An IBE Scheme to Exchange Authenticated Secret Keys

Waldyr Dias Benits Júnior[1] , **Routo Terada (Advisor)**[1]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo
R. do Matão, 1010 – Cidade Universitária – 05508-900 São Paulo, SP

waldyrbenits@bol.com.br, rt@ime.usp.br

Abstract. We present a variant of the Boneh & Franklin Identiy-based Encryption IBE scheme to derive an authenticated symmetric key-exchange protocol, when combined with a signature scheme. Our protocol uses IBE as a secure channel to establish a symmetric key between two users and, after that, further communication can be done by symmetric cryptography, much faster than pairing-based cryptography.
Key words: Identity-based cryptosystem, pairing, bilinear map, elliptic curve cryptography, key-exchange protocol.

1. Introduction

The concept of identity-based cryptography was first proposed in 1984 by Adi Shamir [9]. In his paper, Shamir presented a new model of asymmetric cryptography in which the public key of any user is a characteristic that uniquely identifies himself/herself, like an e-mail address. Since the public keys are not random numbers, digital certificates are needless. After Shamir's model, many researchers tried to propose a cryptographic scheme in this model, but only in 2001 an efficient one was proposed by Dan Boneh and Matthew Franklin [3], based on pairings.

However, the IBE scheme of Boneh & Franklin (which we call "BF's scheme" in the remainder of this paper) depends on a random number chosen by the sender of the message, and if this random number is not carefully chosen, the security of the scheme is seriously compromised. This occurs because the BF's scheme encrypts a message using an XOR function, and as proved in [4], XOR-based functions can be easily broken by a chosen-plaintext attack if the same key is used more than once.

In this paper, we propose a variant of the BF's scheme, replacing the XOR function by a symmetric encryption algorithm. With this change, BF's scheme can be used as a key-exchange protocol provided that it is combined with a signature scheme to guarantee mutual authentication of the parties involved.

There are similar key-exchange protocols based on pairings, among them we can cite [5, 6, 8, 10, 11]. But they are less efficient.

This paper is organized as follows: Section 2 summarizes ID-based cryptosystems. Section 3 reviews the IBE BF's scheme and shows the need of a carefully chosen random number. Section 4 presents our variant of the BF's scheme, for the key-exchange purpose. Section 5 reviews a signature scheme to be used together with a modified Boneh & Franklin IBE, so as to guarantee authentication of the two parties in the protocol (a.k.a. mutual authentication). Finally, section 6 concludes the paper and proposes further research.

2. Identity-based cryptosystems

2.1. Bilinear maps

Let $\mathbb{G}_1$ be an additive group of prime order q and $\mathbb{G}_2$ be a multiplicative group of the same order, in which the discrete logarithm problem (DLP) is assumed to be hard. Concretely, we can think of $\mathbb{G}_1$ as a group of points on an elliptic curve over $\mathbb{F}_q$, and $\mathbb{G}_2$ as a subgroup of the multiplicative group of a finite field $\mathbb{F}^*_{q^k}$ for some $k \in \mathbb{Z}^*$. Let P be a generator of $\mathbb{G}_1$ and let $e : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ be a mapping with the following properties:

1. bilinearity: A mapping e is bilinear if $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and for all $a, b \in \mathbb{F}_q$, where $\mathbb{F}_q$ is a finite field of order q;
2. non-degeneracy: A mapping is non-degenerate if exists $Q \in \mathbb{G}_1$ so that $e(P, Q) \neq 1$, that is, the mapping does not map all pairs in $\mathbb{G}_1 \times \mathbb{G}_1$ to the identity in $\mathbb{G}_2$;
3. Computability: A mapping is efficiently computable if $e(P, Q)$ can be computed in polynomial-time for all $P, Q \in \mathbb{G}_1$.

Examples of pairings that satisfy these properties are the Weil pairing and the Tate pairing. Due to many improvements on its computation [1, 2], we consider the Tate pairing more efficient than Weil pairing.

2.2. Key generation

In ID-based cryptosystems we need a private key generator (PKG) which generates a pair of keys based on the identity of the user. After generating the keys, the PKG uses a secure channel to send the private key to the owner of the identity.

