Summary

• Introduction
• Signature
  – FDH
  – PSS
  – Forking Lemma
  – Generic Model
• Conclusion
Summary

- Introduction
- Signature
  - FDH
  - PSS
  - Forking Lemma
  - Generic Model
- Conclusion

(Trapdoor) One-Way Functions

- In the following, we consider any function $f$ which is assumed to be one-way:

$$\text{Succ}^{\text{ow}}_f(A) = \Pr_x[f(A(y)) = y | y = f(x)]$$

- This function may be trapdoor:
  - $g$ is the inverse function, available granted a private information

Examples:
- OW function = DL
- Trapdoor OW function = RSA or CDH
- Trapdoor OW permutation = RSA
**Proof by Reduction**

Reduction of a problem $P$ to an attack $Atk$:

- Let $A$ be an adversary that breaks the scheme then $A$ can be used to solve $P$

**Instance** $I$ of $P$  

**Solution of $I$**

$P$ intractable $\Rightarrow$ scheme unbreakable

---

**Complexity Estimates**

Estimates for integer factoring  Lenstra-Verheul 2000

<table>
<thead>
<tr>
<th>Modulus (bits)</th>
<th>Mips-Year ($\log_2$)</th>
<th>Operations ($\text{en log}_2$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>512</td>
<td>13</td>
<td>58</td>
</tr>
<tr>
<td>1024</td>
<td>35</td>
<td>80</td>
</tr>
<tr>
<td>2048</td>
<td>66</td>
<td>111</td>
</tr>
<tr>
<td>4096</td>
<td>104</td>
<td>149</td>
</tr>
<tr>
<td>8192</td>
<td>156</td>
<td>201</td>
</tr>
</tbody>
</table>

Can be used for RSA too

Lower-bounds for DL in $\mathbb{Z}_p^*$
Practical Security

- Complexity theory: $T$ polynomial
- Exact Security: $T$ explicit
- Practical Security: $T$ small (linear)

Authentication

- Signature Algorithm, $S$
- Verification Algorithm, $V$

Security: impossible to forge a valid $\sigma$ without $k_s$
**Basic Goal**

- **Existential Forgery:**
  without the private key,
  it is computationally impossible to forge
  a valid message-signature pair

\[
\text{Succ}^\text{ef} (A) = \Pr[\mathbf{V}(m, \sigma) = 1 | A(k_v) = (m, \sigma)]
\]

---

**Chosen-Message Attacks**

- **Chosen-Message Attacks (CMA)**
  In the list of message-signature pairs,
  the messages are adaptively chosen
  by the adversary
  \[ \Rightarrow \text{strongest attack} \]
• Introduction
• Signature
  – FDH
  – PSS
  – Forking Lemma
  – Generic Model
• Conclusion
FDH Signature

- $f$ is a trapdoor one-way permutation onto $X$
- $g$, is the inverse (granted the trapdoor)
- $H$ is hash function in the full domain $X$ of $f$
- $f$: public key
- $g$: private key

\[
S(m) = g(H(m)) \quad V(m, \sigma) = (f(\sigma) = H(m))
\]

$H =$ identity: Existential Forgery = easy!

$H =$ random oracle: EF-CMA = OW

FDH EF-CMA: Result

\[
\text{Succ}^{\text{ef-cma}}(A) \leq (q_H + q_S + 1) \text{Succ}^{\text{ow}}(t')
\]

where $t' = t + (q_H + q_S) T_f$

\[
\text{Succ}^{\text{ef-cma}}(t) \leq (q_H + q_S + 1) \times \text{Succ}^{\text{ow}}_f(t + (q_H + q_S) T_f)
\]
Comments: FDH

\[ \text{Succ}^{e_{-\text{cma}}}(t) \leq (q_H + q_s + 1) \times \text{Succ}^{ow}_f (t + (q_H + q_s)T_f) \]

Security bound: \(2^{75}\), and
\(2^{55}\) hash queries and \(2^{30}\) signing queries
If one can break the scheme within time \(T\),
one can invert \(f\) within time
\[ T' \leq (q_H + q_s + 1) (T + (q_H + q_s) T_f) \]
\[ \leq 2^{56} T + 2^{112} T_f \]

FDH-RSA

Security bound: \(2^{75}\), and
\(2^{55}\) hash queries and \(2^{30}\) signing queries
RSA (\(K\) bits) small exponent
If one can break the scheme within time \(T\),
one can invert RSA within time
\[ T' \leq 2^{131} + 2^{112} K^2 \]

RSA:
- 1024 bits \(\rightarrow 2^{132}\) (NFS: \(2^{80}\)) \(\times\)
- 2048 bits \(\rightarrow 2^{134}\) (NFS: \(2^{111}\)) \(\times\)
- 4096 bits \(\rightarrow 2^{136}\) (NFS: \(2^{149}\)) \(\checkmark\)
ESIGN

ESIGN is an application of the FDH paradigm to a many-to-one trapdoor OW function \( f \).

