Advanced Cryptographic Primitives: Lecture 2

Teacher : Benoît Libert

15/09/2014

1 Security proofs in the random oracle model

1.1 The Boneh-Lynn-Shacham signature ([1])

1.1.1 Reminder : The BLS signature

Keygen(λ): choose cyclic groups (G, G_T) of prime order $p > 2^\lambda$ with a bilinear map $e : G \times G \to G_T$ and a generator $g \xleftarrow{R} G$. Choose a hash function $H : \{0,1\}^* \to G$. Generate a key pair (PK, SK) with

$$\begin{aligned} PK &:= \{(G, G_T),\ g,\ X = g^x,\ H\} \\ SK &:= x \in_R \mathbb{Z}_p. \end{aligned}$$

Sign(SK, M)**:** compute and output $\sigma = H(M)^x \in G$.
Verify(PK, M, σ)**:** Return 1 if $e(\sigma, g) = e(H(M), X)$. Otherwise, return 0.

1.1.2 Security

Theorem 1. *The BLS signature scheme is secure against chosen-message attacks in the Random Oracle Model (ROM) if the CDH assumption holds in G.*

Proof. Let $\mathcal{A}$ be an attacker against the BLS signature, with advantage ε. We build an algorithm $\mathcal{B}$ that solves CDH with advantage $\frac{\varepsilon}{c(q+1)}$, where c is a constant and q is the number of signing queries of $\mathcal{A}$.

Algorithm $\mathcal{B}$ takes as input (g, g^a, g^b) and has to compute g^{ab}. To this end, $\mathcal{B}$ defines the public key PK so that $X = g^a$ and also controls the random oracle $H : \{0,1\}^* \to G$.

Hash queries: when $\mathcal{A}$ asks for the hash value $H(M)$, $\mathcal{B}$ responds as follows.

- $\mathcal{B}$ returns the previously defined $H(M)$ if it exists.
- Otherwise, $\mathcal{B}$ flips a coin $b_M \in \{0,1\}$ such that $\Pr[b_M = 1] = \delta$ and $\Pr[b_M = 0] = 1 - \delta$ (δ will be chosen later).

  * if $b_M = 0$, $\mathcal{B}$ defines $H(M) = g^{\alpha_M}$ where $\alpha_M \overset{R}{\leftarrow} \mathbb{Z}_p$.
  * if $b_M = 1$, $\mathcal{B}$ defines $H(M) = (g^b)^{\alpha_M}$ where $\alpha_M \overset{R}{\leftarrow} \mathbb{Z}_p$.

  In both cases, $\mathcal{B}$ stores (M, b_M, α_M) in a list L (initially empty).

Signing queries: when $\mathcal{A}$ wants to obtain a signature for a message M, $\mathcal{B}$ does the following. Without loss of generality, we assume that $\mathcal{A}$ has previously queried $H(M)$ (otherwise, $\mathcal{B}$ can make the hash query $H(M)$ for itself). The list L thus contains an entry (M, b_M, α_M).

- If $b_M = 1$, then $\mathcal{B}$ fails since it does not know $\sigma = H(M)^a = (g^{ab})^{\alpha_M}$.
- If $b_M = 0$, $\mathcal{B}$ computes and returns $\sigma = H(M)^a = (g^a)^{\alpha_M}$.

Output: $\mathcal{A}$ outputs (M^*, σ^*). If $\mathcal{A}$ is successful, its output (M^*, σ^*) satisfies $e(\sigma^*, g) = e(H(M^*), g^a)$, so that $\sigma^* = H(M^*)^a$. Since H is a random function from $\mathcal{A}$'s point of view , $\mathcal{A}$ cannot predict $H(M^*)$ with non-negligible probability without explicitly making the hash query $H(M^*)$. So, we can assume that $\mathcal{A}$ asked for the hash value $H(M^*)$.

$\mathcal{B}$ looks into the list L to find an entry $(M^*, b_{M^*}, \alpha_{M^*})$, which necessarily exists since $H(M^*)$ was asked by $\mathcal{A}$. Then, $\mathcal{B}$ fails if $b_{M^*} = 0$ (since, in this case, $H(M^*) = g^{\alpha_{M^*}}$, which does not depend on g^b). Otherwise, we have $H(M^*) = (g^b)^{\alpha_{M^*}}$ and $\mathcal{B}$ can compute $g^{ab} = \sigma^{*1/\alpha_{M^*}}$ (note that α_{M^*} is invertible modulo p and p is public, so that $\mathcal{B}$ can compute $\alpha_{M^*}^{-}1 \mod p$ with the extended euclidean algorithm).

