Security Proofs using the Game-based Methodology

David Pointcheval
Ecole normale supérieure, CNRS & INRIA
Scuola Superiore di Catania
Catania – Italy
April 21st, 2009

Outline

1 Cryptography
   - Introduction
   - Provable Security
2 Game-based Methodology
   - Game-based Approach
   - Transition Hops
3 Assumptions
4 Short Signatures
   - Description of BLS
   - Security Proof
5 Identity-Based Encryption
   - Definition
   - Description of BF
   - Security Proof
6 Conclusion

Cryptography

Public-Key Cryptography

Asymmetric cryptography

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

Introduction

Encryption

Signature

m
k_e
E
D
m
k_d

m
S
V

k_s

σ

0/1

r

c

David Pointcheval – 1/47

David Pointcheval – 2/47

David Pointcheval – 3/47

David Pointcheval – 4/47
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
(interger factoring, discrete logarithm, 3-SAT, etc)

Game-based Methodology

Illustration: OAEP
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 showed the gap for IND-CCA2, under the OWP
Granted his new game-based methodology

FOPS proved the security for IND-CCA2, under the PD-OWP
Using the game-based methodology
Outline

1. Cryptography
   - Introduction
   - Provable Security
2. Game-based Methodology
   - Game-based Approach
   - Transition Hops
3. Assumptions
4. Short Signatures
   - Description of BLS
   - Security Proof
5. Identity-Based Encryption
   - Definition
   - Description of BF
   - Security Proof
6. Conclusion

Game-based Approach

Sequence of Games

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

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

- **Game 3**
  - Adversary
  - Challenger
  - Oracles
  - Simulator 3
  - Distribution 3
  - 0 / 1

**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 ↔ Game 2, Game 2 ↔ Game 3, etc) are clearly identified with specific events

**Transition Hops**
- **Two Simulators**
  - perfectly identical behaviors
  - different behaviors, only if event \( Ev \) happens
    - \( Ev \) is negligible
    - \( Ev \) is non-negligible and independent of the output in \( Game_A \)
      → Simulator B terminates in case of event \( Ev \)

- **Two Distributions**
  - perfectly identical input distributions
  - different distributions
    - statistically close
    - computationally close
### Two Simulations

- Identical behaviors: \( \Pr[\text{Game}_A] - \Pr[\text{Game}_B] = 0 \)

  - The behaviors differ only if \( \text{Ev} \) happens:
    - \( \text{Ev} \) is negligible, one can ignore it
    - \( \text{Ev} \) is non-negligible and independent of the output in \( \text{Game}_A \), Simulator B terminates in case of event \( \text{Ev} \)

  \[
  |\Pr[\text{Game}_A] - \Pr[\text{Game}_B]| = |\Pr[\text{Game}_A|\text{Ev}] \Pr[\text{Ev}] + \Pr[\text{Game}_A|\neg\text{Ev}] \Pr[\neg\text{Ev}] - \Pr[\text{Game}_B|\text{Ev}] \Pr[\text{Ev}] - \Pr[\text{Game}_B|\neg\text{Ev}] \Pr[\neg\text{Ev}]| \\
  = (\Pr[\text{Game}_A|\text{Ev}] - \Pr[\text{Game}_B|\text{Ev}]) \times \Pr[\text{Ev}] + (\Pr[\text{Game}_A|\neg\text{Ev}] - \Pr[\text{Game}_B|\neg\text{Ev}]) \times \Pr[\neg\text{Ev}] \\
  \leq |1 \times \Pr[\text{Ev}] + 0 \times \Pr[\neg\text{Ev}]| \leq \Pr[\text{Ev}] \\
  
  - \( \text{Ev} \) is non-negligible and independent of the output in \( \text{Game}_A \), Simulator B terminates, in case of event \( \text{Ev} \)

### Event \( \text{Ev} \)

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

### Two Distributions

- Identical behaviors: \( \Pr[\text{Game}_A] - \Pr[\text{Game}_B] = 0 \)

