In the point of view security
An efficient scheme with random oracle

Rkia Aouinatou[1], Mostafa *Belkasmi*[2]
[1] *UFR SYSCOM*, Faculty of Sciences, Mohamed V-Agdal B.P. 1014 Rabat, Morocco
$\ast$ *Laboratoire de recherche Informatique et Telecommunication: LRIT*
Email: rkiaaouinatou@yahoo.fr
[2] ENSIAS: University Mohammed V- Souissi, Rabat, Morocco
Email: belkasmi@ensias.ma

Abstract

We present in these papers a scheme, which bypasses the weakness presented in the existed scheme of IBE with random oracle. We propose, a secure scheme which project into Z_p contrary to elliptic curve as with Boneh and Franklin. More, our scheme is basing in its study of simulation in the problem 4-EBDHP which is more efficient than q-BDHIP used by Skai Kasarah. We provide the prove of security of our scheme and we show its efficiency by comparison with the scheme declared above. Even if it we have a little cost in complexity, but as in the field cryptography we are more interested to the security, this makes our proposition more efficient.

Keywords :

Random Oracle, IBE, Security, 4-EBDHP, q-BDHIP, projection into elliptic curve.

1 Introduction

This article is devoted to present a 4^{th} scheme of IBE in the random oracle model[1]. And since this latter is a weaker notion [2] it will be interesting to reduce as possible the weaknesses of the schemes of IBE under this model. That's we will do in this article.

Problem of security with the existed scheme

Firstly we recap that we have three scheme of IBE under the notion of the random oracle : Boneh and Franklin[3], Skai Kasarah[4], Boneh Boyen[5](full version).
Projection expensive of Boneh and Franklin : The identity-based cryptography (especially IBE) was firstly introduced by Shamir[6] in 1984, but it is not realistic until the invention of the scheme of Boneh and Franklin[3] in 2001. Even if this latter is drawn in the model of the Random Oracle, it has some weakness. The hot one is that it can be dissociated of the projectin in the elliptic curves. Which limits the use of these latter and this can influence in the security. To overcome this problem, the work [7] [8] are proposed. In [7] Michael Scott suggested using $H_1(ID) = cH_0(ID)$ with H_0 hashed in the random point, c is the cofactor. The [7] may be attractive, but it can be enjoyed, if we use the Tate pairing instead of the Weil pairing originally used by Boneh and

Franklin. In [8] the authors are based on [20] to project the hash in the ordinary curves instead of super singular, their result is important. But, it was based on the fact that they suggest that the proof of security remains valid if we integrate the random oracle, which requires a thorough study. More than that they deal only with characteristic 3 which is a restrict.

Problem minor with Skai Kasarah The second efficient scheme in the randoms oracles is that of Skai Kasarah [4] in 2003. This scheme poject into Z_p contrary to elliptic curve as with [3], but it has another problem. It is the use of a minor problem : q-BDHIP in the study of simulation which has a security $O(\sqrt[3]{q})$(PDL has O(q)) according to the result of cheon [9]. And this pose a problem of security against malicious attack and even against passive attack.

Organization

We firstly give some preliminaries we then present our scheme in section 3. In section 4 we test the security of our scheme. Section 5 is dedicated to test the efficiency of our scheme by comparison with the existed. And in the end we conclude.

2 Some Preliminaries

2.1 Elliptic Curves

In general the equation of an elliptic curve E over a finite field k, is of the form :
$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$ (*)
The elliptic curve over a field k, is defined as follows :
E(k)={ $(X, Y) \in K^2/(X, Y)$ verifies (*)}
A point P of coordinated (x, y) in an ellitpic curve E is singular, if $\frac{\partial(E)}{\partial(x)}$=0 and $\frac{\partial(E)}{\partial(y)} = 0$. The curve is called singular if it has at least one point singular.
The elliptic curve admits an element neutral noted universally by O, which has the form : (0,1,0) in the projective coordinates.

2.1.1 Group law for elliptic curve

An elliptic curve is fitted with an internal law of composition additive :
Let $P = (X_P, Y_P) \in E(k)$ and $Q = (X_Q, Y_Q) \in E(k)$ so :
$P + O = P, O + P = P$
$P + (-P) = O$, $-P = (X_P, -Y_P - a_1X_P - a_3)$

Explicit formula

Let $P = (X_P, Y_P)$ and $Q = (X_Q, Y_Q)$. The coordinates of P + Q are defined as :
$X_{P+Q} = \lambda^2 + a_1\lambda - a_2 - X_P - X_Q$,
$Y_{P+Q} = -(\lambda + a_1)X_{P+Q} - \nu - a_3$
With, $\lambda = \frac{Y_Q - Y_P}{X_Q - X_P}$ if $P \neq Q$ and $\lambda = \frac{3X_P^2 + 2a_2X_P + a_4 - a_1Y_P}{2Y_Q + a_1X_P + a_3}$ if not .
In general, for a field k of characteristic different to 2 and 3, the coordinates will be as follows :
If $X_P \neq X_Q$, P+Q is the point of coordinate (X_{P+Q}, Y_{P+Q}) such that : X_{P+Q}=$\lambda^2 - X_P - X_Q$
And, $Y_{P+Q} = \lambda(X_P - X_{P+Q}) - Y_P$ with $\lambda = \frac{Y_Q - Y_P}{X_Q - X_P}$
But if : $X_P = X_Q$ with $Y_P \neq Y_Q$, we will have P+Q=O. And if : $Y_P = Y_Q$, we will have a point double 2P of coordinated (X_{2P}, Y_{2P}), such that $X_{2P} = \lambda^2 - 2X_P$ and $Y_{2P} = \lambda(X_P - X_{P+Q}) - Y_P$,

with $\lambda = (3X_P + a)(2Y_P)^{-1}$. Taking into account that the equation of the elliptic curve for a field of characteristic different from 2 and 3 is in the form : $Y^2 = X^3 + aX + b$ after using a suitable change of variable.