Success Probability of $\mathcal{B}$: We denote by $M_1, ..., M_q$ the messages for which $\mathcal{A}$ obtains signatures.

$$\begin{aligned}\Pr[\mathcal{B} \text{ does not fail}] =& \Pr[b_{M^*} = 1] \cdot \Pr[\bigwedge_{i=1..q} b_{M_i} = 0] \\ =& \delta \cdot (1-\delta)^q\end{aligned}$$

This is optimal for $\delta = \frac{1}{q+1}$, and we obtain

$$\begin{aligned}\Pr[B \text{ does not fail}] =& \frac{1}{q+1} \cdot (1 - \frac{1}{q+1})^q \\ \approx& \frac{1}{\exp(1)(q+1)} \text{ for large values of } q.\end{aligned}$$

Finally, $\Pr[\mathcal{B} \text{ succeeds}] = \Pr[\mathcal{B} \text{ does not fail}] \cdot \Pr[\mathcal{A} \text{ succeeds}]$. So if $\mathcal{A}$ has advantage ε, then $\mathcal{B}$ solves CDH with advantage $\frac{\varepsilon}{\exp(1)(q+1)}$, where $\exp(1)$ is the base for the natural logarithm. Since q is polynomial in λ, the latter advantage is non-negligible whenever ε is non-negligible. □

1.2 The Boneh-Franklin IBE ([2])

1.2.1 Description

Setup(λ)**:** choose groups (G, G_T) of prime order $p > 2^\lambda$ with a bilinear map $e : G \times G \to G_T$ and a generator $g \stackrel{R}{\leftarrow} G$. Choose a hash function $H : \{0,1\}^* \to G$. Choose $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_p$. Define $MSK = \alpha$ and

$$MPK = \{(G, G_T),\ g,\ g_1 = g^\alpha,\ H\}.$$

Keygen(MSK, ID)**:** return the private key $d_{ID} = H(ID)^\alpha$.

Encrypt(MPK, ID, M)**:** To encrypt $M \in G_T$ for the identity ID, choose $r \stackrel{R}{\leftarrow} \mathbb{Z}_p$ and compute

$$C = (C_1, C_2) = (g^r, M \cdot e(g_1, H(ID))^r).$$

Decrypt(MPK, d_{ID}, C)**:** Compute $M = C_2 / e(C_1, d_{ID})$.

1.2.2 Security

Theorem 2. *The Boneh-Franklin IBE is IND-ID-CPA secure in the random oracle model if the DBDH assumption holds in* (G, G_T).

Proof. Let $\mathcal{A}$ be an attacker against the BF IBE with advantage ε. We build a DBDH distinguisher $\mathcal{B}$ with advantage $\frac{\varepsilon}{\exp(1)\cdot(q+1)}$, where q is the number of private key queries made by $\mathcal{A}$.

Algorithm $\mathcal{B}$ takes as input (g, g^a, g^b, g^c, T) where either $T = e(g,g)^{abc}$ or $T \in_R G_T$ and $\mathcal{B}$ has to decide which is the case. To this end, $\mathcal{B}$ defines MPK so that $g_1 = g^a$ (implicitly, $MSK = a$) and answers $\mathcal{A}$'s queries as follows.

Hash queries: when $\mathcal{A}$ asks for the hash value $H(ID)$,

- $\mathcal{B}$ returns the previously defined $H(ID)$ if it exists.

- Otherwise, $\mathcal{B}$ flips a coin $b_{ID} \in \{0,1\}$ such that $\Pr[b_{ID} = 1] = \frac{1}{q+1}$ and $\Pr[b_{ID} = 0] = \frac{q}{q+1}$.

    * if $b_{ID} = 0$, $\mathcal{B}$ defines $H(ID) = g^{\beta_{ID}}$ where $\beta_{ID} \overset{R}{\leftarrow} \mathbb{Z}_p$.
    * if $b_M = 1$, $\mathcal{B}$ defines $H(ID) = (g^b)^{\beta_{ID}}$ where $\beta_{ID} \overset{R}{\leftarrow} \mathbb{Z}_p$.

