RSA-OAEP and Cramer-Shoup

Olli Ahonen

Laboratory of Physics, TKK

11th Dec 2007

T-79.5502 Advanced Cryptology

Part I: Outline

- RSA, OAEP and RSA-OAEP
- Preliminaries for the proof
- Proof of IND-CCA2 security for RSA-OAEP
  - Setup and process
  - Decryption oracle service
  - Likelihood of success
  - Fujisaki's method
- Safe modulus size

Basic RSA

- Random primes p and q
- Public $N = pq$; private $\Phi(N) = (p - 1)(q - 1)$
- Random public $e \in \mathbb{Z}^*_{\Phi(N)}$
- Private d such that $ed \bmod \Phi(N) = 1$
- Ciphertext $c = m^e \bmod N$
- Decryption: $m = c^d \bmod N$
- IND-CPA (i.e., semantically) secure

Basic RSA: not secure enough

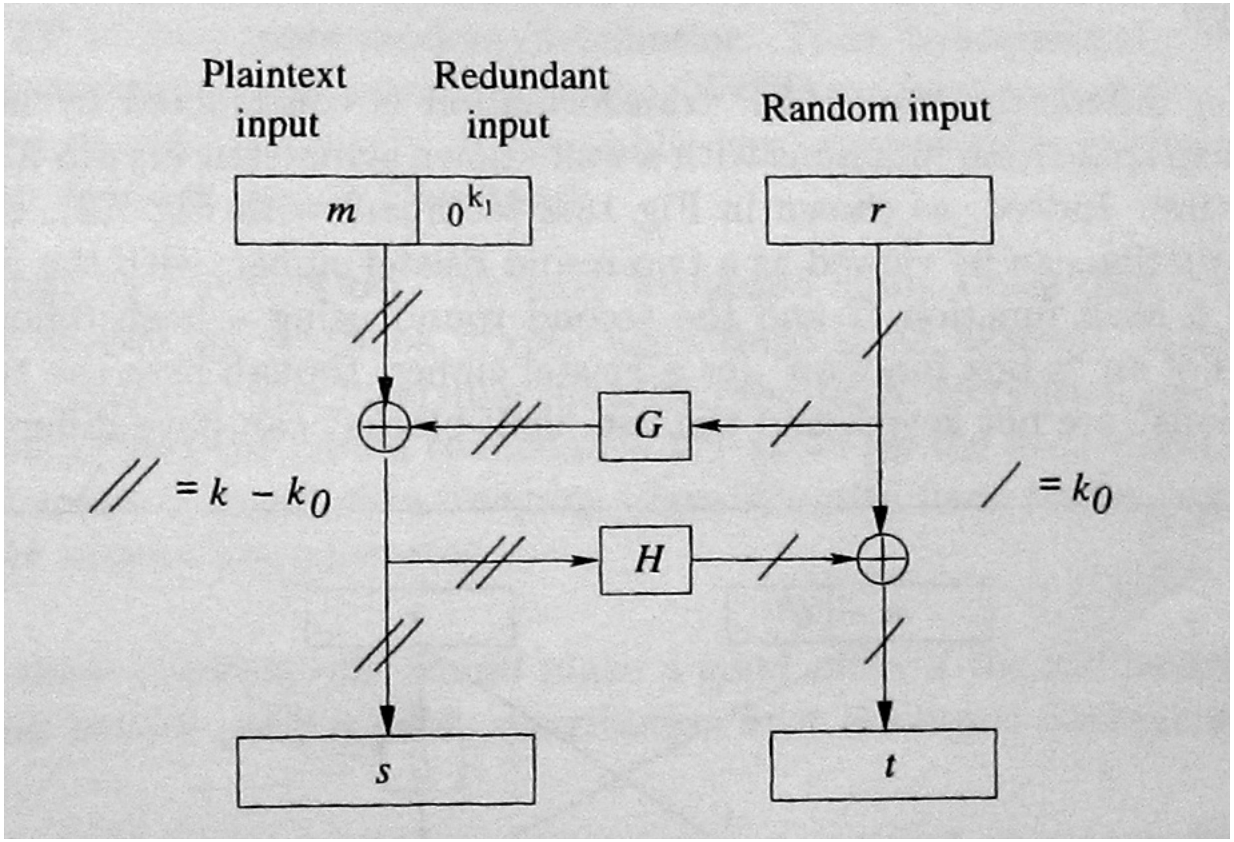
- Assume: Alice acts as a decryption oracle, if the message appears random
- Malice wishes to decrypt $c = m^e \bmod N$
  - Picks random $r \in \mathbb{Z}^*_N$
  - Sends to Alice $c' = r^e c \bmod N$
  - Receives $rm \bmod N$
  - Learns m by division mod N