2.2 Pairing

A pairing is a bilinear map that takes two points on an elliptic curve and gives an element of the group multiplicative of n-th roots of unity.
Considering E(k)[r] (points of r-torsion on elliptic curve E)

2.2.1 Propriety

Bilinear : : $\forall$ P_1,P_2,Q_1, $Q_2 \in E[r]$, $c_r(P_1 + P_2, Q_1) = c_r(P_1, Q_1) \cdot c_r(P_2, Q_1)$ and $c_r(P_1, Q_1 + Q_2) = c_r(P_1, Q_1) \cdot c_r(P_1, Q_2)$
Identity : $\forall P \in E[r]$ $c_r(P, P) = 1$
Alternate : $c_r(P, Q) = c_r(Q, P)^{-1}$
Non-degenerate : If $\forall P \in E[r]$ $c_r(P, Q) = 1$ then $Q = O$ and if $\forall Q \in E[r]$ $c_r(P, Q) = 1$ then $P = O$
It is clear from these properties that we have $c_r \in \mu_r$ (set of the r^{th} roots of unity), since $c_r(P, Q)^r = c_r(rP, Q) = c_r(O, Q) = 1$
Among the pairing we cited : Weil, Tate, Ate, η

2.2.2 Weil Pairing

The Weil pairing is defended as follows : e_r :$E[r] \times E[r] \to \mu_r$ (μ_r is the set of the r^{th} root of the unity) such that : $e_r(P, Q) = \frac{f_{D_Q}(D_P)}{f_{D_P}(D_Q)}$

2.2.3 Tate Pairing

The Tate pairing is the application :
t_r :E(k)[r]×E(k)/rE(k) → $k^*/(k^*)^r$
(P,Q)→$t_r(P, Q)$=$f_{D_P}(D_Q)$ modulo $(k^*)^r$. And to have an exact value, it can be defined as follows : $t_r(P, Q) = (f_{D_P}(D_Q))^{(q^k-1)/r}$

2.3 Random Oracle & Standard Model

Random Oracle : In cryptography, an oracle is a random that answers all queries proposed at random and specific request (for more details we send the interested to[1])
The oracle answers in the same way each time it receives such requests. In other words, a random oracle is a mathematical function used in a mapping, but all its requests have the randomized response within its area of output.
Virtually the Random Oracles are often used to produce hash functions (typically built). These functions use in their regime Random Oracle with the mathematic hypothesis very strong but we can say that there are hash functions which can't operate with the Random Oracle. The opposite of the random oracle is Standard Model.

2.4 IBE security notions

The security of a cryptographic scheme combining the possible goals and attack models. The most important goal are : indistinguishability (IND), Semantic Security. Regarding attacks we have : chosen-plaintext attacks (CPA), chosen-ciphertext attacks (CCA). The relation betwwen all this was given in [21][22]

Definition :IND-ID/sID-{CCA, CPA}

Let Γ = (S,X,E,D) be an IBE scheme, and let A = (A_0, A_1, A_2) be any 3-tuple of PPT oracle algorithms. For ATK = ID/sID-CPA, ID/sID-CCA, we say Γ is IND/sID-ATK secure if for any 3-tuple of PPT oracle algorithms A,| $\wp$r(1)-$\wp$r(2) | $\in neg$, where

$$\wp\text{r(i)}=\left\{ v=0 \left| \begin{array}{l} (\text{id},\gamma) \longleftarrow A_0(1^l)\\ (\text{pms,mk}) \longleftarrow S(1^l)\,;\\ ((m^{(1)},m^{(2)},id_{ch}),\sigma) \longleftarrow A_1^{O_1,O_2}(pms,id,\gamma)\\ c \longleftarrow E(pms,id_{ch},m^{(i)});\\ v \longleftarrow A_2^{O_1,O_2}(\sigma,(id_{ch},c)) \end{array}\right.\right\}.$$

The expression represent the oracles O_1, O_2. Additionally, $m^{(1)}$ and $m^{(2)}$ are required to have the same length ; neither A_1 nor A_2 are allowed to query O_1 on the challenge identity id_{ch}, and A_2 can not query O_2 on the challenge pair (id_{ch}, c). These queries may be asked adaptively (like CCA2 after phase 2), that is, each query may depend on the answers obtained to the previous queries.

2.5 Problem Bilinear of Diffie Hellman

During all this article we use the multiplicative expression instead of the additive one to simplify the proof of security. So we will give the following definition in the multiplicative expression.

Definition 1 :

(k+2-Bilinear Diffie Hellman Exponent Problem (k+1-BDHIP) [see [13]]). Let k be an integer, and x $\in Z_q^*$, $g \in G_2^*$, $g = \psi(g')$, ê : $G_1 \times G_2 \longrightarrow G_T$. Given $(g', g, g^x, g^{x^2}, ..., g^{x^k})$, compute ê$(g', g)^{x^{k+1}}$ is difficult.

Definition 2 :

(k+2- Diffie Hellman Exponent Problem (k+1-BDHIP) [see [13]]). Let k be an integer, and x $\in Z_q^*$, $g \in G_2^*$, $g = \psi(g')$, ê : $G_1 \times G_2 \longrightarrow G_T$. Given $(g', g, g^x, g^{x^2}, ..., g^{x^k})$, compute $g^{x^{k+1}}$ is difficult.

Definition 3 :

(k-Bilinear Diffie Hellman Inversion Problem (k-BDHIP) [see also[13]]). Let k be an integer, and x $\in Z_q^*$, $g \in G_2^*$, $g = \psi(g')$, ê : $G_1 \times G_2 \longrightarrow G_T$. Given $(g'g, g^x, g^{x^2}, ..., g^{x^k})$, compute ê$(g', g)^{\frac{1}{x}}$ is difficult.

Definition 4 :

(k- Diffie Hellman Inversion Problem (k-DHIP) [see also[13]]). Let k be an integer, and x $\in Z_q^*$, $g \in G_2^*$, $g = \psi(g')$, ê : $G_1 \times G_2 \longrightarrow G_T$. Given $(g'g, g^x, g^{x^2}, ..., g^{x^k})$, compute $g^{\frac{1}{x}}$ is difficult.

Definition 5 :

(Bilinear Diffie-Hellman Problem BDHP [see[3]]). Let G_1, G_2 two rings with prime order q. Let ê : $G_1 \times G_2 \longrightarrow G_T$ be an application admissible and bilinear and let g be a generator of G_1. The BDHP in $< G_1, G_2,$ ê $>$ is so : Given < g, g^a, g^b, g^c > for a, b, c $\in Z_q$. Calculate ê$(g, g)^{abc}$ $\in G_2$ is difficult.

3 Our Proposition

We have two kind of Pairing : Asymmetric pairing and the Symmetric one. In this latter we use the supersingular curve until in the first we use the ordinary curve. And it is proven in [10] that the asymmetric pairing are more convenient to the security. So in the following version we use this latter and we prove the security of our scheme under them.

Our scheme

Setup. Given a security parameter k, the parameter generator follows the steps. .

1. Generate four cyclic groups G_1, G_2, G_3 and G_T of prime order q, two isomorphism ψ_2, ψ_3 from respectively G_2 to G_1 and G_3 to G_1, a bilinear pairing map ê : $G_2 \times G_1 \longrightarrow G_T$. Pick a random generator $g_1 \in G_1{}^\star$ and set $g_2 = \psi_2(g_1)$, $g_3 = \psi_3(g_1)$
2. pick : a random a which is the residue quadratics of s i.e a=s^2, after pick $Pub_1 = {g_1}^s$, $Pub_2 = {g_1}^a$
3. Pick four cryptographic hash functions $H_1 : \{0,1\}^* \longrightarrow Z_q^*$, $H_2 : G_T \longrightarrow \{0,1\}^n$, $H_3 : \{0,1\}^n \times \{0,1\}^n \longrightarrow Z_q^*$ and $H_4 : \{0,1\}^n \longrightarrow \{0,1\}^n$ for some integer n > 0.