  In both cases, $\mathcal{B}$ stores (ID, b_{ID}, β_{ID}) in a list L (initially empty).

Private key queries: when $\mathcal{A}$ queries the private key d_{ID} of an identity ID, $\mathcal{B}$ responds as follows. Again, we assume w.l.o.g. that every private key query for an identity ID is preceded by a hash query for the same identity. Hence, $\mathcal{B}$ can recover the entry (ID, b_{ID}, β_{ID}) in the list L.

- If $b_{ID} = 1$, then $\mathcal{B}$ fails and outputs a random bit $\beta \overset{R}{\leftarrow} \{0,1\}$.
- If $b_{ID} = 0$, $\mathcal{B}$ can compute $d_{ID} = H(ID)^a = (g^{\beta_{ID}})^a = (g^a)^{\beta_{ID}}$.

Challenge: $\mathcal{A}$ chooses $M_0, M_1 \in G_T$ and an identity ID^* that has never been queried for private key extraction.
We assume w.l.o.g. that $H(ID^*)$ was queried by $\mathcal{A}$ since, otherwise, $\mathcal{B}$ can make the hash query $H(ID^*)$ for itself. So, $\mathcal{B}$ can recover $(ID^*, b_{ID^*}, \beta_{ID^*})$ from the list L.

- If $b_{ID^*} = 0$, $\mathcal{B}$ fails and outputs a random bit $\beta \overset{R}{\leftarrow} \{0,1\}$.
- If $b_{ID^*} = 1$, $\mathcal{B}$ chooses $\gamma \overset{R}{\leftarrow} \{0,1\}$ and computes the challenge ciphertext as
$$C^* = (C_1, C_2) = (g^c, M_\gamma \cdot T^{\beta_{ID^*}}),$$
  where $\gamma \in_R \{0,1\}$ is chosen at random.

If $T = e(g,g)^{abc}$, then C^* can be written

$$\begin{aligned} C^* =& (g^c, M_\gamma \cdot e(g^a, (g^b)^{\beta_{ID^*}})^c) \\ =& (g^c, M_\gamma \cdot e(g_1, H(ID^*))^c), \end{aligned}$$

which means that it is a valid encryption of M_γ. Otherwise, if $T \in_R G_T$, then C^* can be written as $C^* = (g^c, M_{rand} \cdot e(g_1, H(ID^*))^c)$ for some random message $M_{rand} \in_R G_T$. This is because we can write $T = e(g,g)^{abc+z}$, for some uniformly random $z \in_R \mathbb{Z}_p$ that does not appear anywhere else during the game. This means that

$$C^* = (g^c, M_\gamma \cdot e(g,g)^{z \cdot \beta_{ID^*}} \cdot e(g_1, H(ID^*))^c).$$

Hence, C^* is distributed as an encryption of $M_{rand} = M_\gamma \cdot e(g,g)^{z \cdot \beta_{ID^*}}$, which is uniformly distributed in G_T. Since $z \in_R \mathbb{Z}_p$ is independent of $\mathcal{A}$'s view, M_γ is perfectly hidden by the factor $e(g,g)^{z \cdot \beta_{ID^*}}$ and C^* contains no information on the bit $\gamma \in_R \{0,1\}$.

Output: A outputs $\gamma' \in \{0,1\}$. If $\gamma = \gamma'$, $\mathcal{B}$ outputs 1 (meaning that $T = e(g,g)^{abc}$). If $\gamma \neq \gamma'$, $\mathcal{B}$ outputs 0 (meaning that $T \in_R G_T$).

