Cryptanalysis of an Authenticated Key Agreement Protocol for Wireless Mobile Communications

With the rapid progress of wireless mobile communications, the authenticated key agreement (AKA) protocol has attracted an increasing amount of attention. However, due to the limitations of bandwidth and storage of the mobile devices, most of the existing AKA protocols are not suitable for wireless mobile communications. Recently, Lo and others presented an efficient AKA protocol based on elliptic curve cryptography and included their protocol in 3GPP2 specifications. However, in this letter, we point out that Lo and others’ protocol is vulnerable to an offline password guessing attack. To resist the attack, we also propose an efficient countermeasure.

Keywords: Authenticated key agreement, offline password guessing attack, wireless mobile communication, 3GPP2.

I. Introduction

In 2005, Sui and others proposed an improved authenticated key agreement (AKA) protocol [1] for wireless mobile communications. Their scheme provided perfect forward secrecy but was vulnerable to an offline password attack [2]. To enhance the security, Lu and others proposed an enhanced AKA protocol for wireless mobile communications in 2007 [2]. Later, Chang and others [3] pointed out that the Lu and others’ scheme cannot resist the parallel guessing attack. Chang and others [3] proposed an improved protocol. However, Lo and others [4] demonstrated that Chang and others’ protocol does not offer the mutual authentication property. Lo and others also proposed an improved scheme using elliptic curve cryptography (ECC), included their protocol in 3GPP2 specifications, and claimed their scheme could withstand various attacks. However, in this letter, we will propose an offline password guessing attack against Lo and others’ protocol. To withstand the attack, we also propose an efficient countermeasure.

II. Review of Lo and Others’ Scheme

For convenience, the abbreviations and notations used in this letter are shown in Table 1.

| A, B | Abbreviation/identity of the participants Alice (client) and Bob (server), individually |
| E   | Elliptic curve defined over a finite field $F_q$ with large group order |
| n   | Secure large prime |
| P, Q| Two points on $E$ with large order $n$ |
| D   | Uniformly distributed dictionary of size $| D |$ |
| S   | Low-entropy password shared between Alice and Bob, which is randomly chosen from $D$ |
| t   | Integer derived from the password $S$ in a predetermined way |
| H   | Secure one-way hash function |

Table 1. Notations.
(B).

Step 2. B also chooses a random number \(d_y \in [1, n-1]\) and then computes \(Q_y = (d_y - t)P\), and then computes \(H_y = H(K_y \parallel Y)\). Then, \(B\) sends the \(Q_y\) and \(H_y\) to \(A\).

Step 3. A computes \(X = Q_y + tP\) and then computes \(K_y = d_yX\). Next, \(A\) verifies the equality of \(H(K_y \parallel d_yP)\) and \(H_y\). If it does not hold, the protocol is terminated. Otherwise, \(A\) sends \(H_y = H(K_y \parallel X)\) to \(B\) and sets the session key to \(K_y\).

Step 4. When \(B\) receives the message, it checks the equality of \(H(K_y \parallel d_yP)\) and \(H_y\). Only if the equality holds, \(B\) agrees on the session key \(K_y\). Otherwise, \(B\) terminates the protocol.

III. Weakness in Lo and Others’ Protocol

An offline password guessing attack succeeds when there is information in communications that can be used to verify the correctness of the guessed passwords. Lo and others claimed that their protocol can resist offline password guessing attacks. However, in this section, we will show that the offline password guessing attack, contrary to their claim, is still effective in Lo and others’ protocol. Our attack consists of two phases.

First phase.
1) The adversary \(\mathcal{A}\) generates a random number \(d_x \in [1, n-1]\), computes \(Q_x = d_xP\), and sends \(A\) and \(Q_x\) to \(B\).

2) Upon receiving \(A\) and \(Q_x\), \(B\) also chooses a random number \(d_y \in [1, n-1]\) and then computes \(Q_y = (d_y - t)P\), \(Y = Q_y - tP\), \(K_y = d_yY\), and \(H_y = H(K_y \parallel Y)\). Next, \(B\) sends the \(Q_y\) and \(H_y\) to \(A\).

Upon receiving \(Q_y\) and \(H_y\), the adversary \(\mathcal{A}\) carries out the second phase as follows.