Optimal asymmetric encryption padding (OAEP)

- M. Bellare and P. Rogaway in 1994
  - Add randomness
  - Mix the input
  - Encrypt with a one-way trapdoor permutation (OWTP), e.g., RSA
- IND-CCA2 secure
  - Assuming the OWTP really is one-way
- Practically efficient

OAEP structure

- $k_0 < |N|/2$
- Hash functions G and H
- $s||t$ input to encryption
- E.g:
  $|N| = 2048$
  $k_0 = k_1 = 160$



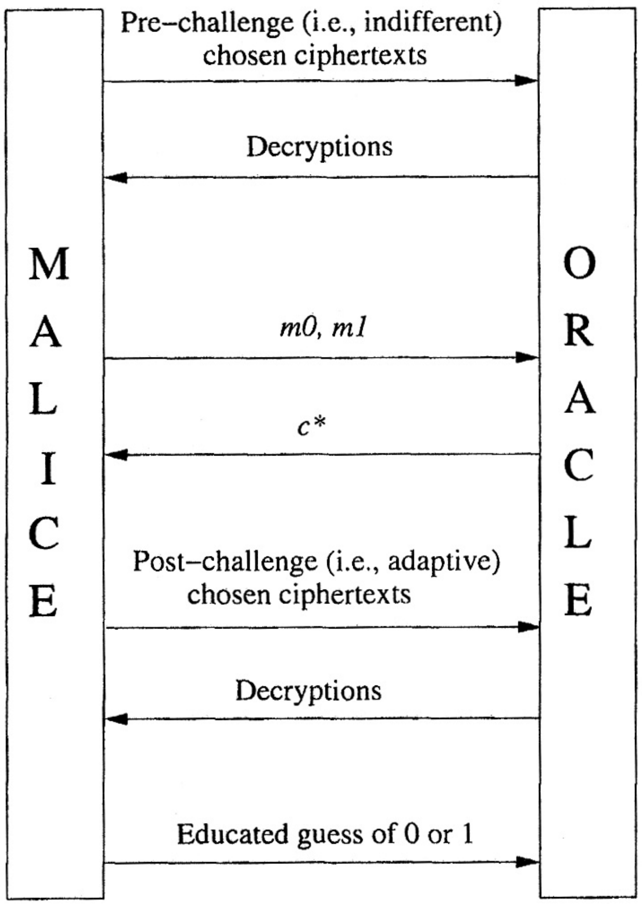
W. Mao, *Modern Cryptography: Theory and Practice* (Prentice Hall, 2004)

RSA-OAEP algorithm

- $|N| = |m| + k_1 + k_0$; 2^{-k_0} and 2^{-k_1} negligible
- Encryption
  - $r = \mathrm{rand}(k_0)$; $s = (m||0..0) \oplus G(r)$; $t = r \oplus H(s)$
  - $c = (s||t)^e \bmod N$
- Decryption
  - $s||t = c^d \bmod N$; $|s| = |m| + k_1$; $|t| = k_0$
  - $u = t \oplus H(s)$; $\mathrm{v} = s \oplus G(u)$
  - If $v == m||0^{k_1}$, extract m; else reject

IND-CCA2 game

- Oracle provides PPT Malice with requested decryptions (except for c^*)
- Malice is capable if he guesses which of the two plaintexts c^* encrypts
- Required: non-negligible Adv = 2 Pr["correct guess" | history] - 1



W. Mao, *Modern Cryptography: Theory and Practice*

Random oracle

- Idealized hash function $\mathcal{G}$: $\{0,1\}^k \longrightarrow \{0,1\}^n$
- Output
  - Uniformly random (really!)
  - Deterministic
  - Efficient
- Imaginary
- Computationally indistinguishable from a good real-world hash function

Simulating a random oracle

- At startup, initialize $\mathcal{G}$-list to empty
- When value $\mathcal{G}(a)$ is queried
  - Lookup a in $\mathcal{G}$-list
  - If not found
    - Generate random value for $\mathcal{G}(a)$
    - Store $(a, \mathcal{G}(a))$ in the $\mathcal{G}$-list
  - Return the stored value
- Precise local simulation in PPT

Proof of IND-CCA2 security

- General idea:

  $\exists$ algorithm *A* that is IND-CCA2 capable

  $\Rightarrow$ OWTP *f* (e.g., RSA) can be inverted

  $\Leftrightarrow$

  OWTP *f* is not invertible

  $\Rightarrow$ IND-CCA2 security
- "Reduction to contradiction"
- PPT algorithms, non-negligible advantages