Success Probability: As in the case of BLS signatures, if we call `Fail` the event that $\mathcal{B}$ fails, we have

$$\begin{aligned}\Pr[\neg\texttt{Fail}] =& \frac{1}{q+1} \cdot (1 - \frac{1}{q+1})^q \\ \approx& \frac{1}{\exp(1) \cdot (q+1)} \text{ for large values of } q.\end{aligned}$$

Then, we have

$$\begin{aligned}
&\Pr[\mathcal{B} = 1 | T = e(g,g)^{abc}] \\
&\quad = \frac{\Pr[\mathcal{B} = 1 \wedge \neg\texttt{Fail} \wedge T = e(g,g)^{abc}]}{\Pr[T = e(g,g)^{abc}]} \cdot \frac{\Pr[\neg\texttt{Fail} \wedge T = e(g,g)^{abc}]}{\Pr[\neg\texttt{Fail} \wedge T = e(g,g)^{abc}]} \\
&\qquad + \frac{\Pr[\mathcal{B} = 1 \wedge \texttt{Fail} \wedge T = e(g,g)^{abc}]}{\Pr[T = e(g,g)^{abc}]} \cdot \frac{\Pr[\texttt{Fail} \wedge T = e(g,g)^{abc}]}{\Pr[\texttt{Fail} \wedge T = e(g,g)^{abc}]} \\
&\quad = \Pr[\mathcal{B} = 1 | T = e(g,g)^{abc} \wedge \neg\texttt{Fail}] \cdot \Pr[\neg\texttt{Fail} | T = e(g,g)^{abc}] \\
&\qquad + \Pr[\mathcal{B} = 1 | T = e(g,g)^{abc} \wedge \texttt{Fail}] \cdot \Pr[\texttt{Fail} | T = e(g,g)^{abc}] \\
&\quad = \Pr[\mathcal{B} = 1 | T = e(g,g)^{abc} \wedge \neg\texttt{Fail}] \cdot \Pr[\neg\texttt{Fail}] \\
&\qquad + \Pr[\mathcal{B} = 1 | T = e(g,g)^{abc} \wedge \texttt{Fail}] \cdot \Pr[\texttt{Fail}] \\
&\quad = \Pr[\mathcal{B} = 1 | T = e(g,g)^{abc} \wedge \neg\texttt{Fail}] \cdot \Pr[\neg\texttt{Fail}] + \frac{1}{2} \cdot \Pr[\texttt{Fail}] \\
&\quad = \frac{1}{2} + \Pr[\neg\texttt{Fail}] \cdot (\Pr[\mathcal{B} = 1 | T = e(g,g)^{abc} \wedge \neg\texttt{Fail}] - \frac{1}{2}),
\end{aligned}$$

where the 4-th equality is due to the fact that $\mathcal{B}$ outputs 1 with probability 1/2 when `Fail` occurs. In the case $T \in_R G_T$, we similarly find

$$\begin{aligned}\Pr[\mathcal{B} = 1 | T \in_R G_T] &= \frac{1}{2} + \Pr[\neg\texttt{Fail}] \cdot (\Pr[\mathcal{B} = 1 | T \in_R G_T \wedge \neg\texttt{Fail}] - \frac{1}{2}) \\ &= \frac{1}{2}\end{aligned}$$

since $\Pr[\gamma' = \gamma | T \in_R G_T \wedge \neg\texttt{Fail}] = 1/2$. Moreover, conditionally on $T = e(g,g)^{abc} \wedge \neg\texttt{Fail}$, we know that $\mathcal{A}$'s view is the same as in the real game, so that we have $\varepsilon = \mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{B} = 1 | T = e(g,g)^{abc} \wedge \neg\texttt{Fail}] - 1/2|$. If $T \in_R G_T$, we have

$$\Pr[\mathcal{B} = 1 | T \in_R G_T \wedge \neg\texttt{Fail}] = \frac{1}{2}.$$

So, we finally obtain

$$\begin{aligned} \mathbf{Adv}_{\mathcal{B}}^{\mathrm{DBDH}}(\lambda) &= |\Pr[\mathcal{B} = 1 | T = e(g,g)^{abc}] - \Pr[\mathcal{B} = 1 | T \in_R G_T]| \\ &= \varepsilon \cdot \Pr[\neg\texttt{Fail}] = \frac{\varepsilon}{\exp(1) \cdot (q+1)}. \end{aligned}$$

□

We remark that the security proof uses the property that $M \in G_T$ to argue that M_γ is perfectly hidden when $T \in_R G_T$. Since G_T is usually a subgroup of the multiplicative group of some finite field, it is crucial to encode M as an element of G_T (rather than an arbitrary finite field element) for the same reasons as in the ElGamal encryption scheme.

