concatenation of messages.

Fig. 1 depicts the Chang-Chang’s EC-PAKA protocol, which works as follows.

Step 1. $A \rightarrow B$: $\{Q_{A1}, Q_{A2}\}$

$A$ first chooses a random number $d_A \in [1, n-1]$, and then computes the followings:

$Q_{A1} = (d_A + t)P$ \hspace{1cm} (1)

$Q_{A2} = d_A^2P$ \hspace{1cm} (2)

Finally, $A$ sends the message $\{Q_{A1}, Q_{A2}\}$ to $B$.

Step 2. $B \rightarrow A$: $\{H_B, Q_{B1}\}$

Upon receiving the message $\{Q_{A1}, Q_{A2}\}$, $B$ also chooses two random numbers $d_{B1}, d_{B2} \in [1, n-1]$, and then computes the followings:

$Y = Q_{A1} - tP = d_A P$ \hspace{1cm} (3)

$Q_{B1} = d_{B1}P + d_{B2}Y$ \hspace{1cm} (4)

$Q_{B2} = d_{B1}Y + d_{B2}Q_{A2}$ \hspace{1cm} (5)

$H_B = H(A||B||Q_{A1}||Q_{B1}||Y)$ \hspace{1cm} (6)

Finally, $B$ sends $\{H_B, Q_{B1}\}$ and to $A$.

Step 3. $A \rightarrow B$: $\{H_A\}$

Upon receiving the message $\{H_B, Q_{B1}\}$, $A$ checks whether the equality

$H(A||B||Q_{A1}||Q_{B1}||Y) \overset{?}{=} H_B$ \hspace{1cm} (7)

holds or not. If it holds, $A$ computes and sends

$H_A = H(B||A||Q_{B1}||Q_{B2}||d_AP)$ \hspace{1cm} (8)

to $B$. Then, $A$ computes

$X = d_AQ_{B1} = d_{B1}d_AP + d_{B2}d_A^2P$ \hspace{1cm} (9)

and sets the session key as

$K_A = X$ \hspace{1cm} (10)
Step 4. Upon receiving the message \( \{ H_A \} \), \( B \) checks whether the equality
\[
H(B||A||Q_{B1}||Q_{A1}||Y) \overset{?}{=} H_A
\]
holds or not. If it holds, \( B \) sets the session key as
\[
K_B = Q_{B2}
\]

III. Cryptanalysis of Chang-Chang’s EC-PAKA Protocol

This section shows that Chang-Chang’s EC-PAKA protocol is still vulnerable to off-line password guessing attacks unlike their claims. Password-based AKA protocols can be vulnerable to password guessing attacks because users usually choose easy-to-remember passwords. Unlike typical private keys, the password has limited entropy, and is constrained by the memory of the user. For example, one alphanumerical character has 6 bits of entropy, and thus the goal of the password guessing attacks can be divided into three classes as follow[6], [7], [8]:

1) Detectable on-line password guessing attacks: an attacker attempts to use a guessed password in an on-line transaction. He/she verifies the correctness of his/her guess using the response from server. A failed guess can be detected and logged by the server.
2) Undetectable on-line password guessing attacks: similar to above, an attacker tries to verify a password guess in an online transaction. However, a failed guess cannot be detected and logged by the server, as the server cannot distinguish between an honest request and an attacker’s request.
3) Off-line password guessing attacks: an attacker guesses a password and verifies his/her guess off-line. No participation of server is required, so the server does not notice the attack as a malicious one.

Based on the above definitions of password guessing attacks, we define the security term needed for security problem analysis of the Chang-Chang’s EC-PAKA protocol as follows:

Definition 1: A weak secret (password \( S \)) is a value of low entropy \( \text{Weak}(k) \), which can be guessed in polynomial time.

An adversary \( \text{Eve} \) can perform the following off-line password guessing attack. Let us assume that an adversary \( \text{Eve} \) has intercepted one of the \( A \) and \( B \)’s past communication messages, i.e., \( \{ Q_{A1}, Q_{A2}, H_{B}, Q_{B1}, H_{A} \} \). Then \( \text{Eve} \) can perform an off-line password guessing attack to obtain the password \( S \) as follows:

