Summary

- Encryption
  - PKCS #1 v1.5
  - PKCS #1 v2.0 : OAEP
- Signature
  - PKCS #1 v1.5/2.0
  - PKCS #1 v2.1 : PSS
- Conclusion
Summary

- Encryption
  - PKCS #1 v1.5
  - PKCS #1 v2.0 : OAEP
- Signature
  - PKCS #1 v1.5/2.0
  - PKCS #1 v2.1 : PSS
- Conclusion

RSA

Rivest - Shamir - Adleman 1978

- $n = pq$ : public modulus
- $e$ : public exponent
- $d = e^{-1} \mod \varphi(n)$ : private

en/de-cryption

$E(m) = m^e \mod n$

$D(c) = c^d \mod n$

Relies on the so-called RSA problem:
extracting $e$-th roots mod $n$
Plain-RSA: Weak Security

• One-Wayness = RSA Problem
• Deterministic:
  cannot achieve **Semantic Security**
  Does $c$ encrypt $m_0$ or $m_1$?
  Re-encrypt $m_0$, and check whether it is $c$

• Multiplicativity:
  cannot prevent **Chosen-Ciphertext Attacks**
  With $c = E_e(m) = m^e \mod n$
  Compute $c' = 2^e c \mod n$, ask for $m'$
  Note that $c' = (2m)^e \mod n$, thus $m = m'/2 \mod n$

⇒ need of padding

PKCS #1 v 1.5

<table>
<thead>
<tr>
<th>EM</th>
<th>00</th>
<th>02</th>
<th>Padding String length ≥ 8 bytes ≠ 0</th>
<th>00</th>
<th>Data Block</th>
</tr>
</thead>
</table>

• Efficient encoding/decoding
• Probabilistic encryption
• Breaks multiplicativity
• But…
  a random ciphertext is valid
  with non-negligible probability $\approx 2^{-16}$
PKCS #1 v 1.5

Valid ciphertext
⇒ the MSB of the encoded message is at zero
   – The bit-security of RSA says that any bit of the $e$-th root is as hard as the whole $e$-th root
   – Any bit-leakage is serious
   – Here: 2 full bytes are leaked!

Breaking PKCS #1 v 1.5

Valid ciphertext $C = EM^e \mod n$
⇒ $2 \times 256^{k-2} \leq EM < 3 \times 256^{k-2}$

Challenge ciphertext $C = EM^e \mod n$
   – Find small $S$ such that $C' = C \times S^e \mod n$ valid:
      for some $0 < r < S$,
      
      $2 \times 256^{k-2} - rn \leq EM \times S < 3 \times 256^{k-2} - rn$

⇒ EM lies in a small interval $[a, b]$
Breaking PKCS #1 v 1.5

Choose a new $S$ such that the sets

$$\{ a \ S \mod n, \ (a+1) \ S \mod n, \ldots, \ b \ S \mod n \}$$

and $[2 \times 256^{k-2}, 3 \times 256^{k-2}]$ overlap

Validity of $C' = C \ S^e \mod n$ tells in which part it is

$\Rightarrow$ new small inverval $[a', b']$ for EM

Approx.: any new valid ciphertext
reduces the interval by $1/2$

---

Breaking PKCS #1 v 1.5

- Reaction Attack (validity requests) breaks the One-Wayness
- Given a challenge ciphertext $c^*$, after a few thousand of requests $c$, one can recover the plaintext $m^*$
Summary

- Encryption
  - PKCS #1 v1.5
  - PKCS #1 v2.0 : OAEP
- Signature
  - PKCS #1 v1.5/2.0
  - PKCS #1 v2.1 : PSS
- Conclusion

**Summary**

- Encryption
  - PKCS #1 v1.5
  - PKCS #1 v2.0 : OAEP
- Signature
  - PKCS #1 v1.5/2.0
  - PKCS #1 v2.1 : PSS
- Conclusion

**OAEP**

Bellare-Rogaway '94

$f$ a trapdoor one-way permutation (e.g. RSA)
then (with $G \rightarrow \{0,1\}^n$ and $H \rightarrow \{0,1\}^\ell$)

\[ M = m || 0^k \]
\[ r \text{ random} \]

\[\begin{align*}
M & \quad \Downarrow \quad G \\
M & \quad \Downarrow \quad H \\
EM & \quad \Downarrow \quad t
\end{align*}\]