  - The behaviors differ only if \( \text{Ev} \) happens:
    - \( \text{Ev} \) is negligible, one can ignore it
    - \( \text{Ev} \) is non-negligible and independent of the output in \( \text{Game}_A \), Simulator B terminates and outputs 0, in case of event \( \text{Ev} \):

      \[
      \Pr[\text{Game}_B] = \Pr[\text{Game}_B|\text{Ev}] \Pr[\text{Ev}] + \Pr[\text{Game}_B|\neg\text{Ev}] \Pr[\neg\text{Ev}] \\
      = 0 \times \Pr[\text{Ev}] + \Pr[\text{Game}_A|\neg\text{Ev}] \times \Pr[\neg\text{Ev}] \\
      = \Pr[\text{Game}_A] \times \Pr[\neg\text{Ev}]
      
      \text{Simulator B terminates and flips a coin, in case of event \( \text{Ev} \)}:

      \[
      \Pr[\text{Game}_B] = \Pr[\text{Game}_B|\text{Ev}] \Pr[\text{Ev}] + \Pr[\text{Game}_B|\neg\text{Ev}] \Pr[\neg\text{Ev}] \\
      = \frac{1}{2} \times \Pr[\text{Ev}] + \Pr[\text{Game}_A|\neg\text{Ev}] \times \Pr[\neg\text{Ev}] \\
      = \frac{1}{2} + (\Pr[\text{Game}_A] - \frac{1}{2}) \times \Pr[\neg\text{Ev}]
      
      \Pr[\text{Game}_A] - \Pr[\text{Game}_B] \leq \text{Adv}(D_{\text{oracles}})
Transition Hops

Two Distributions

\[ \Pr[\text{Game}_A] - \Pr[\text{Game}_B] \leq \text{Adv}(D_{\text{oracles}}) \]

- For identical/statistically close distributions, for any oracle:
  \[ \Pr[\text{Game}_A] - \Pr[\text{Game}_B] = \text{Dist}(\text{Distrib}_A, \text{Distrib}_B) = \text{negl}() \]

- For computationally close distributions, in general, we need to exclude additional oracle access:
  \[ \Pr[\text{Game}_A] - \Pr[\text{Game}_B] \leq \text{Adv}(\text{Distrib}(t)) \]
  where \( t \) is the computational time of the distinguisher

Outline

1. Cryptography
   - Introduction
   - Provable Security

2. Game-based Methodology
   - Game-based Approach
   - Transition Hops

3. Assumptions
   - Short Signatures
   - Identity-Based Encryption
     - Definition
     - Description of BF
     - Security Proof

4. Conclusion

Bilinear Maps

Gap Groups

**Definition (Pairing Setting)**
- Let \( G_1 \) and \( G_2 \) be two cyclic groups of prime order \( p \)
- Let \( g_1 \) and \( g_2 \) be generators of \( G_1 \) and \( G_2 \) respectively
- Let \( e : G_1 \times G_2 \rightarrow G_T \), be a bilinear map

**Bilinear Diffie-Hellman Problems**

We focus on the symmetric case: \( G_1 = G_2 = G \)

**Diffie-Hellman Problems**

- \( \text{CDH in } G: \text{ Given } g, g^a, g^b \in G, \text{ compute } g^{ab} \)
- \( \text{DDH in } G: \text{ Given } g, g^a, g^b, g^c \in G, \text{ decide whether } c = ab \text{ or not} \)

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

**Bilinear Diffie-Hellman Problems**

- \( \text{CBDH in } G: \text{ Given } g, g^a, g^b, g^c \in G, \text{ compute } e(g, g)^{abc} \)
- \( \text{DBDH in } G: \text{ Given } g, g^a, g^b, g^c \in G \text{ and } h \in G_T, \text{ decide whether } h \overset{?}{=} e(g, g)^{abc} \)
### Signature in Gap Groups

Let $G$ be a cyclic group of prime order $p$, with a generator $g$.