The message space is M = $\{0,1\}^n$. The ciphertext space is C= $G_1^* \times \{0,1\}^n \times \{0,1\}^n$. The master public key is $M_{pk} = \{$ q, G_1, G_2, G_3, G_T , ψ_2, ψ_3, ê, n, $g_1, g_2, g_3, Pub_1, Pub_2, e(g_1,g_3) = l, e(g_1,g_3)^a = l^a$, H_1, H_2 ; H_3, H_4 $\}$, and the master secret key is M_{sk} = s, a.
Extract. Given a $ID_A \in \{0,1\}^*$ of an entity A, M_{pk} and M_{sk}, the algorithm pick a random r_{ID} and returns $d_A = (r_{ID},\, ({g_2}^{-r_{ID}} g_3)^{\frac{H_1(ID_A)}{s} + \frac{s}{H_1(ID_A)}})$.
Encrypt. Given a plaintext m ∈ M, ID_A and M_{pk}, the following steps are performed.

1. Pick a random $\sigma \in \{0,1\}^n$ and compute r = $H_3(\sigma, m)$.
2. Choose an arbitrary r and compute g_1^r, g_1^{sr}, g_1^{ar}
3. The ciphertext is C = (g_1^r, g_1^{sr}, g_1^{ar}, $\sigma \oplus H_2(l^{r(a+({H_1}^2(ID_A))})$ = $\sigma \oplus H_2(l^{ra} l^{r{H_1}^2(ID_A)}), m \oplus H_4(\sigma)$)=(u,v,w,x,y)

Decrypt. Given a ciphertext C = (u,v,w,x,y) ∈ C, ID_A, d_A and M_{pk}, follow the steps :

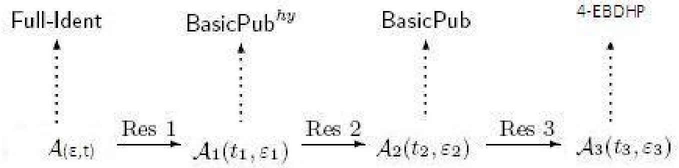
1. Compute z=ê($v^{H_1(ID_A)}, d_{ID}$)ê($w^{r_{ID}} u^{r_{ID}{H_1}^2(ID_A)}, g_2$)
2. Compute $x \oplus H_2(z) = \sigma'$.
3. Compute $y \oplus H_4(\sigma') = m'$ and r'=$H_3(\sigma', m')$
4. Verify if u $\neq g_1^{r'}$ or v$\neq g_1^{sr'}$ or w$\neq g_1^{ar'}$, output ⊥, else return m' as the plaintext.

4 Prove of Security

The security of our scheme can be reduce to the hardness of the 4-BDHEP problem. The reduction is similar to the proof of BF-IBE [3] and Skai Kasarah [13] and as [3,13] we will take into remarque the revision of Galindo [11] in our prove.
The following theorem represent the level of security of our scheme :
Theorem : Our scheme is secure against IND-ID-CCA adversaries with the fact that $H_i(1 \leq i \leq 4)$ are random oracles and 4-EBDHP assumption is rigid. Suppose that there exists an IND-ID-CCA adversary A against our scheme that has advantage ε(k) and running time t(k). Suppose also

that during the attack A makes at most q_d decryption queries and at most q_i queries on H_i for $(1 \leq i \leq 4)$ respectively (note that H_i can be queried directly by A or indirectly by an extraction query, a decryption query or the challenge operation). Then there exists an algorithm A_3 to solve the 4-EBDHP problem with advantage $Adv_{A_3}(k)$ and running time $t_{A_3}(k)$ where :
$Adv_{A_3}(k)=\frac{1}{q_1^2(q_3+q_4)}[(\frac{\varepsilon(k)}{q_1}+1)(1-\frac{2}{q_1}q)^{q_d}-1]$
$t_{A_3}(k) \leq t(k) + O((q_3+q_4)(n+logq) + q_d(\tau_1+\tau_2+\chi))$

Proof : The prove follows immediately as the method of Boneh and Franklin three reduction.



To distinguish these Res_i i ∈ {1,2,3} we combine three lemma :
In lemma 1 we prove that if there exists an IND-ID-CCA adversary, who is able to break our scheme by launching the adaptive chosen ciphertext attacks as defined in the security model [13], then there exists an IND-CCA adversary to break a scheme defined by **BasicPu**b^{hy}. The goal of this step is to shows that private key extraction queries do not help the adversary.
In lemma 2 we show that if such IND-CCA adversary exists, then there must exist an IND-CPA adversary that breaks the corresponding **BasicPub** scheme by merely launching the chosen plaintext attacks. The goal of this step will be end in lemma 3 and so that the adversary the not benefit from the extraction of the private.
Finally, in Lemma 3 we prove that if the **BasicPub** scheme is not secure against an IND-CPA adversary, then the corresponding 4-EBDHP assumption will be attacked.

Lemma 1 : Suppose that H_1 is a random oracle and that there exists an IND-ID-CCA adversary A against our scheme with advantage $\varepsilon(k)$ which makes at most q_1 distinct queries to H_1 (note that H_1 can be queried directly by A or indirectly by an extraction query, a decryption query or the challenge operation). Then there exists an IND-CCA adversary A_1 which runs in time $O(A) + q_d(\tau_1 + \chi)$ against the following **BasicPu**b^{hy} scheme with advantage at least $\frac{\varepsilon(k)}{q_1{}^2}$
BasicPub^{hy} is specified by three algorithms : **keygen**, **encrypt** and **decrypt**.
keygen :

Given a security parameter k, The challenger give the following parameters to A_1.
K_{pub}=<q, G_1, G_2, G_3, G_T , $\psi_2, \psi_3, k_2 = ord(\psi_2), k_3 = ord(\psi_3)$,ê, n, h_i for i ∈ $\{0, ...q_1\}, g_1, g_2, g_3, Pub_1 = g_1^s, Pub_2 = g_1^{s^2}, e(g_1, g_3)^a, e(g_1, g_3)$, H_2 ; H_3, H_4 >

encrypt :

Pick a random $\sigma \in \{0,1\}^n$ and compute r = $H_3(\sigma, m)$.
Compute g_1^r, g_1^{sr}, g_1^{ar}
The ciphertext is $C_i = (g_1^r,\ g_1^{sr},\ g_1^{ar},\ \sigma \oplus H_2((l)^{r(a+(h_i{}^2(ID_A))}) = \sigma \oplus H_2(l^{ra}l^{h_i{}^2(ID_A)}), m \oplus H_4(\sigma))$=$(u_i, v_i, w_i, x_i, y_i)$

decrypt :

Given a ciphertext $C = (u_i, v_i, w_i, x_i, y_i) \in$ C, ID_A, d_A and M_{pk}, follow the steps :

1. Compute z_i=ê$(v_i^{h_i(ID_A)}, d_{ID})$ê$(w_i^{r_{ID}} u_i^{r_{ID} {h_i}^2(ID_A)}, g_2)$
2. Compute $x_i \oplus H_2(z_i) = \sigma'$.
3. Compute $y_i \oplus H_4(\sigma') = m'$ and r'=$H_3(\sigma', m')$
4. Verify if $u_i \neq g_1^{r'}$ or $v_i \neq g_1^{sr'}$ or $w_i \neq g_1^{ar'}$, output $\perp$, else return m' as the plaintext.

