Security Proofs using the Game-based Methodology

David Pointcheval

Ecole normale supérieure, CNRS & INRIA

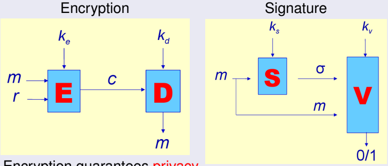
Scuola Superiore di Catania
Catania – Italy
April 21st, 2009

Outline

Outline

Public-Key Cryptography

Asymmetric cryptography



- Encryption guarantees privacy
- Signature guarantees authentication,
  and even non-repudiation by the sender

Strong Security Notions

Signature

Existential Unforgeability under Chosen-Message Attacks

An adversary, allowed to ask for signature on any message of its choice, cannot generate a new valid message-signature pair

Encryption

Semantic Security against Chosen-Ciphertext Attacks

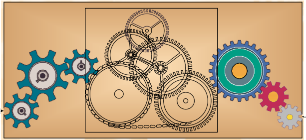
An adversary that chooses 2 messages, and receives the encryption of one of them, is not able to guess which message has been encrypted, even if it is able to ask for decryption of any ciphertext of its choice (except the challenge ciphertext)

Provable Security

One can prove that:

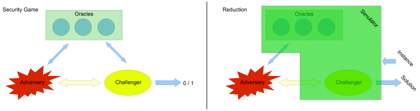
- if an adversary is able to break the cryptographic scheme
- then one can break the underlying problem (integer factoring, discrete logarithm, 3-SAT, etc)

hard instance →



→ solution

Direct Reduction



Unfortunately

- Security may rely on several assumptions
- Proving that the view of the adversary, generated by the simulator, in the reduction is the same as in the real attack game is not easy to do in such a one big step

Game-based Methodology

Illustration: OAEP [Bellare-Rogaway EC '94]

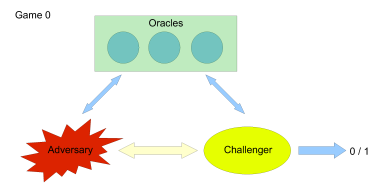
- Reduction proven indistinguishable for an IND-CCA adversary (actually IND-CCA1, and not IND-CCA2) but widely believed for IND-CCA2, without any further analysis of the reduction
  The direct-reduction methodology
- [Shoup - Crypto '01]
  Shoup showed the gap for IND-CCA2, under the OWP
  Granted his new game-based methodology
- [Fujisaki-Okamoto-Pointcheval-Stern – Crypto '01]
  FOPS proved the security for IND-CCA2, under the PD-OWP
  Using the game-based methodology

Outline

Sequence of Games

Real Attack Game

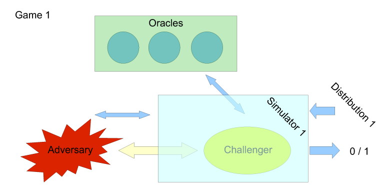
The adversary plays a game, against a challenger (security notion)



Sequence of Games

Simulation

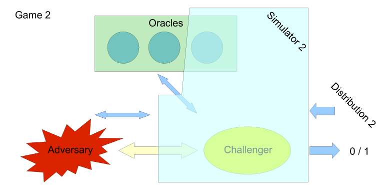
The adversary plays a game, against a sequence of simulators



Sequence of Games

Simulation

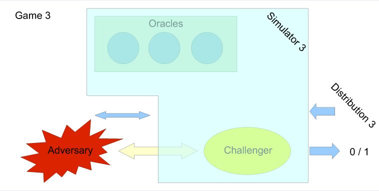
The adversary plays a game, against a sequence of simulators



Sequence of Games

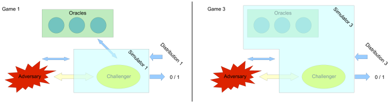
Simulation

The adversary plays a game, against a sequence of simulators

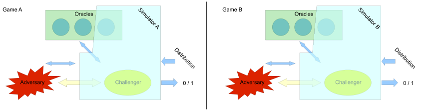


Output

- The output of the simulator in Game 1 is related to the output of the challenger in Game 0 (adversary's winning probability)
- The output of the simulator in Game 3 is easy to evaluate (e.g. always zero, probability of one-half)
- The gaps (Game 1 $\leftrightarrow$ Game 2, Game 2 $\leftrightarrow$ Game 3, etc) are clearly identified with specific events

