Verifiable Security of Boneh-Franklin Identity-Based Encryption

Federico Olmedo
Gilles Barthe  Santiago Zanella Béguelin

IMDEA Software Institute, Madrid, Spain

5th International Conference on Provable Security
2011.10.17
Identity-Based Encryption (IBE)

Problem of standard PKE:

*key management is involved and troublesome*
Identity-Based Encryption (IBE)

Problem of standard PKE:

*key management is involved and troublesome*

Proposed solution by Shamir:

*to use recipient's ID as public key*
Identity-Based Encryption (IBE)

Problem of standard **PKE**: 

*key management is involved and troublesome*

Proposed solution by Shamir:

*to use recipient’s ID as public key*
Identity-Based Encryption (IBE)

Problem of standard PKE:

*key management is involved and troublesome*

Proposed solution by Shamir:

*to use recipient's ID as public key*

1. Encrypt with public key
   
   bob@comp.com

Alice  →  Bob
Identity-Based Encryption (IBE)

Problem of standard PKE:

*key management is involved and troublesome*

Proposed solution by Shamir:

*to use recipient’s ID as public key*

1. Encrypt with public key **bob@comp.com**

2. Bob authenticates **bob@comp.com**’s private key
Identity-Based Encryption (IBE)

Problem of standard PKE:

*key management is involved and troublesome*

Proposed solution by Shamir:

*to use recipient’s ID as public key*

1. Encrypt with public key `bob@comp.com`

2. Bob authenticates `bob@comp.com`’s private key

3. "bob@comp.com"’s private key
Should we rely on IBE schemes?

1984: Conception of identity-based cryptography
2001: First practical provably-secure IBE scheme.
2002-2005: Used as building block for many other protocols
2005: Security proof is flawed (but can be patched)
Verifiable security paradigm

Use formal methods to build certified security proofs of cryptographic systems

- Gives strong evidence of correctness of security arguments
- Enables *automation* in proofs
- Demonstrated *applicability* and *effectiveness*
1. The provably-secure BasicIdent scheme
2. CertiCrypt framework
3. Machine-checked proof of BasicIdent security
4. Summary and perspectives
An identity-based encryption scheme is specified by four polynomial algorithms:

- **Setup**
- **Encrypt**
- **Extract**
- **Decrypt**
An identity-based encryption scheme is specified by four polynomial algorithms:

- **Setup**
  - sec. param

- **Encrypt**

- **Extract**

- **Decrypt**
An identity-based encryption scheme is specified by four polynomial algorithms:

- **Setup**
  - Input: secret parameter
  - Output: public parameters

- **Encrypt**
  - Input: public parameters

- **Extract**
  - Input: public parameters

- **Decrypt**
  - Input: public parameters
An *identity-based encryption scheme* is specified by four polynomial algorithms:
An identity-based encryption scheme is specified by four polynomial algorithms:
An identity-based encryption scheme is specified by four polynomial algorithms:

- **Setup**: Takes `sec. param` as input and produces `public params` and `master key`.
- **Encrypt**: Takes `plaintext`, `ID`, and `public params` as inputs and produces `ciphertext`.
- **Extract**: Takes `ID` and `master key` as inputs and produces `secret key`.
- **Decrypt**: Takes `ciphertext`, `secret key`, and `public params` as inputs and produces `plaintext`.
Boneh-Franklin’s recipe

1. Extend the notions of IND-CPA and IND-CCA to IBE schemes
2. Build an IND-CPA-secure IBE scheme BasicIdent
3. Apply a variant of Fujisaki-Okamoto transformation to turn BasicIdent into an IND-CCA-secure IBE scheme
The BasicIdent scheme (definition)

Consider

- $G_1$ and $G_2$, two cyclic groups of prime order $q$,
- $\hat{e}: G_1 \times G_1 \rightarrow G_2$, an efficiently computable bilinear map

\[
\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} \\
\langle P \rangle = G_1 \implies \langle \hat{e}(P, P) \rangle = G_2
\]

- Two hash functions

\[
\mathcal{H}_1 : \{0, 1\}^* \rightarrow G_1^+ \\
\mathcal{H}_2 : G_2 \rightarrow \{0, 1\}^n
\]