2 IBE in the standard model

There exist examples of cryptographic schemes which have a security proof in the random oracle model but are insecure in any instantiation with a real hash function H. So, we prefer having security proofs in the standard model when it is possible, although cryptosystems in the random oracle model tend to be more efficient.

2.1 Selective Security

As a first towards secure IBE schemes in the standard model, we will consider an example of IBE scheme with a security proof (in the standard model) in the sense of a weaker security definition.

Definition 1. *An IBE is secure against selective-ID attacks (IND-sID-CPA) if no PPT adversary has non negligible advantage in the following game.*

0. *The adversary $\mathcal{A}$ chooses a target identity ID^*.*
1. *The challenger generates $(MPK, MSK) \leftarrow Setup(\lambda)$ and gives MPK to $\mathcal{A}$.*

2. *$\mathcal{A}$ makes private key queries for $ID \neq ID^*$ (polynomially many times). At each query, the challenger returns $d_{ID} \leftarrow Keygen(MSK, ID)$.*

3. *$\mathcal{A}$ chooses messages M_0, M_1 and obtains $C^* \leftarrow Encrypt(MPK, M_\gamma, ID^*)$ with $\gamma \xleftarrow{R} \{0,1\}$.*

4. *$\mathcal{A}$ makes further private key queries (polynomially many times) for identities $ID \neq ID^*$.*

5. *$\mathcal{A}$ outputs $\gamma' \in \{0,1\}$ and wins if $\gamma' = \gamma$.*

The advantage of $\mathcal{A}$ is defined as $\mathbf{Adv}_{\mathcal{A}}(\lambda) = |\Pr[\gamma' = \gamma] - 1/2|$.

This security notion is strictly weaker than IND-ID-CPA because the target identity $ID*$ must be chosen at the beginning of the algorithm, before the generation of the master key pair (MPK, MSK). In for some applications, this security notion will be sufficient.

2.2 The Boneh-Boyen IBE (Eurocrypt '04, see [3])

Setup(λ) :

1. Choose groups (G, G_T) of prime order $p > 2^\lambda$ with a bilinear map $e : G \times G \to G_T$ and generators g, g_2, $h \xleftarrow{R} G$.

2. Choose $\alpha \xleftarrow{R} \mathbb{Z}_p$ and compute $g_1 = g^\alpha \in G$.

3. Define $MSK := g_2^\alpha \in G$ and $MPK := \{(G, G_T), g, g_1 = g^\alpha, g_2, h\}$.

Keygen(MSK, ID): We are given $MSK = g_2^\alpha$ and $ID \in \mathbb{Z}_p$. We suppose that $ID \in \mathbb{Z}_p$. If it is not the case it suffices to hash ID using a collision-resistant hash function from $\{0,1\}^*$ (if $ID \in \{0,1\}^*$) to $\mathbb{Z}_p$. The private key is computed as

$$d_{ID} = (d_1, d_2) = (g_2^\alpha \cdot H_G(ID)^r, g^r)$$

using a random $r \xleftarrow{R} \mathbb{Z}_p$ and where $H_G(ID) = g_1^{ID} \cdot h$ (note that $H_G : \mathbb{Z}_p \to G$ is a number theoretic hash function which is collision-resistant).

We have the following equalities which will be useful for the encryption and decryption algorithms:

$$\begin{aligned}e(d_1,g) =& e(g_2^{\alpha}\cdot H_G(ID)^r, g)\\ =& e(g_2^{\alpha}, g)\cdot e(H_G(ID)^r, g)\\ =& e(g_2, g^{\alpha})\cdot e(H_G(ID), g^r)\\ =& e(g_1,g_2)\cdot e(H_G(ID), d_2).\end{aligned}$$

Encrypt(MPK**,** M**,** ID**):** given $M \in G_T$ and $ID \in \mathbb{Z}_p$, choose $s \overset{R}{\leftarrow} \mathbb{Z}_p$ and compute

$$C = (C_1, C_2, C_3) = (g^s, H_G(ID)^s, M \cdot e(g_1, g_2)^s).$$

Decrypt(MPK**,** C**,** d_{ID}**):** given $d_{ID} = (d_1, d_2)$ compute

$$M = C_3 \cdot \frac{e(C_2, D_2)}{e(C_1, d_1)}.$$

Correctness : We know that private keys (d_1, d_2) satisfy

$$e(d_1, g) = e(g_1, g_2) \cdot e(H_G(ID), d_2).$$

By raising both members of this equality to the power $s \in \mathbb{Z}_p$, we obtain

$$\begin{aligned}e(d_1, g^s) &= e(g_1, g_2)^s \cdot e(H_G(ID)^s, d_2),\\ \text{i.e. } e(d_1, C_1) &= e(g_1, g_2)^s \cdot e(C_2, d_2).\end{aligned}$$

Correctness follows from this last equality.