The process of generating keys is the following: let ID be an identity of a user Alice (e.g., her e-mail address). First, PKG computes Alice's public key (Q_{alice}), by mapping her identity to a point of the elliptic curve (using a hash function H_1) and then, PKG uses his master key $s \in \mathbb{F}_q$ to compute Alice's private key (S_{alice}), multiplying Q_{alice} by s. This process is summarized below:

$$Q_{alice} = H_1(ID) \quad (1)$$

$$S_{alice} = sQ_{alice} \quad (2)$$

Here, to strengthen the security model, we can view the hash function H_1 as a random oracle, defined as follows:

$$H_1 : \{0, 1\}^* \longrightarrow \mathbb{G}_1$$

3. The IBE BF's scheme

Let (Q_{bob}, S_{bob}) be a pair of Bob's identity-based keys; R_{PKG} be a standard public key of the entity that generates Bob's keys, so that $R_{PKG} = sP$, where s is PKG's master key and P a generator of $\mathbb{G}_1$. Let m be a message that Alice intends to secretly send to Bob.

The original IBE scheme proposed by Dan Boneh and Matthew Franklin [3] is as follows:

3.1. Encryption

Alice chooses a random number $r \in \mathbb{F}_q$ and computes[1]:

$$\begin{cases} U & = rP \\ V & = m \oplus H_3(e(R_{PKG}, rQ_{bob})) \end{cases} \tag{3}$$

and sends the ciphertext (U, V) to Bob.
To increase security, we can view the hash function H_3 as a random oracle, defined as follows:

$$H_3 : \mathbb{G}_2 \longrightarrow \{0, 1\}^*.$$

Notice that Alice uses Bob's ID-based public key to encrypt m. To do that, she needs only to know his identity (e.g. Bob's e-mail address).

3.2. Decryption

Bob, after receiving (U, V) from Alice, performs the following computation to recover cleartext m:

$$m = V \oplus H_3(e(U, S_{bob})) \tag{4}$$

Clearly, we can see that Bob uses his ID-based private key to find m.

3.3. Verification

Let us show how Bob is able to recover m:

$$\begin{aligned} V \oplus H_3(e(U, S_{bob})) &= V \oplus H_3(e(rP, S_{bob})), && \text{as } U = rP \text{ by}(3) \\ &= V \oplus H_3(e(rP, sQ_{bob})), && \text{as } S_{bob} = sQ_{bob} \text{ by}(2) \\ &= V \oplus H_3(e(P, Q_{bob}))^{rs}, && \text{by bilinearity} \\ &= V \oplus H_3(e(sP, rQ_{bob})), && \text{by bilinearity} \\ &= V \oplus H_3(e(R_{PKG}, rQ_{bob})), && \text{as } R_{PKG} = sP \\ &= m, && \text{due to equation } (3) \\ & && V = m \oplus H_3(e(R_{PKG}, rQ_{bob})) \end{aligned}$$

3.4. Security of BF's scheme

Schemes based on pairings, like BF's scheme, depend not only on the hardness of DLP, but also on the hardness of a problem known as bilinear Diffie-Hellman problem (BDHP). First, we define BDHP as follows:

[1] $\oplus$ represents exclusive-OR (XOR).

- BDHP – Given $(P, aP, bP, cP) \subset \mathbb{G}_1$ compute $e(P, P).^{abc}$

We are going to show that, even if the DLP was hard, an adversary could obtain advantages if the BDHP was easy.

Suppose that Alice sent a message m to Bob, using BF's scheme. As we saw, Bob received (U, V) from Alice. If an adversary Charles intends to decrypt m, he must compute m using equation (4). However, as S_{bob} is unknown to Charles and assuming that DLP is hard, Charles fails to compute $e(S_{bob}, U)$ and cannot recover m.

Now, let us assume that Charles is able to easily solve the BDHP in the group chosen by Alice. We can see that Charles knows the following values:

- $R_{PKG} = sP$;
- $Q_{bob} = H_1(ID_{bob}) = hP$ (for some $h \in \mathbb{F}_q$, because Q_{bob} is a point in the elliptic curve);
- $U = rP$, because we assumed that Charles had intercepted (U, V).