The BasicIdent IBE-scheme is defined as

Setup($k$) : $P \leftarrow \$ G_1^+$; $mk \leftarrow \$ \mathbb{Z}_q^+$; $P_{pub} \leftarrow mk \cdot P$; return ($(P, P_{pub}), mk$)

Extract($mk, ID$) : $Q_{ID} \leftarrow \mathcal{H}_1(ID)$; return $mk \cdot Q_{ID}$

Encrypt($ID, m$) : $Q_{ID} \leftarrow \mathcal{H}_1(ID)$; $c \leftarrow \mathbb{Z}_q^+$; $m' \leftarrow \mathcal{H}_2(e(Q_{ID}, P_{pub})^c)$;
return $(c \cdot P, m \oplus m')$

Decrypt($sk, (u, v)$) : return $v \oplus \mathcal{H}_2(\hat{e}(sk, u))$
The BasicIdent scheme (security proof)

- Proof by reduction (in the random oracle model)
  - Define security goal (and adversarial model)
  - Consider a computational assumption
  - Reduce the security of the scheme to the intractability assumption.

\[
\Pr \left[ A \text{ breaks the scheme} \right] \leq \mathcal{F} \left( \Pr \left[ B \text{ solves the hard problem} \right] \right)
\]
Proof by reduction (in the random oracle model)

- Define security goal (and adversarial model)
  - **Indistinguishability under Chosen Plaintext Attack**
    - *Strengthened notion of PKE IND-CPA for IBE*
  - Consider a computational assumption
  - Reduce the security of the scheme to the intractability assumption.

\[
\Pr \left[ A \text{ breaks the scheme} \right] \leq \mathcal{F} \left( \Pr \left[ B \text{ solves the hard problem} \right] \right)
\]
The BasicIdent scheme (security proof)

- **Proof by reduction (in the random oracle model)**
  - Define security goal (and adversarial model)
    - **Indistinguishability under Chosen Plaintext Attack**
      - Strengthened notion of \( PKE \) IND-CPA for \( IBE \)
  - Consider a computational assumption
    - **Bilinear Diffie-Hellman assumption**
      - It is hard to compute \( \hat{\epsilon}(P, P)^{abc} \) given a random tuple \( (P, a \cdot P, b \cdot P, c \cdot P) \).
  - Reduce the security of the scheme to the intractability assumption.

![Problem instance diagram](image)

\[
\Pr \left[ A \text{ breaks the scheme} \right] \leq \mathcal{F} \left( \Pr \left[ B \text{ solves the hard problem} \right] \right)
\]
The BasicIdent scheme (security proof)

- Proof by reduction (in the random oracle model)
  - Define security goal (and adversarial model)
    - Indistinguishability under Chosen Plaintext Attack
      Strengthened notion of PKE IND-CPA for IBE
  - Consider a computational assumption
    - Bilinear Diffie-Hellman assumption
      It is hard to compute \( \hat{e}(P, P)^{abc} \) given a random tuple \((P, a \cdot P, b \cdot P, c \cdot P)\).
  - Reduce the security of the scheme to the intractability assumption.

\[
Pr \left[ \mathcal{A} \text{ breaks the scheme} \right] \leq \mathcal{F} \left( Pr \left[ \mathcal{B} \text{ solves the hard problem} \right] \right)
\]

\[
\text{Adv}_{\text{IND-ID-CPA}}^{\mathcal{A}} \leq \text{Adv}_{\text{BDH}}^{\mathcal{B}} \frac{\exp(1) q_{\mathcal{H}_2} (1 + q \varepsilon \chi)}{2}
\]
The game-playing technique

Security Goal

\[
\text{Game } G_0 \\
\ldots \\
\ldots \leftarrow A(\ ) \\
\ldots \\
\text{Pr}_{G_0} [S_0] \leq f_1(\text{Pr}_{G_1} [S_1]) \leq \ldots \leq f_n(\text{Pr}_{G_n} [S_n])
\]

Reduction

\[
\text{Game } G_1 \\
\ldots \\
\ldots \\
\text{Game } G_n \\
\ldots \\
\ldots \leftarrow B(\ ) \\
\ldots
\]
CertiCrypt: machine-checked crypto proofs

Certified framework for building and verifying crypto proofs in the Coq proof assistant