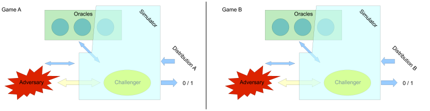


Two Simulators



- perfectly identical behaviors **[Hop-S-Perfect]**
- different behaviors, only if event **Ev** happens
  - **Ev** is negligible **[Hop-S-Negl]**
  - **Ev** is non-negligible **[Hop-S-Non-Negl]**
    and independent of the output in **Game**$_A$
    $\rightarrow$ Simulator B terminates in case of event **Ev**

Two Distributions



- perfectly identical input distributions **[Hop-D-Perfect]**
- different distributions
  - statistically close **[Hop-D-Stat]**
  - computationally close **[Hop-D-Comp]**

Two Simulations

- Identical behaviors: $\Pr[\textbf{Game}_A] - \Pr[\textbf{Game}_B] = 0$
- The behaviors differ only if **Ev** happens:
  - **Ev** is negligible, one can ignore it
    Shoup's Lemma: $\Pr[\textbf{Game}_A] - \Pr[\textbf{Game}_B] \leq \Pr[\textbf{Ev}]$

$$\begin{aligned}
&|\Pr[\textbf{Game}_A] - \Pr[\textbf{Game}_B]| \\
&= \left| \begin{array}{l} \Pr[\textbf{Game}_A|\textbf{Ev}]\Pr[\textbf{Ev}] + \Pr[\textbf{Game}_A|\neg\textbf{Ev}]\Pr[\neg\textbf{Ev}] \\ -\Pr[\textbf{Game}_B|\textbf{Ev}]\Pr[\textbf{Ev}] - \Pr[\textbf{Game}_B|\neg\textbf{Ev}]\Pr[\neg\textbf{Ev}] \end{array} \right| \\
&= \left| \begin{array}{l} (\Pr[\textbf{Game}_A|\textbf{Ev}] - \Pr[\textbf{Game}_B|\textbf{Ev}]) \times \Pr[\textbf{Ev}] \\ +(\Pr[\textbf{Game}_A|\neg\textbf{Ev}] - \Pr[\textbf{Game}_B|\neg\textbf{Ev}]) \times \Pr[\neg\textbf{Ev}] \end{array} \right| \\
&\leq |1 \times \Pr[\textbf{Ev}] + 0 \times \Pr[\neg\textbf{Ev}]| \leq \Pr[\textbf{Ev}]
\end{aligned}$$

  - **Ev** is non-negligible and independent of the output in $\textbf{Game}_A$,
    Simulator B terminates, in case of event **Ev**

Two Simulations

- Identical behaviors: $\Pr[\textbf{Game}_A] - \Pr[\textbf{Game}_B] = 0$
- The behaviors differ only if **Ev** happens:
  - **Ev** is negligible, one can ignore it
  - **Ev** is non-negligible and independent of the output in $\textbf{Game}_A$,
    Simulator B terminates and outputs 0, in case of event **Ev**:

$$\begin{aligned}
\Pr[\textbf{Game}_B] &= \Pr[\textbf{Game}_B|\textbf{Ev}]\Pr[\textbf{Ev}] + \Pr[\textbf{Game}_B|\neg\textbf{Ev}]\Pr[\neg\textbf{Ev}] \\
&= 0 \times \Pr[\textbf{Ev}] + \Pr[\textbf{Game}_A|\neg\textbf{Ev}] \times \Pr[\neg\textbf{Ev}] \\
&= \Pr[\textbf{Game}_A] \times \Pr[\neg\textbf{Ev}]
\end{aligned}$$

