Cryptanalysis of Noel McCullagh and Paulo S. L. M. Barreto's two-party identity-based key agreement

Guohong Xie
School of Mathematics Science,
Peking University,People's Republic of China
xgh@pku.edu.cn

Abstract

Noel McCullagh and Paulo S. L. M. Barreto[1] proposed a two-party identity-based key agreement protocol in 2004,which can be used in either escrowed or escrowless mode. They also described conditions under which users of different Key Generation Centres can agree on a shared secret key. In this paper, we show that these two protocols are insecure against the key compromis impersonate attack,and the fix protocol has not the property of Perfect-Forword-Secrecy.We modify these protocols in three ways,which are secure against all attack and satisfy the property of Known-Key Security**,** Perfect-Forward-Secrecy, Key-Compromise Impersonation, Unknown Key-Share,and Key control and so on.

Keywords: Cryptanalysis, Weil Pairing, ID-based, Key Agreement, Authentication

1 Introduction

Traditionally, asymmetric cryptographic schemes are based on either the discrete logarithm problem or the factoring problem. The most concerned issue in implementation of cryptographic schemes is the computation cost of modular exponentiation. To overcome such a problem, the elliptic curve cryptography becomes a good choice because it reduces the computation cost while remaining the same security level. It is noticed that the decision of Diffie-Hellman is regarded as a hard problem in discrete logarithm, however, it is not a hard problem in elliptic curve cryptography due to Weil/Tate pairing. Weil/Tate pairing is a new primitive and is interesting to cryptography societies. Several cryptographic schemes are designed based on the Weil pairing,and enjoy the convenience of the property of Weil/Tate pairing. For a sound key exchange protocol, Wilson and Menezes defined several desirable security attributes[2]. We show these attributes in the following. Here we assume A and B are two honest entities.

Known-Key Security

In each round of key agreement protocol, A and B should generate a unique secret key. Each key generated in one protocol round is independent and should not be exposed if other secret keys are compromised.

Forward Secrecy

The Forward Secrecy property is that if A and B's secret keys are compromised, the session keys used in the past should not be recovered.

Key-Compromise Impersonation

A protocol which is secure against the key compromise impersonation attack means that if A's secret key is compromised, the adversary who knows the value can not impersonate others to A.

Unknown Key-Share

After the protocol, A ends up believing he shares a key with B, and B mistakenly believes that the key is instead shared with an adversary. Therefore, a sound authenticated key agreement protocol should prevent the unknown key-share situation.

In this paper, we show Noel McCullagh and Paulo S. L. M. Barreto's protocols are insecure against the key-compromise impersonation attack .The rest of this paper is organized in the following. In Section 2, we briefly review Noel McCullagh and Paulo S. L. M. Barreto's two part key agreement protocol from Weil/Tate pairing and show its insecurity. In Section 3, we proposed a scheme to prevent the protocol from the key impromise impersonate attack and conclude this paper.

2.Noel McCullagh and Paulo S. L. M. Barreto's protocol

2.1 An Authenticated Key Agreement With Escrow

as with all other identity-based cryptosystems there existence of a trusted Key Generation Centre (KGC) that is responsible for the creation and secure distribution of users private keys. This agreement algorithm uses the modified Tate pairing,

Setup:

The KGC randomly generates a master secret s, and picks a group G_0 of order q. The KGC calculates sP, for some publicly known group generator point P. The KGC makes public the points P and sP, and the group G_0.

Extract:

The KGC checks that a user has a claim to a particular online Identifier.If they do, the KGC generates their private key and communicates it privately to them. Let Alice's online identifier be $a \in {}_rZ_q^*$, possibly obtained as the hash of an arbitrary string. Alice's public key is Apub= (a + s)P, which can be computed as aP + sP. The KGC computes Alice's private key as Apri = $(a + s)^{-1}$P.