Second phase.
1) The adversary \(\mathcal{A}\) guesses a password \(S'\) from \(D\) and derives the corresponding \(t'\).

2) \(\mathcal{A}\) computes \(Y' = Q_y - t'P\), \(d_x'P = Q_x + t'P\), \(d_y'P = d_y(d_y' - t' - P)\), and \(K_y' = d_yY' = d_y(d_y' - t' - P)\). Upon receiving \(Q_y\) and \(H_y\), \(B\) also chooses a random number \(d_y \in [1, n-1]\) and then computes \(Q_y = d_yP - tQ\), \(Y = Q_y - tQ\), \(K_y = d_yY\), and \(H_y = H(K_y \parallel Y)\). Next, \(B\) sends the \(Q_y\) and \(H_y\) to \(A\).

Upon receiving \(Q_y\) and \(H_y\), the adversary \(\mathcal{A}\) computes \(d_y'P = Q_x + t'Q\) and \(d_y'd_y'P = d_y(d_y'P)\), \(t'd_y'P = t'(d_y'P)\), and \(K_y' = d_yY = d_y(d_y'P - tQ)\). Upon receiving \(Q_y\) and \(H_y\), \(B\) also chooses a random number \(d_y \in [1, n-1]\) and then computes \(Q_y = d_yP - tQ\), \(Y = Q_y - tQ\), \(K_y = d_yY\), and \(H_y = H(K_y \parallel Y)\). Next, \(B\) sends the \(Q_y\) and \(H_y\) to \(A\).

3) \(\mathcal{A}\) verifies the equality of \(H(K_y' \parallel Y')\) and \(H_y\). If it does hold, the adversary gets the correct password. Otherwise, \(\mathcal{A}\) repeats 1), 2), and 3) until finding the correct password.

From the above description, we know the adversary can get the correct password. Therefore, Lo and others’ scheme is vulnerable to the offline password guessing attack.

IV. Countermeasure

In Lo and others’ scheme, the session key is simply a linear combination of \(d_yP, d_yP,\) and \(tP\). The adversary can deduce the session key upon identifying two out of the three values correlating to \(d_y\), \(d_y\), and \(t\). Then, having guessed what the password might be, the adversary can verify whether or not the guess is correct. To withstand such an attack, we introduce another point, \(Q\) on \(E\), to introduce complexity to the relationships in the session key.

First, the system selects a random point \(Q\) on \(E\). However, \(Q\) is an important parameter and should be chosen carefully to ensure that it is computationally difficult for an adversary to find the discrete logarithm of \(Q\) with \(P\) as the base. Otherwise, the protocol will be insecure.

Step 1. \(A\) picks a random number \(d_x \in [1, n-1]\) and computes \(Q_x = d_xP + tQ\). Then, \(A\) sends out its identity \(A\) and \(Q_x\) to \(B\).

Step 2. \(B\) also chooses a random number \(d_y \in [1, n-1]\) and then computes \(Q_y = d_yP - tQ\), \(Y = Q_y - tQ\), \(K_y = d_yY\), and \(H_y = H(K_y \parallel Y)\). Next, \(B\) sends the \(Q_y\) and \(H_y\) to \(A\).

Step 3. \(A\) computes \(X = Q_y + tQ\) and then computes \(K_y = d_yX\). Next, \(A\) verifies the equality of \(H(K_y \parallel d_yP)\) and \(H_y\). If it does not hold, the protocol is terminated. Otherwise, \(A\) sends \(H_y = H(K_y \parallel X)\) to \(B\) and sets the session key to \(K_y\).

Step 4. When \(B\) receives the message, it checks the equality of \(H(K_y \parallel d_yP)\) and \(H_y\). Only if the equality holds, \(B\) agrees on the session key \(K_y\). Otherwise, \(B\) terminates the protocol.

V. Conclusion

In this letter, we reviewed Lo and others’ protocol [4] and showed that their protocol cannot resist an offline password guessing attack. We then demonstrated how to fix the protocol to ensure that it is robust against attacks.
Acknowledgements

The author thanks Dr. Kyu-Seok Lee, Ms. Julie Turner, and the anonymous reviewers for their valuable comments.

References