Simulator B terminates and flips a coin, in case of event **Ev**:

$$\begin{aligned}
\Pr[\textbf{Game}_B] &= \Pr[\textbf{Game}_B|\textbf{Ev}]\Pr[\textbf{Ev}] + \Pr[\textbf{Game}_B|\neg\textbf{Ev}]\Pr[\neg\textbf{Ev}] \\
&= \tfrac{1}{2} \times \Pr[\textbf{Ev}] + \Pr[\textbf{Game}_A|\neg\textbf{Ev}] \times \Pr[\neg\textbf{Ev}] \\
&= \tfrac{1}{2} + (\Pr[\textbf{Game}_A] - \tfrac{1}{2}) \times \Pr[\neg\textbf{Ev}]
\end{aligned}$$

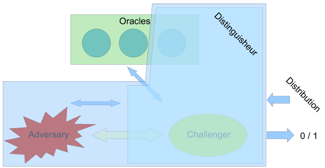
Two Simulations

- Identical behaviors: $\Pr[\textbf{Game}_A] - \Pr[\textbf{Game}_B] = 0$
- The behaviors differ only if **Ev** happens:
  - **Ev** is negligible, one can ignore it
  - **Ev** is non-negligible and independent of the output in $\textbf{Game}_A$,
    Simulator B terminates in case of event **Ev**

Event Ev

- Either **Ev** is negligible, or the output is independent of **Ev**
- For being able to terminate simulation B in case of event **Ev**, this event must be *efficiently* detectable
- For evaluating $\Pr[\textbf{Ev}]$, one re-iterates the above process, with an initial game that outputs 1 when event **Ev** happens

Two Distributions



$$\Pr[\textbf{Game}_A] - \Pr[\textbf{Game}_B] \leq \mathbf{Adv}(\mathcal{D}^{\text{oracles}})$$

Two Distributions

$$\Pr[\mathbf{Game}_A] - \Pr[\mathbf{Game}_B] \leq \mathbf{Adv}(\mathcal{D}^{\text{oracles}})$$

- For identical/statistically close distributions, for any oracle:

$$\Pr[\mathbf{Game}_A] - \Pr[\mathbf{Game}_B] = \mathbf{Dist}(\mathbf{Distrib}_A, \mathbf{Distrib}_B) = \text{negl}()$$

- For computationally close distributions, in general, we need to exclude additional oracle access:

$$\Pr[\mathbf{Game}_A] - \Pr[\mathbf{Game}_B] \leq \mathbf{Adv}^{\mathbf{Distrib}}(t)$$

where t is the computational time of the distinguisheur

Outline

Gap Groups

Definition (Pairing Setting)

- Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of prime order p
- Let g_1 and g_2 be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively
- Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}^T$, be a bilinear map

Definition (Admissible Bilinear Map)

Let $(p, \mathbb{G}_1, g_1, \mathbb{G}_2, g_2, \mathbb{G}^T, e)$ be a pairing setting, with $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}^T$ a non-degenerated bilinear map

- Bilinear: for any $g \in \mathbb{G}_1$, $h \in \mathbb{G}_2$ and $u, v \in \mathbb{Z}$,

$$e(g^u, h^v) = e(g, h)^{uv}$$

- Non-degenerated: $e(g_1, g_2) \neq 1$

Bilinear Diffie-Hellman Problems

We focus on the symmetric case: $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$

Diffie-Hellman Problems

- **CDH** in $\mathbb{G}$: Given $g, g^a, g^b \in \mathbb{G}$, compute g^{ab}
- **DDH** in $\mathbb{G}$: Given $g, g^a, g^b, g^c \in \mathbb{G}$, decide whether $c = ab$ or not

CDH can be hard to solve, but **DDH** is easy in gap-groups

Bilinear Diffie-Hellman Problems

- **CBDH** in $\mathbb{G}$: Given $g, g^a, g^b, g^c \in \mathbb{G}$, compute $e(g, g)^{abc}$
- **DBDH** in $\mathbb{G}$: Given $g, g^a, g^b, g^c \in \mathbb{G}$ and $h \in \mathbb{G}^T$, decide whether $h \stackrel{?}{=} e(g, g)^{abc}$