Key Agreement:

Assume that Alice and Bob have private keys issued by the same KGC, respectively Apri and Bpri. Alice and Bob each generate one unique random nonce x, y $\in {}_rZ_q^*$, respectively.Alice compute A_{KA} = x(bP + sP),and send A_{KA} to Bob,and Bob compute B_{KA} = y(aP + sP) and send it to Alice.After received A_{KA} and B_{KA} respectively,Alice compute $key_A = \hat{}t(B_{KA},Apri)^x = \hat{}t(P,P)^{xy}$
And Bob compute $key_B = \hat{}t(A_{KA},Bpri)^y = \hat{}t(P,P)^{xy}$.so Alice and Bob share the same value key = $key_A = key_B$ as their session key.

$$\begin{tikzcd}
\text{Alice} & & \text{Bob} \\
A_{KA} = x(bP + sP) \arrow[r] & {} & \\
& {} & B_{KA} = y(aP + sP) \arrow[l] \\
keya = \hat{}t(B_{KA},Apri)^x & & keyb = \hat{}t(A_{KA},Bpri)^y
\end{tikzcd}$$

This scheme is consistent because:

$$Key_A = \hat{}t(B_{KA},Apri)^x = \hat{}t(P, P)^{xy} = \hat{}t(A_{KA},Bpri)^{xy} = key_B.$$

The escrow property derives from the KGC's ability to recover the shared session key by computing:

$$xP = (s + b)^{-1}A_{KA},\ yP = (s + a)^{-1}B_{KA},$$
$$key = \hat{}t(xP, yP).$$

The scheme is role symmetric, with each party performing the same operations and thus incurring the same computational cost.

2.2 An Authenticated Key Agreement Without Escrow

The key agreement without escrow differs only slightly from the algorithm given above. Again there are three algorithms, Setup, Extract and Key Agreement.This key agreement protocol uses the conventional Tate pairing, not themodified Tate pairing as the escrowed scheme.

Setup:

The KGC randomly generates a random master secret s, and picks two groups, G_0 a group of order r in the base field and G1, the trace zero group in the largest extension field, again of order q. The KGC calculates sP, for some publicly known group generator point P of the group G_0. The KGC makes public the points P and sP, and the two groups G_0 and G1.

Extract:

The KGC checks that a user has a claim to a particular online identifier.If they do, the KGC generates their private key and communicates it privately to them. Let Alice's online identifier be a $\in_r Z_q^*$, possibly obtained as the hash of an arbitrary string. Alice's public key is (a + s)P, which can be computed as aP + sP. Alice's private key is generated as $(a + s)^{-1}Q$,where Q is a generator point in the group G1. It is important that the discrete logarithm between _(P) and Q is unknown3. This can be achieved by obtaining P and Q as the output of random oracles H1 : $\{0, 1\}^* \rightarrow$ G1 and H2 : $\{0, 1\}^* \rightarrow$ G2 evaluated on publicly known constant strings.

Key Agreement:

Assume that Alice and Bob have private keys issued by the same KGC, respectively Apri and Bpri. Alice and Bob each generate one unique random nonce x, y $\in_r Z_q^*$, respectively.

$$\begin{array}{lcl} \text{Alice} & & \text{Bob} \\ A_{KA} = x(bP + sP) & \longrightarrow & \\ & \longleftarrow & B_{KA} = y(aP + sP) \\ keya = \hat{}t(B_{KA},Apri)^x & & keyb = \hat{}t(A_{KA},Bpri)^y \end{array}$$

This scheme is consistent because:

$$Key_A = \hat{}t(B_{KA},Apri)^x = \hat{}t(P, Q)^{xy} = \hat{}t(A_{KA},Bpri)^{xy} = key_B.$$

We also note that, although the KGC has the ability to generate the private keys of both users in the protocol, it is not able to obtain the shared session key for any particular run of the protocol. The KGC can, in this instance, easily compute $t(P,Q)^x$ and $t(P,Q)^y$, but calculating the key from these values involves solving the Computational Diffie - Hellman Problem (CDHP) over the group G2.

2.3 Key Agreement Between Members of Distinct Domains

For key agreement to be possible between members of different groups all that is needed is for the points P, Q (or just P in the case of the escrow enabled system) and the curve description to be the same (standardised). Once these group generator points and curves have been agreed upon, each KGC can generate their own random master secret.Alice's private key is generated by KGC1 with a master secret s1. Bob's private key is generated by KGC2 with a master secret s2. Alice's public key is (a+s1)P and her private key is Apri = $(a+s1)^{-1}P$. Likewise, Bob's public key is (b + s2)P and his private key is Bpri = $(b + s2)^{-1}P$. Notice that now Alice must obtain the point s2P as issued by Bob's KGC and vice-versa.Alice and Bob now perform the authenticated key agreement:

$$\begin{array}{lcl} \text{Alice} & & \text{Bob} \\ A_{KA} = x(bP + s2P) & \longrightarrow & \\ & \longleftarrow & B_{KA} = y(aP + s2P) \\ \text{keya} = \hat{}t(B_{KA},\text{Apri})^x & & \text{keyb} = \hat{}t(A_{KA},\text{Bpri})^y \end{array}$$