2.2.1 Security

Theorem 3. *The previous scheme is IND-sID-CPA secure in the standard model if the DBDH assumption holds in* (G, G_T)*.*

Proof. Let $\mathcal{A}$ be an IND-sID-CPA adversary for the Boneh-Boyen IBE with non negligible advantage ε. We construct a DBDH distinguisher B with advantage ε.

Algorithm $\mathcal{B}$ takes as input (g, g^a, g^b, g^c, T) where either $T = e(g,g)^{abc}$ or $T \in_R G_T$.

Init: $\mathcal{A}$ chooses ID^* as a target identity.

Setup: $\mathcal{B}$ defines MPK with

$$g_1 = g^a$$
$$g_2 = g^b$$
$$h = (g^a)^{-ID^*} \cdot g^\omega$$

with $\omega \xleftarrow{R} \mathbb{Z}_p$ so that h is uniformly distributed in G. The adversary is run on input of

$$MPK = \{(G, G_T), g, g_1, g_2, h\}$$

an MSK is implicitly defined as $g_2^a = g^{ab}$.

Private key queries: for any identity $ID \neq ID^*$, $\mathcal{B}$ picks $r \xleftarrow{R} \mathbb{Z}_p$ and computes

$$\begin{aligned} d_{ID} &= (d_1, d_2) \\ &= (H_G(ID)^r \cdot (g^b)^{-\omega/(ID-ID^*)}, g^r \cdot (g^b)^{-1/(ID-ID^*)}) \end{aligned}$$

Letting $\tilde{r} = r - b/(ID - ID^*)$, we have

$$\begin{aligned} d_1 &= H_G(ID)^r \cdot (g^b)^{-\omega/(ID-ID^*)} \\ &= H_G(ID)^{\tilde{r}+b/(ID-ID^*)} \cdot (g^b)^{-\omega/(ID-ID^*)} \\ &= H_G(ID)^{\tilde{r}} \cdot ((g^a)^{ID-ID^*} \cdot g^\omega)^{b/(ID-ID^*)} \cdot (g^b)^{-\omega/(ID-ID^*)} \\ &= H_G(ID)^{\tilde{r}} \cdot g^{ab} \end{aligned}$$

and $d_2 = g^r \cdot (g^b)^{-1/(ID-ID^*)} = g^{\tilde{r}}$. So, the obtained $d_{ID} = (g^{ab} \cdot H_G(ID)^{\tilde{r}}, g^{\tilde{r}})$ has the same distribution as outputs of the real Keygen algorithm.

Challenge: $\mathcal{A}$ chooses $M_0, M_1 \in G_T$. Then, $\mathcal{B}$ chooses a random bit $\gamma \xleftarrow{R} \{0,1\}$ and computes the challenge ciphertext as

$$C^* = (C_1, C_2, C_3) = (g^c, (g^c)^\omega, M_\gamma \cdot T).$$

We know that

$$\begin{aligned} H_G(ID^*)^c &= (g_1^{ID^*} \cdot h)^c \\ &= (g_1^{ID^*} \cdot g_1^{-ID^*} \cdot g^\omega)^c \\ &= (g^\omega)^c, \end{aligned}$$

so that $C_2 = H_G(ID^*)^c$.

- If $T = e(g,g)^{abc}$, then

$$\begin{aligned} C^* =& (C_1, C_2, C_3) \\ =& (g^c, H_G(ID^*)^c, M_\gamma \cdot e(g^a, g^b)^c) \\ =& (g^c, H_G(ID^*)^c, M_\gamma \cdot e(g_1, g_2)^c) \end{aligned}$$

  The distribution of C^* is the same as in a valid encryption of M_γ.