Outline

Signature in Gap Groups

[Boneh-Lynn-Shacham – Asiacryp '01]

Let $\mathbb{G}$ be a cyclic group of prime order p, with a generator g
Assumption: $\mathbb{G}$ gap-group (**DDH** easy, whereas **CDH** intractable)

Signature Scheme

- Key generation: choose $x \in \mathbb{Z}_p$, and set $y = g^x$;
- Signature of $M \in \mathbb{G}$: $\sigma = M^x$;
- Verification of (M, σ): check $\mathbf{DDH}(g, y, M, \sigma)$

Full-Domain Hash

$$\mathcal{H} : \{0,1\}^* \to \mathbb{G}$$

- In order to sign m, one first computes $M = \mathcal{H}(m) \in \mathbb{G}$
- then $\sigma = M^x = \mathbf{CDH}(g, y, \mathcal{H}(m))$

EUF-CMA Security

EUF-CMA

Existential Unforgeability under Chosen-Message Attacks
An adversary, allowed to ask for signature on any message of its choice, cannot generate a new valid message-signature pair

Theorem

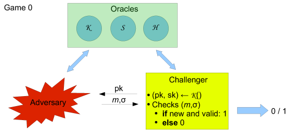
*The BLS signature achieves EUF-CMA security, under the **CDH** assumption in $\mathbb{G}$, in the Random Oracle Model:*

$$\mathbf{Adv}^{\mathsf{euf-cma}}(t) \le q_H \times \mathbf{Adv}^{\mathbf{cdh}}(t + q_H \tau_e)$$

Assumptions:

- any signing query has been first asked to $\mathcal{H}$
- the forgery has been asked to $\mathcal{H}$

Real Attack Game



Random Oracle

$\mathcal{H}(m)$: $M \overset{R}{\leftarrow} \mathbb{G}$, output M

Key Generation Oracle

$\mathcal{K}()$: sk $\overset{R}{\leftarrow} \mathbb{Z}_p$, pk $= g^{\mathsf{sk}}$

Signing Oracle

$\mathcal{S}(m)$: $M = \mathcal{H}(m)$, output $\sigma = M^{\mathsf{sk}}$

Simulations

- **Game**$_0$: use of the oracles $\mathcal{K}$, $\mathcal{S}$ and $\mathcal{H}$
- **Game**$_1$: use of the *simulation of the Random Oracle*

Simulation of $\mathcal{H}$

$\mathcal{H}(m)$: $\mu \stackrel{R}{\leftarrow} \mathbb{Z}_p$, output $M = g^\mu$

$\Longrightarrow$ **Hop-D-Perfect**: $\Pr[\mathbf{Game}_1] = \Pr[\mathbf{Game}_0]$

- **Game**$_2$: use of the *simulation of the Signing Oracle*

Simulation of $\mathcal{S}$

$\mathcal{S}(m)$: find μ such that $M = \mathcal{H}(m) = g^\mu$, output $\sigma = \mathsf{pk}^\mu$

$\Longrightarrow$ **Hop-S-Perfect**: $\Pr[\mathbf{Game}_2] = \Pr[\mathbf{Game}_1]$

$\mathcal{H}$-Query Selection

- **Game**$_3$: random index $t \stackrel{R}{\leftarrow} \{1, \ldots, q_H\}$

Event Ev

If the t-th query to $\mathcal{H}$ is not the output forgery

We terminate the game and output 1 if **Ev** happens
$\Longrightarrow$ **Hop-S-Non-Negl**
Then, clearly

$$\Pr[\mathbf{Game}_3] = \Pr[\mathbf{Game}_2] \times \Pr[\neg\mathbf{Ev}] \qquad \Pr[\mathbf{Ev}] = 1 - 1/q_H$$

$$\Pr[\mathbf{Game}_3] = \Pr[\mathbf{Game}_2] \times \frac{1}{q_H}$$

CDH Instance

- **Game**$_4$: **CDH** instance $(g, A = g^a, B = g^b)$
  Use of the *simulation of the Key Generation Oracle*