This scheme is consistent because:

$$\text{Key}_A = \hat{}t(B_{KA},\text{Apri})^x = \hat{}t(P, P)^{xy} = \hat{}t(A_{KA},\text{Bpri})^{xy} = \text{key}_B.$$

3.Key-Compromise-Impersonation attack on the protocol

In this subsection, we give the following scenario to show that Noel McCullagh and Paulo S. L. M. Barreto's protocol is insecure against the key compromise impersonation attack. We now take a look how the attack apply to the protocol with escrow.That is an adversary Adv who knows A's secret key can impersonate B to A.

$$\begin{array}{lcl} \text{Alice} & & \text{Adv(Bob)} \\ A_{KA} = x(bP + sP) & \longrightarrow & \\ & \longleftarrow & B_{KA}' = y(aP + sP) \end{array}$$

Alice and Adv then copmute the shared session key separately as below:

$$\text{keya} = \hat{}e(B_{KA}', \text{Apri})^x = \hat{}e(y(b+s)P, x(a+s)^{-1}P)$$
$$\text{keyb}' = \hat{}e(A_{KA}, \text{Apri})^y = \hat{}e(y(b+s)P, x(a+s)^{-1}P)$$

so keya = keyb',thus Adv can share a session key with Alice.

First, the adversary Adv selects a random number y and computes B_{KA}'=y(b+s)P,Next, Adv sends B_{KA}' to A. Adv → A : B_{KA}'.A selects a random number a and compute A_{KA}= x(b+s)P,and send it to Adv who A thinks as B.Upon the received messages, A computes the session key: keya = $\hat{}e(B_{KA}', \text{Apri})^x = \hat{}e(y(b+s)P, x(a+s)^{-1}P)$.Having knowning Alice's long-term private key Adv can compute the session key: keyb'= $\hat{}e(A_{KA}, \text{Apri})^y = \hat{}e(x(b+s)P, (a+s)^{-1}P)^y = \hat{}e(y(b+s)P, x(a+s)^{-1}P)$= keya.

Namely, with the knowledge of A's secret key a, Adv can impersonate B to A.From Above we can know that the attack can apply not only the protocol with escrow but also the protocol without escrow and the protocol with different KGC.

Although Noel McCullagh and Paulo S. L. M. Barreto proposed a fix protocol to avoid the KCI affect,but the fix protocol can not satisfy the property of Perfect-Forword-Secrecy.The fix one is as below:

$$\begin{array}{lcl} \text{Alice} & & \text{Bob} \\ A_{KA} = x(bP + sP) & \longrightarrow & \\ & \longleftarrow & B_{KA} = y(aP + sP) \end{array}$$

Alice and Bob then copmute the shared session key separately as below:

$$keya = \hat{}e(B_{KA}, Apri)\ \hat{}e(P, P)^x = \hat{}e(P, P)^{x+y}$$
$$keyb = \hat{}e(A_{KA}, Apri)\hat{}e(P, P)^y = \hat{}e(P, P)^{x+y}$$

The protocol has the property of Forword-Secrecy,that is ,an adversary who has only one user's long-term private kay can not compute the session key.But the adversay can compute all the session keys holding previously if he has known both user's long-term private key as below:

$$\hat{}e(A_{KA}, Bpri) = \hat{}e(P, P)^x,\ \hat{}e(B_{KA}, Apri) = \hat{}e(P, P)^y,$$
$$\hat{}e(P, P)^x \hat{}e(P, P)^y = \hat{}e(P, P)^{x+y}$$

4 modified protocol

4.1 modified protocol with signature

To prevent this attack we modify the protocol with a signature from the idea of K.C. Reddy[1] and Divya Nalla[2]'s scheme[3] .