$E(m,r) : \text{Compute } a,b \text{ then } c = f(s||t) = EM^e \mod n$

$D(c) : \text{Compute } EM = s||t = f^{-1}(c) = c^d \mod n,$
inverting OAEP, and check redundancy
In 1994, Bellare and Rogaway proved that
• the OAEP construction provides an IND-CPA cryptosystem under the OW of $f$
• it is plaintext-aware (PA94)

\[
\text{proven: } \text{IND-CPA} + \text{PA94} \Rightarrow \text{IND-CCA1}
\]

But widely believed:
• not proven: \(\text{IND-CPA} + \text{PA94} \Rightarrow \text{IND-CCA2}\)

and namely for OAEP…

In 1998, improved plaintext-awareness (PA98)

\[
\text{proven: } \text{IND-CPA} + \text{PA98} \Rightarrow \text{IND-CCA2}
\]

But… PA98 of OAEP never studied
And IND-CCA2 of OAEP still widely believed under the sole OW of $f$

and namely for RSA-OAEP

\[
\text{RSA-OAEP: the most efficient and “provably secure” construction}
\]

\[
\Rightarrow \text{became the new PKCS #1 v2.0}
\]
IND-CCA2 under OW of $f$

However, in 2000, Shoup showed a counter-example:
- a trapdoor one-way permutation $f$
- so that $f$-OAEP can be broken: malleable
from a ciphertext $c$ of an unknown message $m$,
  one can build a ciphertext $c'$ of $m' = m \oplus 1$
$\Rightarrow$ break OW-CCA2, and thus IND-CCA
  Given a challenge $c$, the encryption of $m$,
  one derives the ciphertext $c'$ of $m \oplus 1$
  one request to the decryption oracle is enough!

Counter-Example

- Let $g$ be a trapdoor one-way permutation
  so that there exists an algorithm $A$,
  which on $a$ and $g(x)$ computes $g(x \oplus a)$
- Let us define $f(s,t) = s \parallel g(t)$, which is clearly
  a trapdoor one-way permutation
**Malleability**

\[ m \oplus \delta \rightarrow m' \]

\[ r \]

\[ T = H(s) \oplus H(s') \]

\[ s \]

\[ t \]

\[ \oplus \delta \| 0^k \rightarrow s' \]

\[ \oplus T \rightarrow t' \]

---

**Malleability (details)**

One receives \( c = E(m,r) = f(s,t) = s \| g(t) \) where

\[ M = m \| 0^k, \quad s = M \oplus G(r), \quad t = r \oplus H(s) \]

- One gets \( s \), and computes \( s' = s \oplus \Delta \)
  for some \( \Delta = \delta \| 0^k \)

- One computes \( T = H(s) \oplus H(s') \), and \( t' = t \oplus T \)
  as well as \( g(t \oplus T) \) granted \( A \) on \( g(t) \) and \( T \)

\[ r' = t' \oplus H(s') = t \oplus T \oplus H(s') = t \oplus H(s) = r \]

\[ M' = s' \oplus G(r) = s \oplus \Delta \oplus G(r) = M \oplus \Delta = (m \oplus \delta) \| 0^k \]

- \( c' = f(s',t') \) is a new ciphertext: of \( m \oplus \delta \)
Partial-Domain One-Wayness

From \( c = E(m,r) = f(s,t) \)

\[ \Rightarrow c' = E(m \oplus \delta, r) \] for any \( \delta \) of his choice

- without asking \( G(r) \Rightarrow \text{OW of } f \) not broken
- but asking \( H(s) \Rightarrow \text{partial-domain OW of } f \)

This intuition can be made formal:

Break IND-CCA2 of \( f \)-OAEP,

\[ \Rightarrow \text{partially invert } f \]

Fujisaki-Okamoto-Pointcheval-Stern Crypto ‘01

RSA: a Particular Case

The RSA permutation is particular:

Partial Domain One-Wayness

\[ \Leftrightarrow \text{One-Wayness} \]

Consequence:

RSA-OAEP is IND-CCA2 under the classical RSA assumption

Note: Shoup repaired the proof for RSA exponent 3 only, we repaired it for any exponent
PKCS #1 v 2.0