RSA-inverting algorithm *M*

- Input: Random point $c^* = f(w^*)$
- Output: Preimage $w^* = f^{-1}(c^*)$
- Encapsulates IND-CCA2 capable *A*
- Random-oracle simulator of the OAEP hash functions *G* and *H* for *A*
- Decryption oracle for *A*
  - Based on the *G*- and *H*-lists
  - May reject even if *A* submits a valid ciphertext

$w^* = s^* || t^* = f^{-1}(c^*)$

Inversion process

- M plays two IND-CCA2 games with A
  - Round 1: M challenges A with c^*
    - c^* has nothing to do with (m_0, m_1)!
  - Round 2: M challenges A with $c^*_2 = c^* \alpha^e \bmod N$
    - Random $\alpha \in \mathbb{Z}^*_N$ (probability of bad α negligible)
- If A queries $H(s^*)$ and $H(s^*_2)$, M finds $f^{-1}(c^*)$
  - PT lattice method by Fujisaki *et al*.
- How probable are the queries?
- What if A discovers c^* is a hoax?

$s = (m||0..0) \oplus G(r)$

$t = r \oplus H(s)$

$c = f(s||t)$

Decryption oracle service

- Maintain a list of potential ciphertext-plaintext tuples $\{(f(w_i), w_i, v_i)\}_i$

  For each $(g, G(g))$ for each $(h, H(h))$

  $w = h||(g \oplus H(h))$; $v = G(g) \oplus h$
- If $f(w_i) = c^*$, $w_i = w^* = f^{-1}(c^*)$; success!
- To decrypt c
  - If $c = f(w_i)$ and $v_i = \Delta||0..0$, return $\Delta = m$
  - Else reject

$s||t = f^{-1}(c)$
$r = t \oplus H(s)$
$m||0..0 = s \oplus G(r)$

Quality of the decryption service

- If A creates a valid c without G or H, M rejects c illegally
- $(s, H(s))$ missing $\Rightarrow$ Pr["r correct"] = 2^{-k_0}
  $\Rightarrow \Pr[s \oplus G(r) = \Delta||0^{k_1}] = 2^{-k_1}$
- Similarly for missing $(r, G(r))$
- If $G(r)$ or $H(s)$ not queried, reject is correct except for (negligible) Pr ~ $2^{-k_0} + 2^{-k_1}$
- Good decryption quality

$s^*||t^* = f^{-1}(c^*)$
$r^* = t^* \oplus H(s^*)$
$m^*||0..0 = s^* \oplus G(r^*)$

Likelihood of successful inversion

- Define the following events
- **DBad** = *M* rejects a valid ciphertext
- **AskH** = *A* has queried for $H(s^*)$
- **AskG** = *A* has queried for $G(r^*)$
- **AskH** or **AskG** may reveal the deception in c^*
  - **Bad** = **AskH** U **AskG** U **DBad**
- **AWins** = *A* can correctly guess the IND-CCA2 game challenge bit *b*

$\Pr[A,B]$
$= \Pr[A|B]\ \Pr[B]$
$\le \Pr[B]$

Likelihood of successful inversion

- $\Pr[\mathbf{AWins} | \neg\mathbf{Bad}]$
  $\equiv \Pr[\mathbf{AWins}, \neg\mathbf{Bad}] \,/\, \Pr[\neg\mathbf{Bad}] = 1/2$
  $\Rightarrow \Pr[\mathbf{AWins}, \neg\mathbf{Bad}] = (1 - \Pr[\mathbf{Bad}])/2$
- $\mathrm{Adv} + 1/2 = \Pr[\mathbf{AWins}]$
  $\equiv \Pr[\mathbf{AWins}, \neg\mathbf{Bad}] + \Pr[\mathbf{AWins}, \mathbf{Bad}]$
  $\le \Pr[\mathbf{AWins}, \neg\mathbf{Bad}] + \Pr[\mathbf{Bad}]$
  $= \Pr[\mathbf{Bad}]/2 + 1/2$
- $\Rightarrow \Pr[\mathbf{Bad}] \ge 2\mathrm{Adv}$

Pr[A U B]
= Pr[A] + Pr[B] – Pr[A,B]
≤ Pr[A] + Pr[B]