The scheme is as foloows:

Alice Bob

$A_{KA} = x(bP + sP)$

S_A =hash(A_{KA}) Apri + xApri

$(A_{KA,}\ S_A)$ $\longrightarrow$

$B_{KA} = y(aP + sP)$

S_B =hash(B_{KA}) Bpri + xBpri

$\longleftarrow$ $(B_{KA,}\ S_B)$

Having received $(B_{KA,}\ S_B)$,A verifies $\hat{}e(S_B, Bpub) = \hat{}e\ (hash(B_{KA})P,P)\ \hat{}e\ (B_{KA}\ , Apri\ )$. If the equation holds, then A computes

$$keya = \hat{}t(B_{KA},Apri)^x\ .$$

B verifies $\hat{}e(S_A, Apub) = \hat{}e\ (hash(A_{KA})P,P)\ \hat{}e\ (A_{KA}\ , Bpri\ )$. If the equation holds, then B computes

$$keyb = \hat{}t(A_{KA},Bpri)^y$$

where Apub=(a+s)P from aP+sP and Bpub=(b+s)P from bP+sP .

$$\begin{aligned}\hat{}e(S_A, Apub) &= \hat{}e(hash(A_{KA})(a+s)^{-1}P+x(a+s)^{-1}P,(a+s)P)\\ &= \hat{}e(hash(A_{KA})P,P)\hat{}e\ (x(a+s)^{-1}P,(a+s)P\ )\\ &= \hat{}e(hash(A_{KA})P,P)\hat{}e(xP,P)\\ &= \hat{}e\ (hash(A_{KA})P,P)\hat{}e(x(b+s)\ P,(b+s)^{-1}P)\\ &= \hat{}e(hash(A_{KA})P,P)\ \hat{}e\ (A_{KA}\ , Bpri\ ),\end{aligned}$$

anyone else not knowning both of the knowledge of Apri and the ephemeral key x can not make the signature S_A from A_{KA}.If an adversary D have got Alice's private key Bpri,he can randomly choose y to compute y(b+s)P in orderto impersonate Bob to share a secret key with Alice.If he uses y to compute S_B,the equation $\hat{}e(S_B, Bpub) = \hat{}e(hash(B_{KA})P,P)\ \hat{}e\ (B_{KA}\ , Apri\ )$ will not holds,because

$$\begin{aligned}\hat{}e(S_B, Bpub) &= \hat{}e(hash(B_{KA})(b+s)^{-1}P+y(b+s)^{-1}P,(b+s)P)\\ &= \hat{}e(hash(B_{KA})P,P)\hat{}e\ (y(b+s)^{-1}P,(b+s)P\ )\\ &= \hat{}e\ (hash(B_{KA})P,P)\hat{}e(y(b+s)\ P,(b+s)^{-1}P)\\ &= \hat{}e(hash(B_{KA})P,P)\ \hat{}e\ (B_{KA}\ , Bpri\ ),\\ &\neq \hat{}e(hash(B_{KA})P,P)\ \hat{}e\ (B_{KA}\ , Apri\ )\end{aligned}$$

D can both satisfy the verify equation and impersonate Bob to Alice successly unless he knows the relationship between Apri and Bpri,but this is a CDH problem.

4.2 modified protocol without signature

Assume that Alice and Bob have private keys issued by the same KGC, respectively Apri and Bpri. Alice and Bob each generate one unique random nonce x, $y \in_R Zq$, respectively.

$$\begin{array}{lcl}
\text{Alice} & & \text{Bob} \\
A_{KA} = x(bP + sP) & \longrightarrow & \\
 & \longleftarrow & B_{KA} = y(aP + sP) \\
key_A = ê(B_{KA},Apri)^{(x+1)}\ ê(P,P)^x & & key_B = ê(A_{KA},Bpri)^{(y+1)}\ ê(P,P)^y
\end{array}$$

This scheme is consistent because:

$$\begin{aligned}
key_A &= ê(B_{KA},Apri)^{(x+1)}\ ê(P,P)^x = ê(P,P)^{y(a+s)(a+s)^{-1}(x+1)+x} \\
&= ê(P,P)^{xy+x+y} = ê(P,P)^{x(y+1)+y} \\
&= ê(P,P)^{x(b+s)(b+s)^{-1}(y+1)+y} \\
&= ê(A_{KA},Bpri)^{(y+1)}\ ê(P,P)^y \\
&= key_B
\end{aligned}$$

The escrow property derives from the KGC's ability to recover the shared session key by computing:

$$xP = (s + b)^{-1}A_{KA},$$
$$yP = (s + a)^{-1}B_{KA},$$
$$key = ê(xP, yP)\ ê(xP, P)\ ê(P, yP).$$

Our scheme is role symmetric, with each party performing the same operations and thus incurring the same computational cost.