We construct an adversary A_1 that mounts an IND-CCA attack on the $BasicPub^{hy}$ scheme with the public key K_{pub} using the help of A. The attack will be as follows.
Firstly we note by $\mathbf{BasicPu}b_{1-H_1}{}^{hy}$ when the adversary A_1 has the advantage to ask the challenger, the queries of his $H_1(ID_{A_1})$. If the challenger responds to him, the responds will be as follow :
$(r_{ID_{A_1}}, ({g_2}^{-r_{ID_{A_1}}} g_3)^{\frac{H_1(ID_{A_1})}{s} + \frac{s}{H_1(ID_{A_1})}}, H_1(ID_{A_1}))$
So algorithm A_1 can calculate $({g_2}^{-r_{ID_{A_1}}} g_3)^{\frac{s}{H_1(ID_{A_1})}}$ as he know Pub_2, Pub_1, ψ_2 and ψ_3
Then he calculate $({g_2}^{-r_{ID_{A_1}}} g_3)^{\frac{H_1(ID_{A_1})}{s} + \frac{s}{H_1(ID_{A_1})}} . ({g_2}^{-r_{ID_{A_1}}} g_3)^{-\frac{s}{H_1(ID_{A_1})}}$
So algorithm A_1 can compute easily $({g_2}^{-r_{ID_{A_1}}} g_3)^{\frac{1}{s}}$
For an h_i let a_{h_i} such that $r_{h_i} + a_{h_i}$=$r_{ID_{A_1}}$ we have so : $({g_2}^{-r_{h_i} - a_{h_i}} g_3)^{\frac{1}{s}}$
We can calculate the exact key if k_2=$\frac{a_{h_i}}{s}$ or a_{h_i} = ord(g_2), if not it can abort
A_1 check this by verifying $\psi_2(Pub_1)^{k_2} = \psi_2(g_1)^{a_{h_i}}$
Note that the second part of the key : $({g_2}^{r_{h_i}} g_3)^{\frac{s}{h_i}}$ can be calculate easily as we know $Pub_1, Pub_2, \psi_2, \psi_3$
Algorithm A_1 simulates the algorithm Setup of our scheme for A by supplying A with the master public key M_{pk} = { q, G_1, G_2, G_3, G_T , ψ_2, ψ_3, ê, n, $g_1, g_2, g_3, Pub_1, Pub_2, e(g_1, g_3)^a, e(g_1, g_3)$, H_1, H_2 ; H_3, H_4 } where H_1 is a random oracle controlled by A_1. A_1 does not know the master secret key {s,a}.
Adversary A can make queries on H_1 at any time. These queries are handled by the following algorithm.
H_1-query (ID_i) :

> A_1 maintains a list of tuples (ID_i, h_i, d_i) indexed by ID_i as explained below. We refer to this list as ${H_1}^{list}$. The list is initially empty. When A queries the oracle H_1 at a point ID_i, A_1 responds as follows :
>
> 1. If ID_i already appears on the ${H_1}^{list}$ in a tuple (ID_i, h_i, d_i), then A_1 responds with $H_1(ID_i) = h_i$.
> 2. Otherwise, A_1 selects a random integer h_i(i > 0) from K_{pub} which has not been chosen and use the method we announced above and stores the tuple into the list. A_1 responds with $H_1(ID_i) = h_i$.

Phase 1 :

> A launches Phase 1 of its attack, by making a series of requests, each of which is either an extraction or a decryption query. A_1 replies to these requests as follows.
> **Extraction query** (ID_i) : A_1 first looks through list ${H_1}^{list}$. If ID_i is not on the list, then A_1 queries $H_1(ID_i)$. A_1 then checks the value d_i : if $d_i \neq \perp$, A_1 responds with d_i ; otherwise, A_1 aborts the game (**Event 1**).
> **Decryption query** (ID_i, c_i) : A_1 first looks through list ${H_1}^{list}$. If ID_i is not on the list, then A_1 queries $H_1(ID_i)$. If $d_i = \perp$, then A_1 sends the decryption query $c_i = (u_i, v_i, x_i, y_i, z_i)$ to C and simply relays the plaintext got from C to A directly. Otherwise, A_1 decrypts the ciphertext b

Challenge :

At some point, A decides to end Phase 1 and picks ID_{ch} and two messages (m_0, m_1) of equal length on which it wants to be challenged. Based on the queries on H_1 so far, A_1 responds differently.

1. If the query on H_1 has been issued, and so d_{ch}=$\perp$, A_1 continues,
   – Otherwise, A_1 aborts the game (**Event 2**).
2. if the tuple corresponding to ID_{ch} is on the list ${H_1}^{list}$ (and so d_{ch} $\neq\perp$), then A_1 aborts the game (**Event 3**)

A_1 passes C the pair (m_0, m_1) as the messages on which it wishes to be challenged. C randomly chooses b $\in$ {0,1} encrypts m_b and responds with the ciphertext C_{ch} = (u', v' w', x',y'). Then A_1 forwards C_{ch} to A.

Phase 2 :

A_1 continues to respond to requests in the same way as it did in Phase 1. Note that the adversary will not issue the extraction query on ID_{ch} (for which d_{ch} =$\perp$) and the decryption query on (ID_{ch}, C_{ch}).

Observation : A_1 will not abort the game in phase 2, as it is not allowed to answer the queries of ID_{ch} and C_{ch}.

Guess :

A makes a guess b' for b. A_1 outputs b' as its own guess.
This simulation (study) is identical to the real attack if it does not abort.

Claim :

If the algorithm A_1 does not abort during the simulation then algorithm A's view is identical to its view in the real attack.