Under specific probabilistic properties, the previous proof still applies, but:

- A given \( y \) has many pre-images
- The signing oracle chooses a random one each time
- The simulator knows only one!

No EF but against SO-CMA only.

FDH-RSA: Improved Reduction

- In the case that \( f \) is random self-reducible, the reduction may be improved

\[
\text{Succ}_{\text{ef-cma}}(t) \leq (q_H + q_s + 1) \times \text{Succ}_{f}^{ow}(t + (q_H + q_s)T_f)
\]

\[
\Downarrow
\]

\[
\text{Succ}_{\text{ef-cma}}(t) \leq \frac{q_s + 1}{e} \times \text{Succ}_{f}^{ow}(t + (q_H + q_s + 1)T_f)
\]

Cf. Coron ‘00
FDH-RSA EF-CMA: Game 0

Adversary $A$
- $A_1(n,e) \rightarrow (m^*, \sigma^*)$

with permanent access to
- the signing oracle $S$ $q_S$ queries
- the random oracle $H$ $q_H$ queries

- One checks whether $(\sigma^*)^e \mod n = H(m^*)$
  
  Note: it may make one more call to $H$

- If the equality holds, and $m^* \notin \Lambda_S$, $s=1$,
  otherwise $s=0$

On this probability space, we consider event $S$: $s = 1$

In Game $i$: $S_i$

Note that

\[
\Pr[S_0] = \text{Succ}^{\text{ef-cma}}(A)
\]
FDH-RSA EF-CMA: Game 1

Any signing query is asked first to the random oracle $H$

One does not modify the probability space, but note that $q_H$ becomes $q'_H = q_H + q_s$:
$$\Pr[S_1] = \Pr[S_0]$$

FDH-RSA EF-CMA: Game 2

We replace the random oracle $H$ by the usual simulation: the list $\Lambda_H$
- is initially set to an empty list
- any new random answer is appended

One does not modify the probability space:
$$\Pr[S_2] = \Pr[S_1]$$
FDH-RSA EF-CMA: Game 3

One simulates the answers of $H$, using $y^*$, an external data $y^* = (x^*)^e \mod n$
For the $i^{th}$ query $m_i$, one flips a biased coin $b$
which is 1 with probability $p$, and 0 otherwise
One chooses $x$, computes $y = (y^*)^b x^e \mod n$
and sets $H(m) \leftarrow y$
Then $\Lambda_H \leftarrow (m,y,b,x)$, and $y$ is the output

One does not modify the probability space,
since $f$ is a permutation:
$$\Pr[S_3] = \Pr[S_2]$$

FDH-RSA EF-CMA: Game 4

One now simulates the signing oracle $S$:
For a query $m$, one looks for $(m,y,b,x) \in \Lambda_H$,
and outputs $x$ as the signature

By construction, $H(m) = y = (y^*)^b x^e \mod n$,
thus the simulation is perfect, unless $b = 1$.
One just conditions the game by an independent event, $b = 0$, of probability $1-p$:
$$\Pr[S_4] \geq \Pr[S_3] \times (1-p)^q_s$$
FDH-RSA EF-CMA: Game 4

One is given $y^*$
• $A_1(f) \rightarrow (m^*, \sigma^*)$

with permanent access to
- the signing oracle $S$ simulation
- the random oracle $H$ simulation
and $H(m^*) \leftarrow (y^*)^{b^*} (x^*)^e \mod n$

• One checks whether $(\sigma^*)^e \mod n = H(m^*)$
Event $S_4 \Rightarrow (\sigma^*)^e = H(m^*) = (y^*)^{b^*} (x^*)^e \mod n$

Thus $(\sigma^*/x^*)^e = y^* \mod n$ if $b^* = 1$

FDH-RSA EF-CMA: Sum up

• $\Pr[S_0] = \text{Succ}_{\text{ef-cma}}(A)$
• $\Pr[S_3] = \Pr[S_2] = \Pr[S_1] = \Pr[S_0]$
• $\Pr[S_4] \geq \Pr[S_3] \times (1-p)^q_s$
• $\Pr[S_4] \leq \text{Succ}_{\text{ow}}(t_4) / p$

\[
\Pr[S_0] = \text{Succ}_{\text{ef-cma}}(A) \leq \Pr[S_4] / (1-p)^q_s \\
\leq \text{Succ}_{\text{ow}}(t_4) / p (1-p)^q_s
\]
**FDH-RSA EF-CMA: Result**

\[
\text{Succ}^{ef-cma}(A) \leq \text{Succ}^{ow}(t') / p \cdot (1-p)^q_s
\]

where \( t' = t + (q_H + q_s + 1) T_f \)