**Assumption:** $G$ gap-group (DDH easy, whereas CDH intractable)

#### Signature Scheme

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

#### Full-Domain Hash

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

In order to sign $m$, one first computes $M = H(m) \in G$

then $\sigma = M^x = \text{CDH}(g,y,H(m))$

### EUF-CMA Security

**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 $G$, in the Random Oracle Model:

$$\text{Adv}_{\text{euf-cma}}^\text{euf-cma}(t) \leq q_H \times \text{Adv}_{\text{cdh}}(t + q_H T_e)$$

**Assumptions:**

- any signing query has been first asked to $H$
- the forgery has been asked to $H$
Security Proof

Simulations

- **Game 0**: use of the oracles $K$, $S$ and $H$
- **Game 1**: use of the simulation of the Random Oracle

Simulation of $H$

$H(m): \mu \xleftarrow{\$} \mathbb{Z}_p$, output $M = g^\mu$

$\implies$ **Hop-D-Perfect**: $\Pr[Game_1] = \Pr[Game_0]$

- **Game 2**: use of the simulation of the Signing Oracle

Simulation of $S$

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

$\implies$ **Hop-S-Perfect**: $\Pr[Game_2] = \Pr[Game_1]$

$H$-Query Selection

- **Game 3**: random index $t \xleftarrow{\$} \{1, \ldots, q_H\}$

Event $Ev$

If the $t$-th query to $H$ is not the output forgery

$\implies$ **Hop-S-Non-Negl**

Then, clearly

$\Pr[Game_3] = \Pr[Game_2] \times \Pr[\neg Ev]$

Pr[Ev] = 1 - 1/q_H

$\Pr[Game_3] = \Pr[Game_2] \times \frac{1}{q_H}$

CDH Instance

- **Game 4**: CDH instance $(g, A = g^a, B = g^b)$

Use of the simulation of the Key Generation Oracle

Simulation of $K$

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

Modification of the simulation of the Random Oracle

Simulation of $H$

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

$\implies$ **Hop-S-Perfect**: $\Pr[Game_4] = \Pr[Game_3]$

Conclusion

In **Game 4**, when the output is 1, $\sigma = \text{CDH}(g, A = g^a, B = g^b)$ and the simulator computes one exponentiation per hashing:

$\Pr[Game_4] \leq \text{Adv}^{\text{cdh}}(t + q_H \tau_e)$

$\Pr[Game_4] = \Pr[Game_3]$

$\Pr[Game_3] = \Pr[Game_2] \times \frac{1}{q_H}$

$\Pr[Game_2] = \Pr[Game_1]$

$\Pr[Game_1] = \Pr[Game_0]$

$\Pr[Game_0] = \text{Adv}^{\text{euf-\text{cma}}}(A)$

$\text{Adv}^{\text{euf-\text{cma}}}(A) \leq q_H \times \text{Adv}^{\text{cdh}}(t + q_H \tau_e)$
Identity-Based Cryptography

Public-Key Cryptography

Each user $ID$ owns
- a public key $pk$
- a certificate that guarantees the link between $ID$ and $pk$
- a private key $sk$, related to $pk$

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

Identity-Based Cryptography

Each user $ID$ owns
- a private key $sk$, related to $ID$
- the public key $pk$ is indeed $ID$ itself

Security Model: IND − ID − CCA

Definition (IND − ID − CCA Security)

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

The challenger flips a bit $b$, and encrypts $m_b$ for $ID^*$ into $c^*$
- $A$ asks any extraction-query, and any decryption-query
- $A$ outputs its guess $b'$ for $b$

Restriction: $ID^*$ never asked to the extraction oracle, and $(ID^*, c^*)$ never asked to the decryption oracle.

CPA: no decryption-oracle access

$$Adv^{\text{ind−id−cca}} = 2 \times \Pr[b' = b] - 1$$
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 $\text{msk} = s \in \mathbb{Z}_p$.
- It publishes the public parameters: $\text{PK} = (p, \mathbb{G}, e, g, P = g^s)$.

Extraction

- Given an identity $ID$, the authority computes the private key $\text{sk} = H(ID)^s$.
- Note that $\text{sk}$ is a BLS signature of $ID$: $e(\text{sk}, g) = e(H(ID), P)$.

BF IBE (Cont’d)

Encryption

- In order to encrypt a message $m$ to a user $ID$:
  - one chooses a random $r \in \mathbb{Z}_p$.
  - computes $A = g^r$ and $K = e(P, H(ID)^r)$.
  - sends $(A, B = K \times m)$.