After Bleichenbacher’s attack, the OAEP construction was adopted by RSA in PKCS #1 v 2.0 (and still in v 2.1)

Even if a construction is provably secure, careless implementations often lead to very weak cryptosystems

*e.g. invalidity reasons* must be indistinguishable
– MSB different of zero
– Redundancy not satisfied

---

Summary

- Encryption
  - PKCS #1 v1.5
  - PKCS #1 v2.0 : OAEP
- Signature
  - PKCS #1 v1.5/2.0
  - PKCS #1 v2.1 : PSS
- Conclusion
Plain-RSA Signature

\[ S(m) = m^d \mod n \]
\[ V(m, \sigma) = (m = \sigma^e \mod n) \]

Existential forgery:
- choose a random \( \sigma \)
- compute \( m = \sigma^e \mod n \)

PKCS #1 v 1.5/2.0

<table>
<thead>
<tr>
<th>EM</th>
<th>00</th>
<th>01</th>
<th>Padding String length ≥ 8 bytes ≠ 0</th>
<th>00</th>
<th>Digest Info</th>
</tr>
</thead>
</table>

\[ S(m) = f^{-1}(EM) \]
\[ V(m, \sigma) = (f(\sigma) = EM) \]
PKCS #1 v 1.5/2.0

<table>
<thead>
<tr>
<th>EM</th>
<th>00</th>
<th>01</th>
<th>Padding String length ≥ 8 bytes ≠ 0</th>
<th>00</th>
<th>Digest Info</th>
</tr>
</thead>
</table>

- Digest Info = \( \text{HashID and } H(m) \)
  - It is small, and the padding string can be long… under the control of the adversary
- Using the multiplicativity of RSA
  - A weakness has been found in 1999

### Attack Idea

- A lot of freedom in the Padding String
- Get many \( \text{EM}_i \) for several \( i \) such that
  \[
  \text{EM} = \prod \text{EM}_i \Rightarrow \sigma = \prod \sigma_i
  \]
  - After several queries to the signing oracle, one can build a new signature
New Version

Applied on a slight variant of ISO 9796-1
But theoretical only on PKCS #1 v1.5
... less efficient than factoring!

Anyway, a provably secure construction was better.

Summary

- Encryption
  - PKCS #1 v1.5
  - PKCS #1 v2.0: OAEP
- Signature
  - PKCS #1 v1.5/2.0
  - PKCS #1 v2.1: PSS
- Conclusion
**Probabilistic Signature Scheme**

Bellare-Rogaway ‘96

\[ m \quad r \]

\[ H \quad G \quad \oplus \]

\[ 0 \quad w \quad s \quad t \]

\[ F \]

\[ k = k_0 + k_1 + k_2 + 1 \]
\[ \{0,1\}^{k-1} \subset X \subset \{0,1\}^k \]
\[ f : X \to X \]

\[ y = 0 \| w \| s \| t \]
\[ \sigma = f^{-1}(y) \]

---

**RSA - PSS**

- \( n, k \)-bit RSA modulus \((k = k_0 + k_1 + k_2 + 1)\)
- \( n, e \): public key
- \( d \): private key

\[
F : \{0,1\}^{k_2} \to \{0,1\}^{k_0} \text{ and } G : \{0,1\}^{k_2} \to \{0,1\}^{k_1} \\
H : \{0,1\}^x \to \{0,1\}^{k_2}
\]

\[ w = H(m, r), s = G(w) \oplus r, t = F(w) \]

\[ y = 0 \| w \| s \| t \text{ and } \sigma = y^d \mod n \]
RSA - PSS

- Provably secure
  no existential forgeries
  under chosen-message attacks
- *Efficient* security proof
  $\Rightarrow$ practical security
- Probabilistic
  $\Rightarrow$ PKCS #1 v2.1

Summary

- Encryption
  - PKCS #1 v1.5
  - PKCS #1 v2.0 : OAEP
- Signature
  - PKCS #1 v1.5/2.0
  - PKCS #1 v2.1 : PSS
- Conclusion
Almost all the previously defined paddings, without any security proof, have been showed to be flawed.

Any new standard (ISO, IEEE, IETF, PKCS, …) needs a security proof.