4.3 modified protocol with confirmation

Assume H1 is a random oracle ,We can use the key agreement with confirmation in escrow mode as below:

$$\begin{array}{lcl}
\textbf{Alice} & & \textbf{Bob} \\
A_{KA} = x(bP + sP) & \longrightarrow & \\
 & \longleftarrow & B_{KA}=y(aP+sP)||H(A_{KA}||B_{KA}||ê(P,P)^x) \\
H(B_{KA}||ê(P,P)^y) & \longrightarrow & \\
key_A = ê(B_{KA},Apri)^{(x+1)}\ ê(P,P)^x & & key_B = ê(A_{KA},Bpri)^{(y+1)}\ ê(P,P)^y
\end{array}$$

This scheme is consistent because:

$$\begin{aligned}
key_A &= ê(B_{KA},Apri)^{(x+1)}\ ê(P,P)^x = ê(P,P)^{y(a+s)(a+s)^{-1}(x+1)+x} \\
&= ê(P,P)^{xy+x+y} = ê(P,P)^{x(y+1)+y} \\
&= ê(P,P)^{x(b+s)(b+s)^{-1}(y+1)+y} \\
&= ê(A_{KA},Bpri)^{(y+1)}\ ê(P,P)^y \\
&= key_B
\end{aligned}$$

or in escrowless mode as below:

$$\begin{array}{lcl}
\textbf{Alice} & & \textbf{Bob} \\
A_{KA} = x(bP + sP) & \longrightarrow & \\
 & \longleftarrow & B_{KA}=y(aP+sP)||H(A_{KA}||B_{KA}||ê(P,Q)^x)
\end{array}$$

$$H(B_{KA}||\hat{e}(P,Q)^{y}) \longrightarrow$$

$$key_A = \hat{e}(B_{KA}, Apri)^{(x+1)}\ \hat{e}(P,Q)^{x} \qquad key_B = \hat{e}(A_{KA}, Bpri)^{(y+1)}\ \hat{e}(P,Q)^{y}$$

This scheme is consistent because:

$$\begin{aligned} key_A &= \hat{e}(B_{KA}, Apri)^{(x+1)}\ \hat{e}(P,Q)^{x} = \hat{e}(P,Q)^{y(a+s)(a+s)^{-1}(x+1)+x} \\ &= \hat{e}(P,Q)^{xy+x+y} = \hat{e}(P,Q)^{x(y+1)+y} \\ &= \hat{e}(P,Q)^{x(b+s)(b+s)^{-1}(y+1)+y} \\ &= \hat{e}(A_{KA}, Bpri)^{(y+1)}\ \hat{e}(P,Q)^{y} \\ &= key_B \end{aligned}$$

Any one not knowing the corresponding private key can not be able to calculate the value$\hat{e}(P,Q)^{x}$ or $\hat{e}(P,Q)^{y}$ in escrow mode or $\hat{e}(P,Q)^{x}$ or $\hat{e}(P,Q)^{y}$ in escowless mode,unless he can solve the DLP.So having got the value a user can confirm that the participant has get the correct message indeed. So this is a AKC protocol.

The AK protocol need two pairings,two expotions and one scale multiply,and the AKC protocol need two pairings ,two expentation and one scale multiply also,and can precompute one pairing.

4. Conclusion

Noel McCullagh and Paulo S. L. M. Barreto's protocol is insecure against the key compromise impersonation attack,and the fix protocol has not the property of Perfect-Forword-Secrecy.We propose some schemes to modify the protocol,and the modified protocol is secure against all attack and satisfy the property of Known-Key Security**,** Perfect-Forward-Secrecy, Key-Compromise Impersonation, Unknown Key-Share,and Key control and so on.

References

[1]. Noel McCullagh[1] and Paulo S. L. M. Barreto[2]. A New Two-Party Identity-Based Authenticated Key Agreement. Cryptology ePrint Archive, Report 2004/122, 2004. http://eprint.iacr.org/2004/122/.

[2] S. B.Wilson, and A. Menezes, "Authenticated Diffie-Hellman key agreement protocols",Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC '98), Lecture Notes in Computer Science, pp. 339-361, 1999.

[3] K.C. Reddy[1] and Divya Nalla[2]. Identity Based Authenticate Group Key Agreement Protocol. Lecture Notes in Computer Science Volume 2551/2002 , Springer-Verlag Heidelberg