Note that \( p \mapsto p \cdot (1-p)^q_s \) is maximal

for \( p = 1 / (q_s+1) \) and is approximately, but less than \( e / (q_s+1) \)

\[
\text{Succ}^{ef-cma}(t) \leq \frac{q_s+1}{e} \times \text{Succ}^{ow}_f\left( t + (q_H + q_s + 1)T_f \right)
\]

---

**Comments: FDH-RSA**

Security bound: \( 2^{75} \), and \( 2^{55} \) hash queries and \( 2^{30} \) signing queries

If one can break the scheme within time \( T \), one can invert RSA within time

\[
T' \leq (q_s + 1) \left( T + (q_H + q_s + 1) T_f \right) / e
\]

\[
\leq 2^{30} T + 2^{85} T_f
\]
FDH-RSA

Security bound: $2^{75}$, and $2^{55}$ hash queries and $2^{30}$ signing queries

RSA ($K$ bits) small exponent

If one can break the scheme within time $T$, one can invert RSA within time

$$T' \leq 2^{105} + 2^{85} K^2$$

RSA:
- 1024 bits $\rightarrow 2^{106}$ (NFS: $2^{80}$) $\times$
- 2048 bits $\rightarrow 2^{107}$ (NFS: $2^{111}$) $\checkmark$
- 4096 bits $\rightarrow 2^{109}$ (NFS: $2^{149}$) $\checkmark$

Summary

- Introduction
- Signature
  - FDH
  - PSS
  - Forking Lemma
  - Generic Model
- Conclusion
Probabilistic Signature Scheme
Bellare-Rogaway '96

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

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

RSA-PSS

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

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

\[
w = H(m, r), s = G(w) \oplus r, t = F(w)\]
\[
y = 0||w||s||t \quad \text{and} \quad \sigma = y^d \mod n\]
**RSA-PSS EF-CMA: Game 0**

Adversary $A$
- $A_1(n,e) \rightarrow (m^*, \sigma^*)$

with permanent access to
- the signing oracle $S$ ($q_S$ queries)
- the random oracles $F,G,H$ ($q_F, q_G, q_H$ queries)

On this probability space, we consider event $S$: $V(m^*, \sigma^*) = 1$

In Game $i$: $S_i$

Note that

$$\Pr[S_0] = \text{Succ}^{ef-cma}(A)$$

$V(m^*, \sigma^*) = 1 \iff$ with $y = f(\sigma^*) = 0||w||s||t$

and $r = G(w) \oplus s$

then $t = F(w)$ and $w = H(m^*, r)$
We replace the random oracles $F$, $G$ and $H$ by the usual simulations: the lists $\Lambda_F$, $\Lambda_G$ and $\Lambda_H$

- initially set to an empty list
- any new random answer is appended

One does not modify the probability space:

$\Pr[S_1] = \Pr[S_0]$
RSA-PSS: Game 1 to Game 2

\[ w, s \text{ and } t \text{ are uniformly distributed, thus } t, r \oplus s \text{ and } w \text{ are so too} \]

The distributions are thus unchanged.

A problem may occur if \( F(w) \) or \( G(w) \) have already been queried or defined.

- \( q_F \) values for \( w \) have been queried to \( F \) by \( A \)
- \( q_G \) values for \( w \) have been queried to \( G \) by \( A \)
- \( q_s \) values for \( w \) have been queried to \( F/G \) by \( S \)
- \( q_H \) values for \( w \) have been defined for \( F \) and \( G \)

\[ | \Pr[S_2] - \Pr[S_1] | \leq (q_s + q_H)(q_F + q_G + q_s + q_H) / 2^{k_2} \]

RSA-PSS EF-CMA: Game 3

In the simulation of \( H \) in Game 2:

“choose a random \( u \in \mathbb{Z}_n \),
and compute \( y = (y^*)^b \cdot u^e \mod n \),
until the most significant bit is 0”

This may take a long time:
we limit it to \( k_2 \) iterations

This makes a difference,
only if \( y \) is still undefined after \( k_2 \) iterations:

\[ | \Pr[S_3] - \Pr[S_2] | \leq (q_s + q_H) / 2^{k_2} \]
RSA-PSS EF-CMA: Game 4

One now simulates the signing oracle $S$: Before simulating it, one stops the game if the signature of $m$ involves a pair $(m,r,b=1,*,*) \in \Lambda_H$ (already asked by $A$)