As DLP is hard, the values s, h and r are unknown to Charles, but assuming that he can solve the BDHP, if he knows sP, hP and rP he can compute $e(P, P)^{shr}$.
Nevertheless:

$$\begin{aligned} e(S_{bob}, U) &= e(sQ_{bob}, rP) \\ &= e(shP, rP) \\ &= e(P, P)^{shr} \end{aligned}$$

Thus, assuming that the BDHP is easy, an adversary can compute correctly the pairing $e(S_{bob}, U)$ and recover m, even if the DLP is hard, that is, an adversary needs neither S_{bob} nor s to decrypt m.

We can guarantee the security in pairing-based cryptosystems by choosing appropriately the parameters k and q so that both the DLP and the BDHP are hard.

3.5. On the selection of a random number *r*

We saw in equation (3) that Alice must choose a random number r. In the original paper, the authors do not mention the importance of this choice. To increase the security of the scheme, Alice must choose a uniformly distributed and independent number, so as not to give any chance for an adversary to obtain advantage. For example, let us assume that Alice chooses r from a regular sequence $r,\ r + d,\ r + 2d, ...$ (for some integer d). The adversary only needs to find one r and so, the others are easily predicted, and the security of the system is compromised.

An extreme situation occurs when Alice chooses the same r for two different messages. If she does that, any adversary can perform an attack similar to the one described in [4] — and summarized below — and recovers m.

Let us suppose that Alice sent messages m_1 and m_2 to Bob and she carelessly chose the same r. In the equation (3), we see that the value U will be the same, since it depends on r, but the value V will be different, as it depends on m. So, let V_1 and V_2 be the values computed by Alice when she sent m_1 and m_2, respectively.

Besides, as the values of $H_3(e(R_{PKG}, rQ_{bob}))$ for both messages (m_1 and m_2) remain constant since they do not depend on m, we can call them x.

Thus, let us rewrite the second equation of (3) as follows:

$$\begin{cases} V_1 &= m_1 \oplus x \\ V_2 &= m_2 \oplus x \end{cases}$$

If an adversary Charles intercepts values V_1 and V_2, he can compute:

$$\begin{aligned} V &= V_1 \oplus V_2 \\ &= (m_1 \oplus x) \oplus (m_2 \oplus x) \\ &= m_1 \oplus m_2 \end{aligned}$$

By knowing the value V, if an adversary obtains m_1 (without loss of generality), he can perform a known plaintext attack and gets m_2 successfully.

This weakness is because m is XOR-ed with a computed value that depends on a random number. To avoid this problem we propose next to replace the XOR function by a nonlinear function.

4. A key-exchange protocol

We propose a variant of the BF's scheme, so that it can be used as a key-exchange protocol. It is well known the existing symmetric cryptography is much faster than asymmetric one and, in practice, asymmetric cryptography is often used as a key exchange protocol to exchange a secret key which is then used to ensure secrecy between the parties.

We can do the same with BF's scheme by replacing the XOR function by a known symmetric encryption algorithm (e,g., 3-DES or AES). With this substitution, if Alice wants to communicate with Bob, she uses IBE only to exchange a secret key with Bob and then they can exchange secure messages using symmetric cryptography, which is known to be much faster than pairing-based cryptosystems.

The proposed variant is as follows: notice, by equations (3) and (4) that Alice and Bob compute the same value (Alice computes $H_3(e(R_{PKG}, rQ_{bob}))$ and we have proved that this value is equal to the one computed by Bob, $H_3(e(U, S_{bob}))$. Let us call this value k.

In our protocol, the value k, instead of being XOR-ed with a message by Alice to compute the encryption of m, (that is, V), it will be used as a symmetric encryption key between Alice and Bob. In this case, Alice does not have to carefully choose a uniformly distributed and independent random number for each message she intends to transmit as occurs in the original scheme. She only needs to choose one random number to exchange a key with Bob and after that, exchange as many encrypted messages as needed. If she later wants to change the secret key, she chooses another random number and establishes another key.

