Cryptographie

Cours no. 3

Jean-Sébastien Coron

coron@clipper.ens.fr

Université du Luxembourg
What is cryptography?

Cryptography’s aim is to construct schemes that achieve some goal despite the presence of an adversary.

Example: encryption, key-exchange, signature, electronic voting...

Scientific approach:

To be rigorous, one must specify what it means to be secure.
Then one tries to construct schemes that achieve the desired goal, in a provable way.
Plain RSA encryption and signature cannot be used!
The RSA signature scheme

Key generation:
- Public modulus: $N = p \cdot q$ where $p$ and $q$ are large primes.
- Public exponent: $e$
- Private exponent: $d$, such that $d \cdot e = 1 \pmod {\phi(N)}$

To sign a message $m$, the signer computes:
- $s = m^d \pmod N$
- Only the signer can sign the message.

To verify the signature, one checks that:
- $m = s^e \pmod N$
- Anybody can verify the signature
Hash-and-sign paradigm

There are many attacks on basic RSA signatures:

- **Existential forgery:** \( r^e = m \mod N \)
- **Chosen-message attack:** \( (m_1 \cdot m_2)^d \equiv m_1^d \cdot m_2^d \mod N \)

To prevent from these attacks, one usually uses a hash function. The message is first hashed, then padded.

- \( m \rightarrow H(m) \rightarrow 1001\ldots 0101\| H(m) \)
- **Example:** PKCS#1 v1.5:
  \( \mu(m) = 0001 \text{ FF}\ldots \text{FF00}\| c_{SHA}\| \text{SHA}(m) \)
- **ISO 9796-2:** \( \mu(m) = 6A\| m[1]\| H(m)\| \text{BC} \)
Proofs for signature schemes

Strongest security notion (Goldwasser, Micali and Rivest, 1988):
- It must be infeasible for an adversary to forge the signature of a message, even if he can obtain the signature of messages of his choice.

Security proof:
- Show that from an adversary who is able to forge signature, you can solve a difficult problem, such as inverting RSA.

Examples of provably secure signature schemes:
- Full Domain Hash (FDH)
- Probabilistic Signature Scheme (PSS)
The FDH scheme

The FDH signature scheme:
- was designed in 1993 by Bellare and Rogaway.

\[ m \rightarrow H(m) \rightarrow s = H(m)^d \mod N \]

- The hash function \( H(m) \) has the same output size as the modulus.

Security of FDH
- FDH is provably secure in the random oracle model, assuming that inverting RSA is hard.
- In the random oracle model, the hash function is replaced by an oracle which outputs a random value for each new query.
We want to show that FDH is a secure signature scheme:

- Even if the adversary requests signatures of messages of his choice, he is still unable to produce a forgery.
- Forgery: a couple \((m', s')\) such that \(s\) is a valid signature of \(m\) but the signature of \(m\) was never requested by the adversary.
Security proof for FDH

- Proof in the random oracle model
  - The adversary cannot compute the hash-function by himself.
  - He must make a request to the random oracle, which answers a random, independently distributed answer for each new query.
    - Randomly distributed in $\mathbb{Z}_N$.

- Idealized model of computation
  - A proof in the random oracle model does not imply that the scheme is secure when a concrete hash-function like SHA-1 is used.
  - Still a good guarantee.
Security proof

Forger

\( (N, e) \)

\( \text{H(m)} = ? \)

\( \text{S(m)} = ? \)

\( (M', s') \)

\( y^d \mod N \)

Reduction

\( (N, e, y) \)
Proof of security

- We assume that there exists a successful adversary.
  - This adversary is an algorithm that given the public-key \((N, e)\), after at most \(q_{hash}\) hash queries and \(q_{sig}\) signature queries, outputs a forgery \((m', s')\).

- We will use this adversary to solve a RSA challenge: given \((N, e, y)\), output \(y^d \mod N\).
  - The adversary’s forgery will be used to compute \(y^d \mod N\), without knowing \(d\).
  - If solving such RSA challenge is assumed to be hard, then producing a forgery must be hard.
Let $q_{hash}$ be the number of hash queries and $q_{sig}$ be the number of signature queries.

Select a random $j \in [1, q_{hash} + q_{sig} + 1]$.

Answering a hash query for the $i$-th message $m_i$:
- If $i \neq j$, answer $H(m_i) = r_i^e \mod N$ for random $r_i$.
- If $i = j$, answer $H(m_j) = y$.