Likelihood of successful inversion

- Pr[**Bad**] ≤ Pr[**AskH** U **AskG**] + Pr[**DBad**]
  = Pr[**AskH**] + Pr[¬**AskH**,**AskG**] + Pr[**DBad**]
  ≤ Pr[**AskH**] + Pr[**AskG**|¬**AskH**] + Pr[**DBad**]
- **AskG**|¬**AskH** = $G(r^*)$ has been queried when $H(s^*)$ has not ⇒ Pr[**AskG**|¬**AskH**] = 2^{-k_0}
- Pr[**AskH**] ≥ $2(\mathrm{Adv} - (2^{-k_0} + 2^{-k_1-1}))$
- *M* obtains *s** with non-negligible probability
  - After this, *M* can let *A* know the truth about *c**

Fujisaki's method

- $|s^*| > |w^*|/2$; $\mathrm{Int}(t^*) < \sqrt{N}$
- Use s^* and s^*_2 to solve for $\mathrm{Int}(t^*)$ in $(2^{k_0}\,\mathrm{Int}(s^*) + \mathrm{Int}(t^*))^e \equiv c^* \pmod{N}$
- q = larger H-list length
- For each pair (s,s_2), solve for $\mathrm{Int}(t)$ twice
- $\Rightarrow$ Inversion takes time $2\tau_A + q^2 O((\log_2 N)^3)$

  τ_A = running time of IND-CCA2 on RSA-OAEP

Practically safe parameters

- Evaluating *H* and *G* is very efficient in reality
- Dedicated attacker may make $q \approx 2^{50}$ queries
- Now RSA inversion time $> 2^{100} \gg 2^{86}$ for the Number Field Sieve method, if $|N| = 1024$
- $|N| = 2048$ considered safe
  - NFS takes time 2^{116}
- $k_0 = k_1 = 160$ considered safe
- Up to 84% of $s||t$ can be actual message m

Part II: Outline

- Decisional Diffie-Hellman problem
- Cramer-Shoup scheme
  - Key setup
  - Encryption and decryption
- Overview of proof of IND-CCA2 security
  - DDH reduction

Decisional Diffie-Hellman problem

- Given
  - Description of an abelian group G
  - $(g, g^a, g^b, g^c) \in G^4$; $g = \text{gen}(G)$
- Is $ab \equiv c \pmod{\text{ord}(G)}$?
- Easy in supersingular elliptic-curve groups
- Hard in groups of finite fields

Cramer-Shoup

- R. Cramer and V. Shoup in 1998
  - CCA2-enhanced ElGamal encryption
  - More public and private parameters
  - Hashing
- IND-CCA2 secure
  - Assuming Finite-Field Decisional D-H is hard
- Data integrity check
- Resource need ~ twice that of ElGamal

Cramer-Shoup key setup

- Large prime q = ord(G); G = plaintext space
- Pick random $g_1, g_2 \in G$
- Pick random $x_1, x_2, y_1, y_2, z \in [0,q)$
- $c = g_1^{x_1} g_2^{x_2}$; $d = g_1^{y_1} g_2^{y_2}$; $h = g_1^{z}$
- Choose a hash function H: $G^3 \rightarrow [0,q)$
- Public key: (g_1, g_2, c, d, h, H)
- Private key: (x_1, x_2, y_1, y_2, z)

Cramer-Shoup operation

- Encryption
  - Message $m \in G$; Pick random $r \in [0,q)$
  - $u_1 = g_1^r$; $u_2 = g_2^r$; $e = h^r m$
  - $\alpha = H(u_1, u_2, e)$; $v = c^r d^{r\alpha}$
  - The ciphertext is (u_1, u_2, e, v)
- Decryption
  - $\alpha = H(u_1, u_2, e)$
  - If $u_1^{x1+y1\alpha} u_2^{x2+y2\alpha} = v$, $m = e/u_1^z$
  - Else reject

Proof of IND-CCA2 security

- Same general idea as with RSA-OAEP:

  $\exists$ algorithm A that is IND-CCA2 capable

  $\Rightarrow$ Finite-Field Decisional Diffie-Hellman can be answered efficiently by M_A

  $\Leftrightarrow$

  FFDDH is hard $\Rightarrow$ IND-CCA2 security

- Better than the proof for RSA-OAEP
  - No need for controversial random oracles
  - Reduction DDH $\rightarrow$ IND-CCA2 is linear

Reduction

- M_A: Can the arbitrary input $(g_1, g_2, u_1, u_2) \in G^4$ be a Diffie-Hellman quadruple? (DDH)
- Play the IND-CCA2 game with A
  - Receive chosen (m_0, m_1), challenge with C^*
- Input is a DHq $\Rightarrow$ C^* encrypts m_b
- Input is not a DHq $\Rightarrow$ C^* uniformly distributed
- Based on A's guess on b, M_A can decide whether (g_1, g_2, u_1, u_2) is a DHq or not