Proof : A_1's responses to H_1 queries are uniformly and independently distributed in Z_q as in the real attack because all response are random and are valid, if A_1 does not abort.
It remain to us, to calculate the probability of not aborting during simulation.
$Pr[A_1 doesnotabort] = Pr[\neg\, event_1 \wedge \neg\, event_2 \wedge \neg\, event_3]$
= Pr[$\neg$ $event_1$]. Pr[$\neg$ $event_1$ / ($\neg$ $event_2$ $\wedge$ $\neg$ $event_3$)]=$\frac{1}{{q_1}^2}$
With time $t_1 = O(A) + q_d(\tau_1 + \chi)$
Where τ_1, χ are respectively the time to calculate the exponentiation and the pairing

The following lemma is a fruit of the result of Fujisaki and Okamoto (Theorem 14 in [12]). With the fact that $BasicPub^{hy}$ is built by applying Fujisaki-Okamoto transformation to a version basic of our scheme (without provide H_3, H_4 in our full version). We remember the basic version in the following

Lemma 2 : Let H_3, H_4 be random oracles. Let A_1 be an IND-CCA adversary against $BasicPub^{hy}$ defined in Lemma 1 with advantage ε_1(k). Suppose A_1 has running time $t_1(k)$, makes at most q_d decryption queries, and makes q_3 and q_4 queries to H_3 and H_4 respectively. Then there exists an IND-CPA adversary A_2 against the following BasicPub scheme, defined by three algorithms : **keygen**, **encrypt** and **decrypt**.

keygen :

Given a security parameter k.

1. The preparation step will be the same as $BasicPub^{hy}$, except that we eliminate H_3 and H_4. But we will late H_2.

2. Pick a hash function $H_2 : G_T \longrightarrow \{o,1\}^n$, M $= \{0,1\}^n$. The ciphertext space is C$=$ $G_1^* \times \{0,1\}^n \times \{0,1\}^n$. The master public key is $M_{pk} = \{$ q, G_1, G_2, G_3, G_T , ψ_2, ψ_3, ê, n, $g_1, g_2, g_3, Pub_1, Pub_2, e(g_1,g_3) = l, e(g_1,g_3)^a = l^a$, h_i, i $\in \{o,1\}^n$, H_2 $\}$, and the master secret key is $M_{sk} = \{$s,a$\}$

encrypt

Choose an arbitrary r $\in Z_q$ and compute g_1^r, g_1^{sr}, g_1^{ar}

The ciphertext is $C_i = (g_1^r,\, g_1^{sr},\, g_1^{ar},\, m \oplus H_2(l)^{r(a+({h_i}^2(ID_A)))}) = (u_i, v_i, w_i, x_i)$

decrypt :

Given a ciphertext $C = (u_i, v_i, w_i, x_i) \in$ C, ID_A, d_A and M_{pk}, follow the steps :

1. Compute z_i=ê$(v^{h_i(ID_A)}, d_{ID})$ê$(w^{r_{ID}} u^{r_{ID}{h_i}^2(ID_A)}, g_2) = e(g_1,g_2)^{a+h_i(ID_A)^2}$
2. Compute $x_i \oplus H_2(z_i) = m$.

According to [12] A_2 has the following advantage $\varepsilon_2(k)$, and the following time t_2

$$\varepsilon_2(k) \geq \tfrac{1}{2(q_3+q_4)}[(\varepsilon_1(k)+1)(1-\tfrac{1}{2})^{q_D} - 1]$$
$$\text{And } t_2(k) \leq t_1(k) + O((q_3+q_4).(n+logq))$$

Lemma 3 Suppose that if there exists an IND-CPA adversary A_2 against the BasicPub defined in Lemma 2 which has advantage $\varepsilon_2(k)$ and queries at most q_2 times H_2 (H_2 is a random oracle). Then there exists an algorithm A_3 to solve the 4-BDHE problem with advantage at least $\frac{\varepsilon_2}{2}$ and running time $O(time(A_2) + q_d\tau_2)$
where τ_2 is the time to calculate the exponentiation in G_2.
Algorithm A_3 is given as input a random 4-BDHE instance $\{$ q,$G_1, G_2, G_3, G_T, \psi_2, \psi_3$,, ê, $k_2, k_3, Q_1, P_{pub} = {Q_1}^x, Q_3$, / $k_3 - k_2 = ord(g_1)$ $\}$ where x is a random element from Z_q
And Q_1, Q_2, Q_3 will be determined latter
The private key is : $d_{partiel} = (r_{ID}, (Q_2^{-r_{ID}} Q_3)^x)$.
We give to the algorithm A_2 : $\{$ $g_1, g_2; g_3, {g_1}^x, {g_1}^{x^2}, {g_1}^{x^3}$ $\}$ Algorithm A_2 finds ê$(g_1,g_2)^{\frac{1}{x}}$ or ê$(g_1,g_3)^{\frac{1}{x}}$ (note that if we can calculate ê$(g_1,g_2)^{\frac{1}{x}}$ we can calculate ê$(g_1,g_3)^{\frac{1}{x}}$, because of ψ_2, ψ_2) by interacting with A_2 as follows :
Algorithm A_3 compute f(x)=$\sum^2{}_{i=0} c_i x^i$ with

$$c_0 = \begin{cases} 0 & A_1 \text{ didn't receive } d_{ID_{A_3}} \text{ for his } H_1 \text{ in phase 1} \\ 1 & A_1 \text{ receive } d_{ID_{A_3}} \text{ for his } H_1 \end{cases}$$

If $c_0 = 0$, A_1 can calculate firstly the queries in phase 1 in the following manner, with a condition that the challenger publish M_{pub}=$\{$ q,$G_1, G_2, G_3, G_T, \psi_2, \psi_3$,, ê, $k_2, k_3, Q_1 = {g_1}^x, P_{pub} = {Q_1}^x, Q_3$, / $k_3 - k_2 = ord(g_1)$ $\}$
So we have : for each h_j, $\frac{f(x-h_j)-f(-h_j)}{x} = c_1 - 2c_2h_j + c_2x$=E
Also we have $x(f(x-h_j) - f(-h_j))$=$x(c_1 - 2c_2h_j) + c_2x^2$=F
So ${g_1}^E$ and ${g_1}^F$ can be calculate easily
So ${g_1}^{k_3E}$= $({g_1}^{k_3(c_1x+c_2x^2)}{g_1}^{-k_2(2c_2xh_j)})^{\frac{1}{x}}$= $({g_3}^{(c_1x)}{g_2}^{-(2c_2x)h_j})^{\frac{1}{x}}{g_3}^{(c_2x)}$
Then if we pose $Q_2 = {g_2}^x$ and $Q_3 = {g_3}^{c_1x}$
$({g_1}^{k_3E})^{h_j}({g_3}^{c_2x})^{-h_j}$=$({Q_2}^{-2c_2h_j}Q_3)^{\frac{h_j}{x}}$= $({Q_2}^{-r_{h_j}}Q_3)^{\frac{h_j}{x}}$; with $r_{h_j} = 2c_2h_j$
Also we have with the same method :$({g_1}^{k_3F})^{\frac{1}{h_j}}({g_3}^{c_2x^3})^{-\frac{1}{h_j}}$= $({Q_2}^{-r_{h_i}}Q_3)^{\frac{x}{h_j}}$
As a result we have : $({g_1}^{k_3E})^{h_i}({g_3}^{c_2x})^{-h_i}({g_1}^{k_3F})^{\frac{1}{h_j}}({g_3}^{c_2x^3})^{-\frac{1}{h_j}}$ =$({Q_2}^{-r_{h_j}}Q_3)^{\frac{h_j}{x}+\frac{x}{h_j}}$
So we can anser to the querie and we can calculate trivially : ê$({P_{pub}}^{h_j}$,$({Q_2}^{-r_{h_j}}Q_3)^{\frac{h_j}{x}+\frac{x}{h_j}}$)= ê$(Q_1, {Q_2}^{-r_{ID}}Q_3)^{a+{h_j}^2}$