Simulation of $\mathcal{K}$

$\mathcal{K}()$: set $\mathsf{pk} \leftarrow A$

Modification of the *simulation of the Random Oracle*

Simulation of $\mathcal{H}$

If this is the t-th query, $\mathcal{H}(m)$: $M \leftarrow \mathcal{B}$, output M

The unique difference is for the t-th simulation of the random oracle, for which we cannot compute a signature.
But since it corresponds to the forgery output, it cannot be queried to the signing oracle:
$\Longrightarrow$ **Hop-S-Perfect**: $\Pr[\mathbf{Game}_4] = \Pr[\mathbf{Game}_3]$

Conclusion

In **Game**$_4$, when the output is 1, $\sigma = \mathbf{CDH}(g, A = g^a, B = g^b)$ and the simulator computes one exponentiation per hashing:

$$\begin{array}{rcl}
\Pr[\mathbf{Game}_4] & \leq & \mathbf{Adv}^{\mathsf{cdh}}(t + q_H \tau_e) \\
\Pr[\mathbf{Game}_4] & = & \Pr[\mathbf{Game}_3] \\
\Pr[\mathbf{Game}_3] & = & \Pr[\mathbf{Game}_2] \times \dfrac{1}{q_H} \\
\Pr[\mathbf{Game}_2] & = & \Pr[\mathbf{Game}_1] \\
\Pr[\mathbf{Game}_1] & = & \Pr[\mathbf{Game}_0] \\
\Pr[\mathbf{Game}_0] & = & \mathbf{Adv}^{\mathsf{euf-cma}}(\mathcal{A})
\end{array}$$

$$\mathbf{Adv}^{\mathsf{euf-cma}}(\mathcal{A}) \leq q_H \times \mathbf{Adv}^{\mathsf{cdh}}(t + q_H \tau_e)$$

Outline

Identity-Based Cryptography

[Shamir – Crypto '84]

Public-Key Cryptography

Each user $\mathcal{ID}$ owns

- a public key pk
- a certificate that guarantees the link between $\mathcal{ID}$ and pk
- a private key sk, related to pk

One has to access a dictionary in order to get pk, the public key of $\mathcal{ID}$, together with the certificate, in order to encrypt a message to $\mathcal{ID}$

Identity-Based Cryptography

Each user $\mathcal{ID}$ owns

- a private key sk, related to $\mathcal{ID}$
- the public key pk is indeed $\mathcal{ID}$ itself

Identity-Based Encryption

Setup

The authority generates a master secret key msk,
and publishes the public parameters, PK

Extraction

Given an identity $\mathcal{ID}$, the authority computes
the private key sk granted the master secret key msk

Encryption

Any one can encrypt a message m to a user $\mathcal{ID}$
using only m, $\mathcal{ID}$ and the public parameters PK

Decryption

Given a ciphertext, user $\mathcal{ID}$ can recover the plaintext, with sk

Security Model: $\mathbf{IND-ID-CCA}$

Definition ($\mathbf{IND-ID-CCA}$ Security)

- $\mathcal{A}$ receives the global parameters
- $\mathcal{A}$ asks any extraction-query, and any decryption-query
- $\mathcal{A}$ outputs a target identity $\mathcal{ID}^\star$ and two messages (m_0, m_1)

The challenger flips a bit b, and encrypts m_b for $\mathcal{ID}^\star$ into $c^\star$

- $\mathcal{A}$ asks any extraction-query, and any decryption-query
- $\mathcal{A}$ outputs its guess b' for b

Restriction: $\mathcal{ID}^\star$ never asked to the extraction oracle,
and $(\mathcal{ID}^\star, c^\star)$ never asked to the decryption oracle.