Decryption

- Upon reception of $(A, B)$, user $ID$
  - computes $K = e(A, \text{sk})$.
  - gets $m = B / K$.

BF IBE Security Analysis

Theorem

- The BF IBE is 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 IND – ID – CPA secure under the CBDH problem, in the random oracle model.

Theorem

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

- **Game**\(_0\)**: use of the oracles Setup, Ext, and \( \mathcal{H} \)
- **Game**\(_1\)**: use of the simulation of the Random Oracle

**Simulation of \( \mathcal{H} \)**

\[ \mathcal{H}(ID): \mu \overset{R}{\leftarrow} \mathbb{Z}_p, \text{ output } M = g^\mu \]

\( \implies \) **Hop-D-Perfect**: \( \Pr[\text{Game}_1] = \Pr[\text{Game}_0] \)
- **Game**\(_2\)**: use of the simulation of the Extraction Oracle

**Simulation of Ext**

\[ \text{Ext}(ID): \text{find } \mu \text{ such that } M = \mathcal{H}(ID) = g^\mu, \text{ output sk } = P^\mu \]

\( \implies \) **Hop-S-Perfect**: \( \Pr[\text{Game}_2] = \Pr[\text{Game}_1] \)

**Challenge \( ID \)**

- **Game**\(_3\)**: 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 Setup**

\[ \text{Setup}(): \text{ set } P \leftarrow g^\alpha \]

Modification of the simulation of the Random Oracle

**Simulation of \( \mathcal{H} \)**

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

Difference for the \( t \)-th simulation of the random oracle: we cannot extract the secret key. Since this is the challenge \( ID \), it cannot be queried to the extraction oracle:

\( \implies \) **Hop-D-Perfect**: \( \Pr[\text{Game}_4] = \Pr[\text{Game}_3] \)

**H-Query Selection**

- **Game**\(_3\)**: random index \( t \overset{R}{\leftarrow} \{1, \ldots, q_H\} \)

**Event \( \text{Ev} \)**

If the \( t \)-th query to \( \mathcal{H} \) is not the challenge \( ID \)

We terminate the game and flip a coin if \( \text{Ev} \) happens

\( \implies \) **Hop-S-Non-Negl**

\[ \Pr[\text{Game}_3] = \frac{1}{2} + \left( \Pr[\text{Game}_2] - \frac{1}{2} \right) \times \Pr[\neg \text{Ev}] \]

\[ \Pr[\text{Ev}] = 1 - \frac{1}{q_H} \]

**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 \( ID \)

\( \implies \) **Hop-D-Perfect**: \( \Pr[\text{Game}_5] = \Pr[\text{Game}_4] \)
- **Game**\(_6\)**: Random DBDH instance \( (g, g^\alpha, g^\beta, g^\gamma) \) with \( h \overset{R}{\leftarrow} \mathbb{G}_T \)

\( \implies \) **Hop-D-Comp**:

\[ |\Pr[\text{Game}_6] - \Pr[\text{Game}_5]| \leq \text{Adv}\text{dbdh}(t + q_{H^T_e}) \]
In this last Game₆, it is clear that $Pr[\text{Game}_6] = \frac{1}{2}$

$$|Pr[\text{Game}_6] - Pr[\text{Game}_5]| \leq Adv^{\text{dbdh}}(t + q_{H \tau e})$$

$Pr[\text{Game}_5] = Pr[\text{Game}_4]$

$Pr[\text{Game}_4] = Pr[\text{Game}_3]$

$$Pr[\text{Game}_3] = \frac{1}{2} + (Pr[\text{Game}_2] - \frac{1}{2}) \times \frac{1}{q_{H}}$$

$Pr[\text{Game}_2] = Pr[\text{Game}_1]$

$Pr[\text{Game}_1] = Pr[\text{Game}_0]$

$Pr[\text{Game}_0] = \frac{1}{2} + Adv^{\text{ind-id-cpa}}(A)$

$$Adv^{\text{ind-id-cpa}}(A) \leq q_{H} \times Adv^{\text{dbdh}}(t + q_{H \tau e})$$

---

**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]