But if not i.e $c_0 \neq 0$ we have so :
Phase 1 will be unroll as in lemme 1
The adversary A_3 can calculate $({Q_2}^{-r_{h_j}} Q_3)^{\frac{h_j}{x}+\frac{x}{h_j}}$ as the method cited above because we have :
$f(x-h_j) - f(-h_j)$=$c_1x + c_2x^2 - 2c_2h_j$

In both case ($c_0 \neq 0$ and $c_0 = 0$) algorithm A_3 can calculate : $({Q_2}^{-r_{h_j}} Q_3)^{\frac{h_j}{x}+\frac{x}{h_j}}$
So we have :
ê(${P_{pub}}^{h_j}$,$({Q_2}^{-r_{h_j}} Q_3)^{\frac{h_j}{x}+\frac{x}{h_j}}$)= ê$(Q_1, {Q_2}^{-r_{ID}} Q_3)^{a+{h_j}^2}$
And :
ê(${P_{pub}}^{h_j}$,${d_{partiel}}^{\frac{1}{h_j}} {d_{complete}}^{h_j}$)=ê$(Q_1, {Q_2}^{-r_{ID}} Q_3)^{a+{h_j}^2}$

With $d_{complete}$= $({g_1}^{k_3E})({g_3}^{c_2x})$= $({Q_2}^{-r_{h_j}} Q_3)^x$ which is calculate easily by A_3
In recap M_{pub} is a valid public key of BasicPub.
Now A_3 starts to respond to queries as follows. $H_2 - query(X_i)$: At any time algorithm A_2 can issue queries to the random oracle H_2. To respond to these queries A_3 maintains a list of tuples called ${H_2}^{list}$. Each entry in the list is a tuple of the form (X_i, ζ_i) indexed by X_i. To respond to a query on X_i, A_3 does the following operations :

1. If on the list there is a tuple indexed by X_i, then A_3 responds with ζ_i
2. Otherwise, A_3 randomly chooses a string $\zeta_i \in \{0,1\}^n$ and inserts a new tuple (X_i, ζ_i) to the list. It responds to A_2 with ζ_i.

Challenge :

Algorithm A_2 outputs two messages (m_0, m_1) of equal length on which it wants to be challenged. A_3 chooses a random string R $\in \{0,1\}^n$ and a random element r $\in Z_q$, and defines C_{ch} = $({Q_1}^r, {Pub_1}^r, R)$. A_3 gives C_{ch} as the challenge to A_2. Observe that the decryption of C_{ch} is R(H_2(ê$({Pub_1}^r, {Q_2}^{-r_{ID}} Q_3)^x)^{-1}$ê$({Pub_1}^r, {Q_2}^{-r_{ID}} Q_3)^{\frac{1}{x}}$))

Guess :

After algorithm A_2 outputs its guess, A_3 picks a random tuple (X_i, ζ_i) from H_{list}
Remember that ê$(Pub_1, d_{partial})$=ê$(Pub_1, ({Q_2}^{-r_{ID}} Q_3)^x)$=ê$(g_1, g_2)^{-r_{ID}x^4}$.ê$(g_1, g_3)^{x^4}$
and ê$(Q_1, ({Q_2}^{-r_{ID}} Q_3)^x)$=ê$(g_1, g_2)^{-r_{ID}x^3}$.ê$(g_1, g_3)^{x^3}$
We claim that ê(Pub_1, d_{compl})=ê$(Pub_1, ({Q_2}^{-r_{ID}} Q_3)^{\frac{1}{x}})$=ê$(g_1, g_2)^{-r_{ID}x^2}$.ê$(g_1, g_3)^{x^2}$
and ê(Q_1, d_{compl}) = ê$(Q_1, ({Q_2}^{-r_{ID}} Q_3)^{\frac{1}{x}})$=ê$(g_1, g_2)^{-r_{ID}x}$.ê$(g_1, g_3)^x$ are easy to calculate

Let D be the event that algorithm A_2 issues a query for H_2(ê$((Pub_1, d_{partial})$)) at some point during the above simulation. To test if this latter work as in the real attack we need to two claim (this technique was used by [3] we remember it only)
Claim 1 : Pr[D] in the simulation above is equal to Pr[D] in the real attack.
Claim 2 : In the real attack we have Pr[D] $\geq 2\varepsilon_2(k)$.
So as a recap we can say that A_3 produces the correct answer if he success to compute 4-BDHEP and if he work as in the real attack. This latter has a probability at least $\frac{2\varepsilon_2(k)}{q_2}$.
And the time to realise this lemma is $O(time(A_2) + q_d\tau_2)$ where τ_2 is the time to calculate the exponenetiation

We have so lemma1+lemma2+lemma3 = Theorem

5 Efficiency

In this section we will compare our scheme with the existed scheme in the random oracle. And as we are intersted to skirt around the weekness of those existed scheme, we will concentrate firstly in the security.