1) \( \text{Eve} \) generates a candidate password \( S^* \) from password dictionary which called \( D \).
2) \( \text{Eve} \) derives the value \( t^* (\in Z_n^*) \) from the guessed password \( S^* \).
3) \( \text{Eve} \) obtains \( d_A P^* \) by computing
\[
d_A P^* = Q_{A1} - t^* P
\]

where \( Q_{A1} = d_A P + tP \).
4) \( \text{Eve} \) computes \( H_B^* \) as follows:
\[
H_B^* = H(A||B||Q_{A1}||Q_{B1}||d_A P^*)
\]
5) \( \text{Eve} \) compares \( H_B^* \) with the intercepted \( H_B \).
6) If \( H_B^* \) is equal to \( H_B \), then \( \text{Eve} \) has guessed the correct password \( S^* \), otherwise, \( \text{Eve} \) performs steps 1-5 repeatedly until \( H_B^* \equiv H_B \) by choosing another password \( S^{**} \).

The algorithm of the off-line password guessing attacks for getting the password \( S \) is as follows:

\[
\text{Off-line Password Guessing Attacks}(Q_{A1}, H_B, D) \{
\text{for } i := 0 \text{ to } |D| \{
S^* \leftarrow D; \\
t^*(\in Z_n^*) \leftarrow S^*; \\
d_A P^* = Q_{A1} - t^* P; \\
H_B^* = H(A||B||Q_{A1}||Q_{B1}||d_A P^*); \\
\text{if } H_B^* = H_B \text{ then return } S^*
\}
\}
\]

After the adversary \( \text{Eve} \) has obtained the user \( A \)’s password \( S^* \) using the above off-line password guessing attack method, the adversary \( \text{Eve} \) can impersonate \( A \) by forging \( A \)’s sending message \( \{ Q_{A1} = (d_{\text{Eve}} + t)P, Q_{A2} = d_{\text{Eve}}^2 P \} \), where \( d_{\text{Eve}} \) is a random number \( \in [1, n - 1] \). Therefore, Chang-Chang’s EC-PAKA protocol is vulnerable to off-line password guessing attacks.

Real applications for the proposed off-line password guessing attacks are as follows: Passwords are the most common methods of user authentication and key agreement on the Internet or mobile environments. For practical security applications, AKA protocols are required when making use of Internet or mobile network services like E-learning, on-line polls, on-line ticket-order systems, roll call systems, on-line games, etc. In real security applications, users offer the same password as above to access several application servers for their convenience [8]. Thus, an adversary \( \text{Eve} \) may try to use the guessed password \( S \) to impersonate the legal user \( A \) to login to other systems that the user \( A \) has registered with outside this Chang-Chang’s EC-PAKA protocol-based server. If the targeted outside server adopts the normal authentication and key agreement protocol, it is possible that the adversary \( \text{Eve} \) can successfully impersonate the user \( A \) to login to it by using the guessed password \( S \). Therefore, the password breach cannot be revealed by the adversary’s actions.

IV. Conclusions

The EC-PAKA technology has been widely deployed in various kinds of applications. This paper demonstrated that Chang-Chang’s EC-PAKA protocol is still insecure to off-line password guessing attacks. For this reason, Chang-Chang’s
Choose random $d_A \in [1, n-1]$
Compute $Q_{A1} = (d_A + t)P$
Compute $Q_{A2} = d_A^2P$
\[\{Q_{A1}, Q_{A2}\}\]

Choose random $d_B_1, d_B_2 \in [1, n-1]$
Compute $Y = Q_{A1} - tP = d_A P$
Compute $Q_{B1} = d_B_1 P + d_B_2 Y$
Compute $Q_{B2} = d_B_1 Y + d_B_2 Q_{A2}$

Verify $H(A||B||Q_{B1}||Y) = H_B$
Compute $H_A = H(B||A||Q_{B1}||Q_{A1}||d_A P)$
Compute $X = d_A Q_{B1} = d_B_1 d_A P + d_B_2 d_A^2 P$
Set session key $K_A = X$
\[\{H_A\}\]

Verify $H(B||A||Q_{B1}||Q_{A1}||Y) = H_A$
Set session key $K_B = Q_{B2}$

Fig. 1. Chang-Chang’s EC-PAKA protocol for wireless mobile networks

EC-PAKA protocol cannot use for practical application, especially in the resource-limited environments and real-time systems. Further works will be focused on improving the Chang-Chang’s EC-PAKA protocol which can be able to provide greater security and to be more efficient than the existing EC-PAKA protocols by an accurate performance analysis.

ACKNOWLEDGEMENTS
We would like to thank the anonymous reviewers for their helpful comments in improving our manuscript.

REFERENCES