CPA: no decryption-oracle access

$$\mathbf{Adv}^{\mathsf{ind-id-cca}} = 2 \times \Pr[b' = b] - 1$$

Identity-Based Encryption

[Boneh-Franklin – Crypto '01]

Setup

- The authority sets up a gap-group framework:
  a group $\mathbb{G}$ of prime order p, with a generator g,
  equipped with an admissible bilinear map
  $$e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}^T$$
- It selects a master secret key $\mathsf{msk} = s \in \mathbb{Z}_p$
- It publishes the public parameters: $\mathsf{PK} = (p, \mathbb{G}, e, g, P = g^s)$

Extraction

Given an identity $\mathcal{ID}$, the authority computes
the private key $\mathsf{sk} = \mathcal{H}(\mathcal{ID})^s$

Note that sk is a BLS signature of $\mathcal{ID}$: $e(\mathsf{sk}, g) = e(\mathcal{H}(\mathcal{ID}), P)$

BF IBE (Cont'd)

Encryption

In order to encrypt a message m to a user $\mathcal{ID}$

- one chooses a random $r \in \mathbb{Z}_p$
- computes $A = g^r$ and $K = e(P, \mathcal{H}(\mathcal{ID})^r)$
- sends $(A, B = K \times m)$

$$\begin{aligned} K &= e(P, \mathcal{H}(\mathcal{ID})^r) = e(g^s, \mathcal{H}(\mathcal{ID})^r) \\ &= e(g^r, \mathcal{H}(\mathcal{ID})^s) = e(A, \mathsf{sk}) \end{aligned}$$

Decryption

Upon reception of (A, B), user $\mathcal{ID}$

- computes $K = e(A, \mathsf{sk})$
- gets $m = B/K$

BF IBE Security Analysis

Theorem

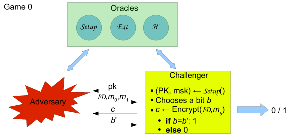
The BF IBE is $\mathbf{IND-ID-CPA}$ *secure*
*under the **DBDH** problem, in the random oracle model*

By masking m with H(K): $B = m \oplus H(K)$,
the BF IBE is $\mathbf{IND-ID-CPA}$ *secure*
*under the **CBDH** problem, in the random oracle model*

Theorem

The BLS signature achieves $\mathbf{EUF-CMA}$ *security, under the **CDH** assumption in* $\mathbb{G}$*, in the Random Oracle Model*

Real Attack Game



Random Oracle

$\mathcal{H}(\mathcal{ID})$: $M \stackrel{R}{\leftarrow} \mathbb{G}$, output M

Setup Oracle

$\mathcal{S}etup()$: $\mathsf{msk} \stackrel{R}{\leftarrow} \mathbb{Z}_p$, $P = g^{\mathsf{msk}}$

Extraction Oracle

$\mathcal{E}xt(\mathcal{ID})$: $M = \mathcal{H}(\mathcal{ID})$, output $\mathsf{sk} = M^{\mathsf{msk}}$

Simulations

- **Game**$_0$: use of the oracles $\mathcal{S}etup$, $\mathcal{E}xt$, and $\mathcal{H}$
- **Game**$_1$: use of the *simulation of the Random Oracle*

Simulation of $\mathcal{H}$

$\mathcal{H}(\mathcal{ID})$: $\mu \stackrel{R}{\leftarrow} \mathbb{Z}_p$, output $M = g^\mu$

$\Longrightarrow$ **Hop-D-Perfect**: Pr[**Game**$_1$] = Pr[**Game**$_0$]

- **Game**$_2$: use of the *simulation of the Extraction Oracle*

Simulation of $\mathcal{E}xt$

$\mathcal{E}xt(\mathcal{ID})$: find μ such that $M = \mathcal{H}(\mathcal{ID}) = g^\mu$, output sk $= P^\mu$

$\Longrightarrow$ **Hop-S-Perfect**: Pr[**Game**$_2$] = Pr[**Game**$_1$]

$\mathcal{H}$-Query Selection

- **Game**$_3$: random index $t \stackrel{R}{\leftarrow} \{1, \ldots, q_H\}$

Event Ev

If the t-th query to $\mathcal{H}$ is not the challence $\mathcal{ID}$

We terminate the game and flip a coin if **Ev** happens
$\Longrightarrow$ **Hop-S-Non-Negl**

$$\Pr[\textbf{Game}_3] = \frac{1}{2} + \left(\Pr[\textbf{Game}_2] - \frac{1}{2}\right) \times \Pr[\neg\textbf{Ev}] \quad \Pr[\textbf{Ev}] = 1 - 1/q_H$$

$$\Pr[\textbf{Game}_3] = \frac{1}{2} + \left(\Pr[\textbf{Game}_2] - \frac{1}{2}\right) \times \frac{1}{q_H}$$