5.1 Comparison in the level Security

To make a comparison in the level security we will cite as a schedule : 1-Problem bilineair of Diffie Hellman, 2-Projection in Elliptic Curve, 3-Symetrique or Asymetrique-Pairing, since :
Study the rigidity of the problem of Diffie Hellman used in the study of simulation of these cryptosystems, give us their weight against passive adversary (CPA) and malicious adversary (CCA2)
And as the projection in the Elliptic Curve limit the selection of the elliptic curve to be used, which pose the problem of security. We will signal so if the cryptosystem considered has a projection in the elliptic curve or not.
For the pairing, because of the danger of the problem MOV [14] caused by the use of the supersingular curve and this can be affected if we use the symetrique pairing. We will signal if the cryptosystem function with symetric pairing or with asymetric pairing

	BF	SK	BB1	Our
Problem bilineair of Diffie Hellman	BDHP	q-BDHIP	BDHP (not sure)	4-EBDHP
Projection in Elliptic Curve	Yes	No	No	No
Sym/Asym-Pairing	Asym but with ver [11]	Asym but with ver [13]	Asym	Asym

Remmebring that [3] have used symmetric pairing and the asymmetric pairing are used in the revisison of Galnido[11]
The version of Chen and Cheng [13] use also asymmetric pairing

5.1.1 A look in the comparison

Look for : Problem bilineair of Diffie Hellman

To compare this poblem of Diffie Helman, we firstly make the following reduction :

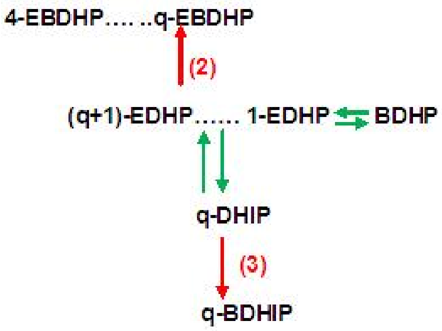


Fig. 1 – Some relations

With k-A ⟶ k-B : if k-A is polynomial-time solvable, so is k-B ;
The relation in green are proved in [13]
The relations in red are trivial since :

To demonstrate for example the relation (3)we have :
Given $(g, g^x, g^{x^2}, g^{x^3}, ..., g^{x^k})$ if we can compute $g^{\frac{1}{x}}$ we can also compute $\hat{e}(g, g)^{\frac{1}{x}}$.
Note that (2) can be done with the same manner, we will only make the following reduction
(q+1)-EDHP $\longrightarrow$ q-EDHP
But Cheon [9] in Eurocrypt show the following :
If g, g^{α}, and g^{α^d} are given for a positive divisor d of p-1, we can compute the secret α in $O(logp(\sqrt{p/d} + \sqrt{d}))$ group operations using $O(max\{\sqrt{p/d} + \sqrt{d}\})$ memory. If g^{α^i} (i = 0, 1, 2,..., d) are provided for a positive divisor d of p + 1, α can be computed in $O(logp(\sqrt{p/d} + d))$ group operations using $O(max\{\sqrt{p/d} + \sqrt{d}\})$ memory. This implies that the strong Diffie-Hellman problem and its related problems have computational complexity reduced by $O(\sqrt{d})$ from that of the discrete logarithm problem (DPL) for such primes.
So if we examine this we can say that if d is long, these problem will be as small as reduced from PDL, so it become less rigid and easy to be attacked
In [13] the q is related to q_H (because we construct the oracle in the BasicPub from q-BDHIP-see the [13]). Following [15] we need $q_H = 2^{50}$ for a level of security equal to 80. And we will augment this for a higher level of security
So as we have $2^2 < 2^{50}$, the 4-EBDHP guarantee more security compared to q-BDHIP
We can say the same for BDHP as it has the same level as 1-EDHP (see figure 1)
According to this, only our scheme, Boneh Franklin and BB1 are efficient.
But as the scheme of Boneh and Franklin cannot be separate from the projection in elliptic curve. Which limit the selection of these latter and this pose the problem of security. It make to us only our scheme and that's of BB1 [5][16](in [16] Boyen have prove the benefit of BB1 by comparison with [3][4]) as an efficient scheme in the point of view security.
But we are not sure about the problem of Diffie Hellman used by BB1, because we haven't the exact proof of security (the proof was made with sID-CPA).
In the meantime of this we move to compare our scheme and BB1 in the point of view complexity.

5.2 Comparison with BB1 : Compute of Complexity

To compare our scheme with BB1 we remember firstly this latter (version given in [17]).

Scheme of Boneh Boyen : Full version

Setup :

To generate IBE system parameters, pick α, β
$\gamma \in Z_p$. Set $g_1 = g^{\alpha}$ and $g_3 = g^{\gamma}$ in G, and compute v_0 = e(g, ĝ)$^{\alpha\gamma}$.
(Note that $g_2 = g^{\beta}$ is not needed.) The public system parameters params and the master secret key masterk are given by :
params = (g, $g_1, g_3, v_0) \in G_3 \times G_t$, masterk = (ĝ, $\alpha, \beta, \gamma) \in$ Ĝ $\times Z_p{}^3$.
The generator ĝ need not be kept secret as it is needed by the authority, it can be retained in masterk rather than published in params.

Extract :

To generate a private key d_{ID} for an identity ID $\in$ {0,1}*, using the master key, the trusted authority picks a random r $\in Z_p$ and outputs : d_{ID} = (ĝ$^{\alpha\gamma+(\alpha H_1(ID)+\gamma)r}$, ĝ^r) $\in$ Ĝ $\times$ Ĝ.

Encrypt :

To encrypt a message M $\in$ {0, 1}^{l} for a recipient ID $\in$ {0,1}*, the sender first picks a random s $\in Z_p$, computes k = ${v_0}^s \in G_t$, assigns c = M $\oplus H_2(k) \in$ {0,1}^{l}, calculates $c_0 = g^s$ and $c_1 = {g_3}^s {g_1}^{H_1(ID)s}$ in G, sets t = s + $H_3(k, c, c_0, c_1)$ mod p, and then outputs :
C = (c, $c_0, c_1, t) \in$ {0,1}^{l} $\times G \times G \times Z_p$.

Decrypt :

To decrypt a given ciphertext C = (c, c_0, c_1, t) using the private key $d_{ID} = (d_0, d_1)$, the recipient computes : k =$e(c_0, d_0)/e(c_1, d_1) \in G_t$, s = t - $H_3(k, c, c_0, c_1) \in Z_p$. Then, if the component-wise equality (k,c_0) $=^?$ $({v_0}^s, g^s)$ does not hold for both elements, the ciphertext is rejected. Otherwise, the plaintext is given by : M = c $\oplus H_2(k) \in$ {0, 1}^{l}.