Answering a signature query for $m_i$:
- If $i \neq j$, answer $r_i = H(m_i)^d \mod N$, otherwise $(i = j)$ abort.
- We can answer all signature queries, except for message $m_j$. 
Using the forgery

Let \((m', s')\) be the forgery

- We assume that the adversary has already made a hash query for \(m'\), i.e., \(m' = m_i\) for some \(i\).
- Otherwise we can simulate this query.

Then if \(i = j\), then \(s' = H(m_j)^d = y^d \mod N\).

We return \(s'\) as the solution to the RSA challenge \((N, e, y)\).
Success probability

- Our reduction succeeds if \( i = j \)
  - This happens with probability \( \frac{1}{q_{hash} + q_{sig} + 1} \)
- From a forger that breaks FDH with probability \( \varepsilon \) in time \( t \), we can invert RSA with probability \( \varepsilon' = \frac{\varepsilon}{q_{hash} + q_{sig} + 1} \) in time \( t' \) close to \( t \).
- Conversely, if we assume that it is impossible to invert RSA with probability greater than \( \varepsilon' \) in time \( t' \), it is impossible to break FDH with probability greater than
  \[
  \varepsilon = (q_{hash} + q_{sig} + 1) \cdot \varepsilon'
  \]
  in time \( t \) close to \( t' \).
Improving the security bound

Instead of letting \( H(m_i) = r_i^e \mod N \) for all \( i \neq j \) and \( H(m_j) = y \), one lets

\[
\begin{align*}
\diamond & \quad H(m_i) = r_i^e \mod N \text{ with probability } \alpha \\
\diamond & \quad H(m_i) = r_i^e \cdot y \mod N \text{ with probability } 1 - \alpha
\end{align*}
\]

Idea (published at CRYPTO 2000 by me).

\[
\begin{align*}
\diamond & \quad \text{When } H(m_i) = r_i^e \mod N \text{ one can answer the signature query but not use a forgery for } m_i. \\
\diamond & \quad \text{When } H(m_i) = r_i^e \cdot y \mod N \text{ one cannot answer the signature query but can use the forgery to compute } y^d \mod N. \\
\diamond & \quad \text{Optimize for } \alpha.
\end{align*}
\]
Improving the bound

- Probability that all signature queries are answered:
  - A signature query is answered with probability $\alpha$
  - At most $q_{\text{sig}}$ signature queries $\Rightarrow P \geq \alpha^{q_{\text{sig}}}$

- Probability that the forgery $(m_i, s')$ is useful:
  - Useful if $H(m_i) = r_i^c \cdot y \mod N$
  - $s' = H(m_i)^d = r_i \cdot y^d \mod N \Rightarrow y^d = s' / r_i \mod N$

- Global success probability:
  - $f(\alpha) = \alpha^{q_{\text{sig}}} \cdot (1 - \alpha)$
  - $f(\alpha)$ is maximum for $\alpha_m = 1 - 1/(q + 1)$
  - $f(\alpha_m) \approx 1/(e \cdot q_{\text{sig}})$ for large $q_{\text{sig}}$
Success probability

From a forger that breaks FDH with probability $\varepsilon$ in time $t$, we can invert RSA with probability $\varepsilon' = \frac{\varepsilon}{4 \cdot q_{sig}}$ in time $t'$ close to $t$.

Conversely, if we assume that it is impossible to invert RSA with probability greater than $\varepsilon'$ in time $t'$, it is impossible to break FDH with probability greater than $\varepsilon = 4 \cdot q_{sig} \cdot \varepsilon'$ in time $t$ close to $t'$.

Concrete values

- With $q_{hash} = 2^{60}$ and $q_{sig} = 2^{30}$, we obtain $\varepsilon = 2^{32} \varepsilon'$ instead of $\varepsilon = 2^{60} \cdot \varepsilon'$
- More secure for a given modulus size $k$.
- A smaller modulus can be used for the same level of security: improved efficiency.
The PSS signature scheme

- PSS (Bellare and Rogaway, Eurocrypt’96)
  - IEEE P1363a and PKCS#1 v2.1.
  - 2 variants: PSS and PSS-R (message recovery)
  - Provably secure against chosen-message attacks

- PSS-R:
  \[ \mu(M, r) = \omega \| s \]

\[ M \| r \]

\[ H \]

\[ G \]

\[ \omega \]

\[ s \]
What is cryptography?

Cryptography’s aim is to construct schemes that achieve some goal despite the presence of an adversary.

Scientific approach:

To be rigorous, one must specify what it means to be secure.

Then one tries to construct schemes that achieve the desired goal, in a provable way.

Plain RSA encryption and signature cannot be used!