- Combination of programming language techniques and cryptographic-specific tools
- Game-based methodology, natural to cryptographers
- Several case studies:
  - Encryption schemes: ElGamal, Hashed ElGamal, OAEP
  - Signature schemes: FDH, BLS
  - Zero-Knowledge protocols: Schnorr, Okamoto, Diffie-Hellman, Fiat-Shamir
Inside CertiCrypt (language syntax)

Language-based proofs

Formalize security definitions, assumptions and games using a probabilistic programming language.

pWhile: a probabilistic programming language

\[
C ::= \text{skip} \quad \text{nop}
\]

\[
| C; C \quad \text{sequence}
\]

\[
| V \leftarrow \mathcal{E} \quad \text{assignment}
\]

\[
| V \leftarrow \mathcal{D} \quad \text{random sampling}
\]

\[
| \text{if } \mathcal{E} \text{ then } C \text{ else } C \quad \text{conditional}
\]

\[
| \text{while } \mathcal{E} \text{ do } C \quad \text{while loop}
\]

\[
| V \leftarrow \mathcal{P}(\mathcal{E}, \ldots, \mathcal{E}) \quad \text{procedure call}
\]

- \( x \leftarrow d \): sample the value of \( x \) according to distribution \( d \)

- The language of expressions (\( \mathcal{E} \)) and distribution expressions (\( \mathcal{D} \)) admits user-defined extensions
Observational equivalence

\[\models c_1 \sim^I_0 c_2\]

Example

\[\models x \leftarrow \{0, 1\}^k; y \leftarrow x \oplus z \sim_{\{x, y, z\}} \{0, 1\}^k; x \leftarrow y \oplus z\]

- Useful to relate probabilities

\[
\begin{align*}
\text{fv}(A) &\subseteq O \\
\models c_1 \sim^I_0 c_2 &\quad m_1 \equiv m_2 \\
\Pr[c_1, m_1 : A] &\equiv \Pr[c_2, m_2 : A]
\end{align*}
\]
Fundamental lemma of game-playing

If $G_1$ and $G_2$ are identical up to $\text{bad}$, then

$$|\Pr [G_1, m : A] - \Pr [G_2, m : A]| \leq \max\{\Pr [G_1, m : \text{bad}], \Pr [G_2, m : \text{bad}]\}$$
We extended CertiCrypt with:

- Types and operators for the groups $G_1, G_2$
- An operator for a bilinear map $\hat{e} : G_1 \times G_1 \to G_2$
- Simplification rules for computing normal forms of applications of the bilinear map $\hat{e}$
- An instruction for sampling from Bernoulli distributions
Formalizing the security goal:

\[
\text{Game } G_{\text{IND-ID-CPA}}:
\]

\[
\begin{align*}
(pparams, mk) & \leftarrow \text{Setup}(k); \\
(m_0, m_1, ID_\mathcal{A}) & \leftarrow \mathcal{A}_1(params); \\
b & \leftarrow \{0, 1\}; \\
c & \leftarrow \text{Encrypt}(ID_\mathcal{A}, m_b); \\
b_\mathcal{A} & \leftarrow \mathcal{A}_2(c)
\end{align*}
\]

- The adversary is modeled by two procedures (of unknown code) \( \mathcal{A}_1 \) and \( \mathcal{A}_2 \) that communicate through shared variables.
- \( \mathcal{A}_1 \) and \( \mathcal{A}_2 \) have oracle access to the extraction algorithm and to both random oracles.
- Neither \( \mathcal{A}_1 \) nor \( \mathcal{A}_2 \) is allowed to query the challenge \( ID_\mathcal{A} \) to the extraction oracle.

\[
\text{Adv}^\mathcal{A}_{\text{IND-ID-CPA}} \overset{\text{def}}{=} \left| \Pr_{G_{\text{IND-ID-CPA}}} [b = b_\mathcal{A}] - \frac{1}{2} \right|
\]
Our proof in CertiCrypt

Formalizing the assumptions

- The Bilinear Diffie-Hellman assumption

  Game $G^{B\text{BDH}}$:
  \[
  P \leftarrow \mathbb{G}_1^+; \ a, b, c \leftarrow \mathbb{Z}_q^+; \ z \leftarrow \mathcal{B}(P, a \cdot P, b \cdot P, c \cdot P) \]

