**OAEP 3-Round**
**A Generic and Secure Asymmetric Encryption Padding**

Duong Hieu Phan  
ENS – France

David Pointcheval  
CNRS-ENS – France

Asiacrypt ’04  
Jeju Island - Korea  
December 6th 2004

---

**Asymmetric Encryption**

An asymmetric encryption scheme $\pi = (G, E, D)$ is defined by 3 algorithms:

- **G** – key generation  
  $\omega \rightarrow G \rightarrow (k_e, k_d)$

- **E** – encryption  
  $m \rightarrow r \rightarrow (k_e, k_d) \rightarrow c \rightarrow m$

- **D** – decryption  
  $m \rightarrow (k_e, k_d) \rightarrow c \rightarrow m$

---

**Summary**

- Asymmetric Encryption
- OAEP 3-Round
  - Review
  - Limitations
- New Results
- Conclusion

---

**Security Notion: IND-CCA2**

An IND-CCA2 secure encryption scheme satisfies the following conditions:

1. **IND**
   - $b' \neq b$  
   - $b' \neq c'$

2. **CCA1**
   - $c \neq c'$
   - $m \rightarrow m_0 \rightarrow m_1 \rightarrow m$

3. **CCA2**
   - $c \neq c'$
   - $m \rightarrow m_0 \rightarrow m_1 \rightarrow m$

---
**IND: Probabilistic**

To achieve indistinguishability, a public-key encryption scheme must be probabilistic otherwise, with the challenge \( c = E(m_b) \) one computes \( c_0 = E(m_o) \) and checks whether \( c_0 = c \).

For any plaintext, the number of possible ciphertexts must be lower-bounded by \( 2^k \), for a security level in \( 2^k \):

- at least \( \text{length}(c) \geq \text{length}(m) + k \)

**CCA: Redundancy?**

- For IND-CCA2: redundancy
  - Plaintext-awareness = invalid ciphertexts
- Last year, we proposed:
  - Full-Domain Permutation
  - OAEP 3-Round
  - **IND-CCA2 without redundancy**

**OAEP 3-Round**

- \( E(m) : c = f(t \parallel u) \)
- \( D(c) : t \parallel u = f^{-1}(c) \)

then invert OAEP, and return \( m \)

**Security Result: Asiacrypt ’03**

With a random of size \( k_0 \), but no redundancy

In the ROM, a \((t, \varepsilon)-\text{IND-CCA2}\) adversary helps to partially invert \( f \) within time \( t' \approx t + qD Q / 2^{k_0} \), with success probability \( \geq \varepsilon - qD Q / 2^{k_0} \)

Limitations:
- Requires a trapdoor OW permutation
- Reduction to the partial-domain one-wayness
Intuition

- From the view of the challenge $c^*$
  - OAEP (with redundancy): [Sh01] showed that an adversary could produce a ciphertext $c$, with $r = r^*$
  - [FOPS01] ... but needs to query $H(s^*)$
  - OAEP 2-round (w/t redundancy): we thought that no easy proof could lead to $H(s^*)$ but...
  - OAEP 3-round (w/t redundancy): could prove the requirement of the query $H(t^*)$
    $\Rightarrow$ Partial-Domain OW
- This paper: requirement of both $G(s^*)$ and $H(t^*)$ $\Rightarrow$ Full-Domain OW

New Security Result

With a random of size $k_0$, but no redundancy

In the ROM, a $(t, \epsilon)$-IND-CCA2 adversary helps to invert $f$ within time $t' \approx t + q g_q H^t r^t$

with success probability $\geq \epsilon/2 - 5q_d Q / 2^{k_0}$

where $Q$ is the global number of queries

Simulation of the decryption oracle on $c$:
- look for all the tuples $(s, G(s), t, H(t))$
- check whether $f(t \parallel H(t) \oplus s) = c$
- compute $m = s \oplus F(t \oplus G(s))$ or random

Permutation Requirement

- The permutation requirement rules out many candidates: ElGamal, Paillier, Rabin, NTRU, ...
- Could we apply it to trapdoor one-way probabilistic injections?

\[ f : (x, \rho) \rightarrow y = f(x, \rho) \]
- injection in $x$: at most one $x$ for each $y$
  (but possibly many $\rho$)
- hard to invert
- a trapdoor helps to recover $x$

\[ E(m, r \parallel \rho) = f(t \parallel u, \rho) \]

Problems for the Simulation

- Simulation of the decryption oracle on $c$:
  - look for all the tuples $(s, G(s), t, H(t))$
  - check whether $f(t \parallel H(t) \oplus s) = c$ (existence of $\rho$)
  - compute $m = s \oplus F(t \oplus G(s))$ or random
- Need of a decisional oracle: Same($c$, $c'$)
  - Do $c$ and $c'$ encrypt the same element?
  - Computational problem given access to a decisional oracle $\Rightarrow$ Gap Problem
- And what about $c = f(t' \parallel H(t') \oplus s^*, \rho)$?
  - Same($c$, $c'$) is true, but $m = m^*$ is unknown
Relaxed Chosen-Ciphertext Security

- [ADR02] Generalized CCA:
  - $R$ is a decryption-respecting relation
  - Intuition: R formalizes a trivial relation between ciphertexts encrypting the same plaintext.
  - The adversary is not allowed to ask decryption queries on $c$ in relation with $c^*$
- [CKN03] Replayable CCA:
  - On $c$ which encrypts either $m_0$ or $m_1$: answer = TEST
- Relaxed CCA: $(m, r, \rho) \rightarrow c = E(m, r || \rho)$
  - On $c = E(m^*, r^* || \rho)$: answer = TEST

Relations

- Generalized CCA: is the most natural
  - non-significant bits in the ciphertext cannot be used in the attack.
- Replayable CCA: TEST reveals some information
  - RCCA security $\Rightarrow$ Replayable CCA
    - a RCCA simulator decrypts more often
  - On $c = E(m^*, r^* || \rho)$ $\Rightarrow$ $m$ is $m_b$ and thus either $m_0$ or $m_1$
- If $|\rho| = 0$
  - TEST on $c^*$ only: RCCA = CCA
  - Same is the equality test: no more Gap Problem

Security Result

With a random of size $k_0$, but no redundancy
In the ROM, a $(t, \varepsilon)$-IND-RCCA adversary helps to invert $f$ within time $t' \approx t + qDq_Aq_H (T_f + T_{\text{Same}})$
with success probability $\geq \varepsilon / 2 - 5qDQ / 2^{k_0}$
after less than $qDq_Aq_H$ queries to the Same oracle
- quite loose reduction in general:
  - large security parameters
  - but small overhead: 160 bits of additional randomness

The RSA Case

- The same proof applies to RSA
  - RCCA = CCA
  - Gap-RSA = RSA
  - Proper bookkeeping: better reduction
    - $qDq_Aq_H \rightarrow q_Aq_H$
  - $\Rightarrow$ Cost of the reduction similar to OAEP
    - but relative to the Full-Domain RSA
  - $\Rightarrow$ The most efficient reduction
    - for an RSA-based padding into a $Z^*_n$ element
Conclusion

OAEP 3-Round: the best OAEP-like variant
- the tightest reduction in the RSA case
  - for any exponent
  - relative to the RSA problem
- no redundancy: *almost* optimal bandwidth
- applicable to most of the asymmetric primitives
  - namely ElGamal, relative to the Gap DH