Moreover, as the key value depends on the hash function H_3, we can impose the hash value to be as large as we want. For example, we can define the hash function to map the pairing value to a 128-bit key, or 256-bit key or even a larger key.

The first step of our protocol is presented below:

4.1. Key establishment phase I

As in the encryption phase of BF's scheme, Alice chooses a random number $r \in \mathbb{F}_q$ and computes:

$$\begin{cases} U & = rP \\ k & = H_3(e_t(rQ_{bob}, R_{PKG})) \end{cases} \tag{5}$$

and sends (U) to Bob.

4.2. Key establishment phase II

As in the decryption phase of BF's scheme, Bob will, after receiving (U) from Alice, perform the following computation to recover k:

$$k \;\; = \;\; H_3(e_t(S_{bob}, U)) \tag{6}$$

With this step, Alice is sure that only Bob can recover k, since only he knows the appropriate S_{bob}. However, for this protocol to be considered secure, Bob needs to be sure that the message was sent by Alice. Thus, to complete our protocol, it is necessary that Alice signs some information she sends to Bob, so as to prove that she is, in fact, Alice.

In the next section, we will see the complete key exchange protocol, combining BF's scheme with a signature scheme, to obtain mutual authentication.

5. Using a signature scheme together with BF's scheme

We saw in our key exchange protocol that Alice must sign some information she sends to Bob, in order to prove that she is Alice. In this section we will see an example of a signature scheme proposed by Florian Hess [7], that can be used together with BF's scheme.

5.1. Signature

If Alice wants to sign a message m, first she chooses a random number $t \in \mathbb{F}_q$ and a random point $P_1 \in \mathbb{G}_1^*$ and computes:

$$r = e(P_1, P)^t \tag{7}$$

Afterwards, she computes:

$$h = H_2(m||r) \tag{8}$$

and at last,

$$W = hS_{alice} + tP_1 \tag{9}$$

Now Alice's signature on m is (W, h). We can think of H_2 as a random oracle defined as follows:

$$H_2 : \{0,1\}^* \longrightarrow \mathbb{F}_q$$

Notice by equation (9) that Alice uses her ID-based private key S_{alice} to compute the signature on m. See also that the sum in equation (9) represents a sum of points on an elliptic curve, since hS_{alice} and tP_1 are both points in $\mathbb{G}_1$.

5.2. Verification

If Bob wants to verify that the signature comes from Alice, he must compute:

$$r = e(W, P) \cdot e(Q_{alice}, -R_{PKG})^h \tag{10}$$

After computing r, Bob accepts Alice's signature as valid if, and only if:

$$h = H_2(m||r) \tag{11}$$

Notice that only public parameters are used to verify the signature, meaning that anyone is able to perform such verification.

5.3. Proof

Now, we are going to prove that the equation (10) holds for a valid signature.

$$\begin{aligned}
e(W, P) \cdot e(Q_{alice}, -R_{PKG})^h &= e(hS_{alice} + tP_1, P) \cdot e(Q_{alice}, -R_{PK})^h \\
&= e(hS_{alice} + tP_1, P) \cdot e(Q_{alice}, -sP)^h \\
&= e(hS_{alice} + tP_1, P) \cdot e(Q_{alice}, -P)^{sh} \\
&= e(hS_{alice} + tP_1, P) \cdot e(sQ_{alice}, P)^{-h} \\
&= e(hS_{alice} + tP_1, P) \cdot e(S_{alice}, P)^{-h} \\
&= e(hS_{alice} + tP_1, P) \cdot e(-hS_{alice}, P) \\
&= e(hS_{alice} - hS_{alice} + tP_1, P) \\
&= e(tP_1, P) \\
&= e(P_1, P)^t \\
&= r \qquad \text{(due to equation (7))}
\end{aligned}$$

In Hess' scheme, we can see by equation (11) that a verifier needs to know the message m so as to perform the verification. There are other kinds of signature schemes, which include message recovery, in that the verifier by using sender's public key, is able to recover the message. Examples of the latter are RSA signatures.

If we use Hess' scheme to authenticate the sender Alice, our complete key exchange protocol will be as follows: after computing k by equation (5), Alice signs k with her private key S_{alice}, using k in Hess' scheme (equation 8) instead of m. After that, Alice sends U and (W, h) to Bob, where (W, h) is Alice's signature on k.