  $\text{Adv}_{\text{BDH}}^B \overset{\text{def}}{=} \mathbb{Pr}_{G_{\text{BDH}}^B}[z = \hat{e}(P, P)^{abc}]$

  $\forall \mathcal{B} \cdot \text{PPT}(\mathcal{B}) \implies \text{negl}(\text{Adv}_{\text{BDH}}^B)$

- The random oracle model

  Oracle $\mathcal{H}_1(ID)$:
  
  if $ID \notin \text{dom}(L_1)$ then
  
  $R \leftarrow \mathbb{G}_1^+$;
  
  $L_1(ID) \leftarrow R$
  
  return $L_1(ID)$

  Oracle $\mathcal{H}_2(r)$:
  
  if $r \notin \text{dom}(L_2)$ then
  
  $m \leftarrow \{0, 1\}^n$;
  
  $L_2(r) \leftarrow m$
  
  return $L_2(r)$
Building the reduction...

**Game** $G_{\text{IND-ID-CPA}}$:

- $(\text{parm}, mk) \leftarrow \text{Setup}(k)$;
- $(m_0, m_1, ID_A) \leftarrow \mathcal{A}_1(\text{parm})$;
- $b \leftarrow \{0, 1\}$;
- $c \leftarrow \text{Encrypt}(ID_A, m_b)$;
- $b_A \leftarrow \mathcal{A}_2(c)$

**Game** $G_{\text{BDH}}^B$:

- $P \leftarrow G_1^+$; $a, b, c \leftarrow \mathbb{Z}_q^+$;
- $z \leftarrow B(P, a \cdot P, b \cdot P, c \cdot P)$

\[
\text{Adv}^A_{\text{IND-ID-CPA}} \leq \cdots \leq \text{Adv}^B_{\text{BDH}} \exp(1) q \mathcal{H}_2 (1 + q \varepsilon x) / 2
\]

- Seven intermediate games
- Lazy sampling, fundamental lemma, Coron's technique
- Same bound as Boneh & Franklin proof
Our proof in CertiCrypt

- Our reduction is direct in contrast to Boneh-Franklin proof that goes through an intermediate IND-CPA-secure (non-IBE) encryption scheme
- Used a simpler argument instead of an inductive argument in Boneh-Franklin’s proof that we could not reproduce
- 5000 lines of Coq script
- Built in 3 man-months (but automatically verifiable in 10 minutes)
Contributions

- Presented a machine-checked reduction of the security of the BasicIdent IBE scheme to the Bilinear Diffie-Hellman assumption.
- Demonstrated that CertiCrypt can be extended to deal with complex security proofs of cryptographic schemes.

Perspectives

- Formalize Fujisaki-Okamoto meta-result.
- Eliminate RO assumption on $G_1$: formalize Brier et al work about indifferentiability of hash functions into elliptic curves.
Questions?

Get CertiCrypt (and EasyCrypt) from:
http://certicrypt.gforge.inria.fr
Programs map an initial memory to a distribution of final memories:

$$\llbracket c \in C \rrbracket : \mathcal{M} \rightarrow \mathcal{D}(\mathcal{M})$$

We use Paulin’s measure monad to represent distributions:

$$\mathcal{D}(A) \overset{\text{def}}{=} (A \rightarrow [0,1]) \rightarrow [0,1]$$

For instance

$$\llbracket x \overset{\$}{\leftarrow} \{\text{true}, \text{false}\} \rrbracket \ m = \lambda f \cdot \left( \frac{1}{2} f(m[x/\text{true}]) + \frac{1}{2} f(m[x/\text{false}]) \right)$$

To compute probabilities, just measure the characteristic function of the event:

$$\Pr[c, m : A] \overset{\text{def}}{=} \llbracket c \rrbracket \ m \ 1_A$$
What does it take to trust a proof in CertiCrypt

- **You need to**
  - trust the type checker of Coq
  - trust the definition of the language semantics
  - make sure the security statement and the computational assumption (a few lines in Coq) are what you expect it to be

- **You don’t need to**
  - understand or even read the proof
  - trust proof tactics, program transformations
  - trust program logics, wp-calculus
  - be an expert in Coq
### Our proof in CertiCrypt