Challenge $\mathcal{ID}$

- **Game**$_4$: True **DBDH** instance $(g, g^\alpha, g^\beta, g^\gamma)$ with $h = e(g,g)^{\alpha\beta\gamma}$
  Use of the *simulation of the Setup Oracle*

Simulation of $\mathcal{S}etup$

$\mathcal{S}etup()$: set $P \leftarrow g^\alpha$

Modification of the *simulation of the Random Oracle*

Simulation of $\mathcal{H}$

If this is the t-th query, $\mathcal{H}(\mathcal{ID})$: $M \leftarrow g^\beta$, output M

Difference for the t-th simulation of the random oracle: we cannot extract the secret key. Since this is the challenge $\mathcal{ID}$, it cannot be queried to the extraction oracle:
$\Longrightarrow$ **Hop-D-Perfect**: Pr[**Game**$_4$] = Pr[**Game**$_3$]

Challenge Ciphertext

- **Game**$_5$: True **DBDH** instance $(g, g^\alpha, g^\beta, g^\gamma)$ with $h = e(g,g)^{\alpha\beta\gamma}$
  We have set $P \leftarrow g^\alpha$, and for the t-th query to $\mathcal{H}$: $M = g^\beta$

Ciphertext

Set $A \leftarrow g^\gamma$ and $K \leftarrow h$ to generate the encryption of m_b under $\mathcal{ID}$

$\Longrightarrow$ **Hop-D-Perfect**: Pr[**Game**$_5$] = Pr[**Game**$_4$]

- **Game**$_6$: Random **DBDH** instance $(g, g^\alpha, g^\beta, g^\gamma)$ with $h \stackrel{R}{\leftarrow} \mathbb{G}^T$
  $\Longrightarrow$ **Hop-D-Comp**:

$$|\Pr[\textbf{Game}_6] - \Pr[\textbf{Game}_5]| \leq \mathbf{Adv}^{\mathbf{dbdh}}(t + q_H\tau_e)$$

Conclusion

In this last **Game**$_6$, it is clear that $\Pr[\mathbf{Game}_6] = \frac{1}{2}$

$$
\begin{aligned}
|\Pr[\mathbf{Game}_6] - \Pr[\mathbf{Game}_5]| &\leq \mathbf{Adv}^{\mathbf{dbdh}}(t + q_H\tau_e)\\
\Pr[\mathbf{Game}_5] &= \Pr[\mathbf{Game}_4]\\
\Pr[\mathbf{Game}_4] &= \Pr[\mathbf{Game}_3]\\
\Pr[\mathbf{Game}_3] &= \frac{1}{2} + (\Pr[\mathbf{Game}_2] - \frac{1}{2}) \times \frac{1}{q_H}\\
\Pr[\mathbf{Game}_2] &= \Pr[\mathbf{Game}_1]\\
\Pr[\mathbf{Game}_1] &= \Pr[\mathbf{Game}_0]\\
\Pr[\mathbf{Game}_0] &= \frac{1}{2} + \mathbf{Adv}^{\mathsf{ind-id-cpa}}(\mathcal{A})
\end{aligned}
$$

$$
\mathbf{Adv}^{\mathsf{ind-id-cpa}}(\mathcal{A}) \leq q_H \times \mathbf{Adv}^{\mathbf{dbdh}}(t + q_H\tau_e)
$$

Outline

Conclusion

- The game-based methodology uses a sequence of games
- The transition hops
  - are simple
  - easy to check

It leads to easy-to-read and easy-to-verify security proofs:

- Some mistakes have been found granted this methodology [Analysis of OAEP]
- Some security analyses became possible to handle [Analysis of EKE]

This approach can be automized [CryptoVerif]