Bob, to receive the secret key k, first uses his private key S_{bob} to compute k (this way, Alice is sure that only the authentic Bob could decrypt correctly) and then he checks if (W, h) is a valid signature of Alice on k. If this verification is correct, they can start exchanging messages using k as a secret key; otherwise, he rejects k.

Alice can choose any signature scheme to combine with BF's scheme. If she decides to use a message-recovery signature scheme instead of Hess' scheme, she has to sign the value U, instead of k, otherwise anyone could recover the secret key k by only using Alice's public key. Bob, after receiving the signed value U, performs a signature verification to check if U comes from Alice. If not, he rejects U.

In our key exchange protocol, once Alice and Bob are authenticated and a key is exchanged, any message between them can be encrypted by an agreed upon symmetric algorithm (for example, Alice sends $\mathcal{C} = C_k(m)$ and Bob computes $m = D_k(\mathcal{C})$, where D_k is the inverse of C_k, implemented by choosing secure algorithms such as AES, 3-DES *etc*.

6. Conclusions and further research

We have shown how to use IBE BF's scheme as a key exchange protocol by replacing the XOR operation by a symmetric algorithm and using it together with a signature scheme, so as to establish mutual authentication. This way, we avoid the dependency on a random number choice that can be exploited in the original protocol, if the sender does not carefully choose this random number. Moreover, our protocol allows the pairing-based scheme to be used only as a secure channel to transmit secret keys, and, after that, messages can be exchanged by symmetric cryptography, much faster than asymmetric cryptography, especially pairing-based cryptography.

We suggest, as further research, a comparison between our proposed key-exchange protocol and other existing key-exchange protocols [5, 6, 8, 10, 11] in terms of security and performance. Another interesting line of research would be to conduct attacks against our protocol. We think it is strong against the known attacks but we advice further research to be done.

References

[1] BARRETO, P., Kim, H., Lynn, B. and Scott, M., *Efficient Algorithms for Pairing-Based Cryptosystems*. Advances in Cryptology - Crypto'2002, Lecture Notes in Computer Science 2442, Springer-Verlag (2002), pp. 354–368.

[2] BARRETO, P., Lynn, B. and Scott, M., *On the selection of Pairing-Friendly Groups*. Available at http://eprint.iacr.org/2003/086.

[3] BONEH, D. and Franklin, M., *Identity Based Encryption from the Weil Pairing*. Advances in Cryptology - CRYPTO'2001, Springer-Verlag, LNCS 2139, pp. 213-229, 2001.

[4] BORISOV, N., Golberg, I. and Wagner D., *Intercepting Mobile Communications: The Insecurity of 802.11*. Available at http://www.isaac.cs.berkeley.edu/isaac/mobicom.pdf.

[5] CHEN, L. and Kudla, C., *Identity Based Authenticated Key Agreement from Pairings*. IACR eprint, report 2002/184.

[6] GUNTHER, C.G., *An Identity-Based Key-Exchange Protocol*. In Proceedings of Eurocrypt 1989, Lecture Notes in Computer Science, Springer-Verlag, pp 29-37, 1989.

[7] HESS, F., *Efficient Identity Based Signature Schemes Based on Pairings*. In: Lecture Notes in Computer Science. Springer-Verlag, 2002. v. 2595, p. 310–324.

[8] SCOTT, M., *Authenticated ID-based Key Exchange and Remote Log-in with Insecure Token and PIN Number*.IACR eprint, report 2002/164.

[9] SHAMIR, A., *Identity Based Cryptosystems and Signature Schemes*. Advances in Cryptology - CRYPTO'84, Springer-Verlag, LNCS 196, pp. 47-53, 1985.

[10] SMART, N. P., *An Identity Based Authenticated Key Agreement Protocol Based on the Weil Pairing*. Electronics Letters, Vol 38, pp 630-632, 2002.

[11] ZHANG, F., Liu, S. and Kim, K. *ID-Based One Round Authenticated Tripartite Key Agreement Protocol with Pairings*. IACR eprint, report 2002/122.