**Game CPA:**

- \( L_1, L_2, L_3 \leftarrow \text{nil}; \)
- \( P \leftarrow \mathbb{G}_1^+; \ a \leftarrow \mathbb{Z}_q^+; \)
- \( P_{\text{pub}} \leftarrow aP; \)
- \( (m_0, m_1, ID_A) \leftarrow A_1(P, P_{\text{pub}}); \)
- \( d \leftarrow \{0, 1\}; \)
- \( y \leftarrow E(ID_A, m_d); \)
- \( d_A \leftarrow A_2(y) \)

**Oracle \( \mathcal{E}(ID) \):**

- if \( ID \notin L_3 \) then
  - \( L_3 \leftarrow ID : L_3 \)
  - \( Q \leftarrow \mathcal{H}(ID) \)
  - return \( aQ \)

**Oracle \( \mathcal{H}(ID) \):**

- if \( ID \notin \text{dom}(L_1) \) then
  - \( R \leftarrow \mathbb{G}_1^+; \)
  - \( L_1(id) \leftarrow R \)
  - return \( L_1(ID) \)

**Game BDH:**

- \( P \leftarrow \mathbb{G}_1^+; \ a, b, c \leftarrow \mathbb{Z}_q^+; \)
- \( z \leftarrow B(P, aP, bP, cP) \)
- \( B(P_0, P_1, P_2, P_3) : \)
  - \( L_1, L_2, L_3, V, T \leftarrow \text{nil}; \)
  - while \( |T| < q_{\mathcal{H}_1} \) do
    - \( t \leftarrow t \oplus_p \text{false}; \ T \leftarrow t :: T \)
    - \( P \leftarrow P_0; \ P_{\text{pub}} \leftarrow P_1; \ P' \leftarrow P_2; \)
    - \( (m_0, m_1, ID_A) \leftarrow A_1(P, P_{\text{pub}}); \)
    - \( Q_A \leftarrow \mathcal{H}(ID_A); \ v' \leftarrow V(ID_A)^{-1}; \)
    - \( R \leftarrow \{0, 1\}^n; \ y \leftarrow (v'P_3, R); \)
    - \( d_A \leftarrow A_2(y); \)
    - \( i \leftarrow [1..|L_2|]; \text{ return } \text{fst}(L_2[i]) \)

**Oracle \( \mathcal{E}(ID) \):**

- if \( ID \notin L_3 \) then
  - \( L_3 \leftarrow ID : L_3 \)
  - \( Q \leftarrow \mathcal{H}(ID) \)
  - return \( aQ \)

**Oracle \( \mathcal{H}(ID) \):**

- if \( ID \notin \text{dom}(L_1) \) then
  - \( v \leftarrow \mathbb{Z}_q^+; \)
  - \( V(ID) \leftarrow v; \)
  - if \( T[|L_1|] \) then
    - \( L_1(ID) \leftarrow vP' \)
  - else
    - \( L_1(ID) \leftarrow vP \)
  - return \( L_1(ID) \)

**Oracle \( \mathcal{H}_2(r) \):**

- if \( r \notin \text{dom}(L_2) \) then
  - \( m \leftarrow \{0, 1\}^n; \)
  - \( L_2(r) \leftarrow m \)
  - return \( L_2(r) \)
Semantic security of an IBE scheme

An IBE scheme is IND-ID-CPA-secure iff

$$\forall A \cdot \text{PPT}(A) \implies \left| \Pr[b = b'] - \frac{1}{2} \right| \text{ is negligible}$$
Semantic security of an **IBE** scheme

An **IBE** scheme is *IND-ID-CPA-secure* iff

\[ \forall A \cdot \text{PPT} (A) \implies \left| \Pr [ b = b' ] - \frac{1}{2} \right| \text{ is negligible} \]
An IBE scheme is \textit{IND-ID-CPA-secure} \iff
\[ \forall A \cdot \text{PPT}(A) \implies \left| \Pr [b = b'] - \frac{1}{2} \right| \text{ is negligible} \]
Semantic security of an IBE scheme

An IBE scheme is \textit{IND-ID-CPA-secure} iff

\[ \forall \mathcal{A} \cdot \text{PPT}(\mathcal{A}) \land \Pr \left[ \bigwedge_{i=1}^{m} id_i \neq id_A \right] = 1 \implies \Pr \left[ b = b' \right] - \frac{1}{2} \text{ is negligible} \]