- If $T \in_R G_T$, then we can write $T = e(g,g)^{abc+z}$ for some uniformly random $z \in_R \mathbb{Z}_p$. Therefore we can write

$$C^* = (g^c, H_G(ID^*)^c, M_{rand} \cdot e(g_1, g_2)^c),$$

  where $M_{rand} = M_\gamma \cdot e(g,g)^z$. In this case, the factor $e(g,g)^z$ perfectly hides M_γ since z is random and independent of $\mathcal{A}$'s view. This means that C^* does not reveal any information on $\gamma \in \{0,1\}$.

Output: A outputs $\gamma' \in \{0,1\}$. If $\gamma' = \gamma$, B returns 1 (meaning that $T = e(g,g)^{abc}$). Otherwise, B returns 0 (meaning that $T \in_R G_T$). The same arguments as in the security proof of ElGamal show that $\mathcal{B}$'s advantage as a DBDH distinguisher is identical to $\mathcal{A}$'s advantage as a selective-ID adversary:

$$\mathbf{Adv}_{\mathcal{B}}^{\mathrm{DBDH}}(\lambda) \quad = \quad |\Pr[\mathcal{B} = 1 | T = e(g,g)^{abc}] - \Pr[\mathcal{B} = 1 | T \in_R G_T]| = \varepsilon.$$

□

2.2.2 Full Security (Waters, Eurocrypt'05, see [4])

Idea: The function $H_G(ID) = g_1^{ID} h$ is replaced by a different identity-hashing algorithm

$$H_G(ID) = u_0 \cdot \prod_{i=1}^{L} u_i^{ID[i]},$$

where $ID[i]$ is the i-th bit of the identity $ID \in \{0,1\}^L$, which is represented as a L-bit string, for some $L \in \mathsf{poly}(\lambda)$, and $\{u_i\}_{i \in \{0,\ldots,L\}}$ is a sequence of elements of G, contained in MPK.

Waters [4] proved that, with this choice of the H_G function, the Boneh-Boyen IBE scheme is upgraded to achieve full security (IND-ID-CPA) in the standard model.

Remarks

- The notion of IND-ID-CPA security is *strictly* stronger than that of IND-sID-CPA security when the universe of identities has exponential size in the security parameter λ. If the identity space $\{0,1\}^L$ is sufficiently small (for example, when $L \approx \log \lambda$), then IND-sID-CPA security implies IND-ID-CPA security under a polynomial reduction which consists in guessing the target identity ID^* beforehand. When the number of possible identites is exponential (as is the case in most applications of IBE), the latter reduction is not polynomial since ID^* cannot be guessed with non-negligible probability. In [4], Waters gives a proof of IND-ID-CPA security with a polynomial reduction when L is polynomial in λ.

- It is possible to show (see [2]) that any IND-ID-CPA secure IBE scheme generically implies a signature scheme that provides security under chosen-message attacks. The key pair of the signature scheme is the master key pair (MPK, MSK) of the IBE system and a message M is signed by deriving a private key d_M for the identity M. Verification is achieved by IBE-encrypting a random plaintext under the identity M and checking if the signature d_M allows recevoring the encrypted plaintext. In most cases, the signature verification algorithm can be re-written as a deterministic algorithm (as in the Boneh-Franklin IBE, which implies BLS signatures).

References

[1] Boneh, D., Lynn, B., & Shacham, H. (2001). Short signatures from the Weil pairing. In Advances in *Cryptology—ASIACRYPT 2001* (pp. 514-532). Springer Berlin Heidelberg.

[2] Boneh, D., & Franklin, M. (2001, January). Identity-based encryption from the Weil pairing. In Advances in *Cryptology—CRYPTO 2001* (pp. 213-229). Springer Berlin Heidelberg.

[3] Boneh, D., & Boyen, X. (2004, January). Efficient selective-ID secure identity-based encryption without random oracles. In *Advances in Cryptology-EUROCRYPT 2004* (pp. 223-238). Springer Berlin Heidelberg.

[4] Waters, B. (2005). Efficient identity-based encryption without random oracles. In *Advances in Cryptology–EUROCRYPT 2005* (pp. 114-127). Springer Berlin Heidelberg.