Compute of Complexity

	BB1	Our Scheme
Params	$2Exp_G + 1Mul_{Z_p} + 1pairing + 1Ex_{F_{q^k}}$	$2Exp_G + 1Mul_{Z_p} + 1pairing + 1Ex_{F_{q^k}}$
Extract	$3Mul_{Z_p} + 2Exp_G$	$2div_{Z_p} + 2Exp_G$
Encrypt	$1Exp_{F_{p^k}} + 1Mul_{Z_p} + 3Exp_G$	$3Exp_G + 2Exp_{F_{p^k}} + 2Mul_{Z_p}$
Decrypt	$2pairing + 1inv_{F_{p^k}} + 1Exp_{F_{q^k}} + 1Exp_G$	$2Mul_{Z_p} + 4Exp_G + 2pairing$
Sum	$3pairing + 1div_{F_{p^k}} + 3Exp_{F_{p^k}} + 8Exp_G + 5Mul_{Z_p}$	$3pairing + 2div_{Z_p} + 3Exp_{F_{p^k}} + 11Exp_G + 5Mul_{Z_p}$

In $Exp_{F_{p^k}}$ we have the exponent in F_p. But its base is in F_{p^k}
And for $div_{F_{p^k}}$ we make the division in F_{p^k}
With F_{p^k} is a finite field constructed using the quotient $F_p[X]/P(X)F_p[X]$) with :
$F_p[X]$ is a set of polynomials with coefficient in F_p
P(X) is an irreducible polynomials in F_p with degree k
$P(X)F_p[X]$ is is the set of polynomials which has P(X) as factor (or divided by P(X)).
According to this table, we can balance between BB1 and our scheme since :
$Sum_{BB1} - Sum_{ourscheme}$=$1div_{F_{p^k}} - 2div_{Z_p} - 1Exp_{Z_p} - 3Exp_G$
We can balance $1div_{F_{p^k}}$ with $2div_{Z_p}$ because in $1div_{F_{p^k}}$ we make the div of two polynomials and after we calculate modulus P(x) (which is a hide div) □
We have an overstepping by $3Exp_G$ by comparison with BB1, because we make an r_{ID} in Extract which help us in the proof of CCA2. And if we remove it we can remove so an $1Exp_G$ in Extract and $2Exp_G$ in Decrypt which is $3Exp_G$.
By contrast the BB1 was proved to be only CPA in the selective ID (introduced by[18]) which is a weaker notion [19] and to prove it CCA2 in the random oracle we need another look to BB1 □

6 Conclusion

We have presented in this article with a proof of security in the random oracle, an efficient scheme in the point of view security. Our scheme is based on 4-EBDHP which is more efficient than q-BDHIP used by Skai Kasarah. More than that our scheme project into Z_p by comparison with Boneh and Franklin which project into elliptic curves. This latter is less efficient as it limit the selection of the elliptic curve. The only scheme which can guarantee the same level of security as our in the random oracle is BB1. But it's security is not sure, because it was proved only with sID-CPA and to prove it in the ID-CCA2 we can need to change the look of BB1. While waiting to prove this and using

the syntax given in this article, our scheme offer a competitive to BB1.
Thus in this article we give a fourth efficient scheme in the random oracle for the cryptography IBE.

Acknowledge

We thank gratefully the chef of our laboratory LRIT Mr Aboutajdinne Driss

Références

[1] M. Bellare and P. Rogaway. Random oracles are practical : a paradigm for designing efficient protocols. In Proceedings of the First Annual Conference on Computer and Communications Security, ACM, 1993.

[2] Gaëtan Leurent and Phong Q. Nguyen. How risky is the random-oracle model ? In Halevi [18], pages 445464.

[3] D. Boneh and M. Franklin. Identity based encryption from the Weil pairing. SIAM Journal on Computing, 32(3) :586-615, 2003.

[4] R. Sakai and M. Kasahara. ID based cryptosystems with pairing on elliptic curve. Cryptology ePrint Archive, Report 2003/054.

[5] D. Boneh and X. Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, Advances in Cryptology - EUROCRYPT 2004, volume 3027, pages 223-238, 2004.

[6] A. Shamir. Identity-based cryptosystems and signature schemes. In G. R. Blakley and David Chaum, editors, Advances in Cryptology - CRYPTO'84, volume 196 of Lecture Notes in Computer Science, pages 47-53. Springer-Verlag, 1985.

[7] Michael Scott. A Note on Boneh and Franklin IBE. Available on : ftp ://ftp.compapp.dcu.ie/pub/crypto/note.pdf

[8] Brier, E., Coron, J.S., Icart, T., Madore, D., Randriam, H., Tibouchi, M. : Efficient indifferentiable hashing into ordinary elliptic curves. In : Advances in Cryptology CRYPTO 2010. Lecture Notes in Computer Science, vol. 6223, pp. 237254. Springer (2010)

[9] J. Cheon. Security analysis of the strong Diffie-Hellman problem. In Serge Vaudenay, ed- itor, EUROCRYPT 2006, volume 4004 of LNCS, pages 1-11. Springer-Verlag, Berlin, Germany, May / June 2006.

[10] S. Galbraith, K. Paterson, and N. Smart. Pairings for cryptographers. Discrete Applied Mathematics, 156(16) :3113-3121, 2008.

[11] D. Galindo. Boneh-Franklin identity based encryption revisited. In Proceedings of the 32nd International Colloquium on Automata, Languages and Programming, ICALP 2005. Also available on Cryptology ePrint Archive, Report 2005/117.

[12] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Proceedings of Advances in Cryptology - CRYPTO '99, LNCS 1666, pp. 535-554, Springer-Verlag, 1999.

[13] L. Chen, Zh. Cheng ıı Security Proof of Sakai-Kasahara's Identity-Based Encryption Scheme ıı In Proceedings of Cryptography and Coding 2005.

[14] A. Menezes, T. Okamoto, S. Vanstone q Reducing elliptic curve logarithms to logarithms in a finite field q IEEE Tran. on Info. Th., Vol. 39, pp. 1639-1646, 1993.

[15] E. Kiltz, Y. Vahlis. CCA2 Secure IBE : Standard Model Efficiency through Authenticated Symmetric Encryption. CT-RSA 08, Lecture Notes in Computer Science Vol. , T. Malkin ed., Springer-Verlag, 2008.

[16] X. Boyen. The BB1 identity-based cryptosystem : A standard for encryption and key encapsu- lation. Submitted to IEEE 1363.3, aug 2006. http ://grouper.ieee.org/groups/1363/. (Cited on page 1, 2,3, 11, 12, 13.)

[17] X. Boyen. A tapestry of identity-based encryption : Practical frameworks compared. International Journal of Applied Cryptography, 1(1) :3-21, 2008.

[18] R. Canetti, Sh. Halevi, and J. Katz. A forward-secure public-key encryption scheme. In Eli Biham, editor, Advances in Cryptology - EUROCRYPT 2003, volume 2656 of Lecture Notes in Computer Science, pages 255-271. Springer-Verlag, 2003.

[19] D. Galindo. A separation between selective and full-identity security notions for identity-based encryption Available on : http ://www.cs.ru.nl/paw/publications/separationresultfinal.pdf

[20] Thomas Icart. How to hash into elliptic curves. In Shai Halevi, editor, CRYPTO, volume 5677 of Lecture Notes in Computer Science, pages 303316. Springer, 2009. M. Bellare, A. Desai, D.

[21] Pointcheval, and Ph Rogaway. Relations among notions of security for public-key encryption schemes, volume 1462 Lecture Notes in Computer Science, pages 26-45. Springer-Verlag, 1998.

[22] Galindo and Ichiro Hasuo. Security Notions for Identity Based Encryption. available on : http ://eprint.iacr.org/2005/253