This may only happen if there is a collision on the value of $r$ between
- the $q_H$ possibly defined values
- the $q_S$ queries

$$| \Pr[S_4] - \Pr[S_3] | \leq q_S q_H / 2^{k_1}$$

RSA-PSS EF-CMA: Game 5

One can simulate the signing oracle $S$: Using the same $(m,r)$ as did $S$, by simulation of $H$: for some $(u,w)$,
- $(m,r,0,u,w) \in \Lambda_H$
- $0||w||s||t = y = u^e \mod n$
- $H(m,r)=w$, $F(w) = t$ and $r \oplus G(w) = s$

Thus $u$ is the signature.

The simulation is perfect:

$$\Pr[S_5] = \Pr[S_4]$$
RSA-PSS EF-CMA: Game 5

Adversary $A$

- $A_1(n,e) \rightarrow (m^*, \sigma^*)$ with permanent access to
  - the signing oracle $S$ simulation
  - the random oracles $F,G,H$ simulations

For any query $H(m,r)$ asked by $A$, there exists $(u,w)$ such that

$s(m,r,1,u,w) \in \Lambda_H$

$s \| w \| s \| t = y = y^* u^e \mod n$

$sH(m,r)=w$, $F(w) = t$ and $r \oplus G(w) = s$

Event $S_5$ (without chance)

$\Rightarrow (\sigma^*)^e = y = y^* u^e \mod n$

Thus $(\sigma^*/u)^e = y^* \mod n$

$\Pr[S_5] \leq \text{Succ}^{ow}(t_5) + 1/2^{k_2}$

RSA-PSS EF-CMA: Sum up

- $\Pr[S_0] = \text{Succ}^{ef-cma}(A)$
  \[ \Pr[S_1] = \Pr[S_0] \]

- $| \Pr[S_2] - \Pr[S_1] | \leq (qs + q_H)(q_F + q_G + qs + q_H)/2^{k_2}$

- $| \Pr[S_3] - \Pr[S_2] | \leq (qs + q_H) / 2^{k_2}$

- $| \Pr[S_4] - \Pr[S_3] | \leq qs q_H / 2^{k_1}$

- $\Pr[S_5] \leq \text{Succ}^{ow}(t_5) + 1 / 2^{k_2}$

\[
\text{Succ}^{ef-cma}(t) \leq \frac{qs + q_H}{2^{k_2}}(qs + q_F + q_G + q_H + 1) + \frac{qs q_H}{2^{k_1}} + \text{Succ}^{ow}(t_5) + \frac{1}{2^{k_2}}
\]
Comments: RSA-PSS

Security bound: $2^{75}$, and $2^{55}$ hash queries and $2^{30}$ signing queries

If one can break the scheme within time $T$, one can invert RSA within time $T' \leq T + (q_H + q_s)k_2T_f \leq T + 2^{65}T_f$

RSA-PSS

Security bound: $2^{75}$, and $2^{55}$ hash queries and $2^{30}$ signing queries

RSA (K bits) small exponent

If one can break the scheme within time $T$, one can invert RSA within time $T' \leq 2^{75} + 2^{65}K^2$

RSA:  
- 1024 bits $\rightarrow 2^{85}$ (NFS: $2^{80}$) $\times$
- 2048 bits $\rightarrow 2^{87}$ (NFS: $2^{111}$) $\checkmark$
- 4096 bits $\rightarrow 2^{89}$ (NFS: $2^{149}$) $\checkmark$
Jonsson’s Trick: Game 3

In the simulation of $H$ in Game 2:
“choose a random $u \in \mathbb{Z}_n$,
and compute $y = (y^*)^b \pmod{n}$,
until the most significant bit is 0”
Instead of limiting each simulation to $k_2$ iterations
we limit the global number to $2(q_s + q_H)$

One can show that some $y$ may not be defined,
but with probability $\leq 1 / 2^\ell$ : for any $\ell \leq (q_s + q_H)$
$\left| \Pr[S_3] - \Pr[S_2] \right| \leq 1 / 2^\ell$

Comments: RSA-PSS

Security bound: $2^{75}$, and
$2^{55}$ hash queries and $2^{30}$ signing queries
If one can break the scheme within time $T$,
one can invert RSA within time
$T' \leq T + 2(q_H + q_s)T_f \leq T + 2^{56}T_f$
RSA-PSS: Practical Security

Security bound: $2^{75}$, and $2^{55}$ hash queries and $2^{30}$ signing queries

RSA ($K$ bits) small exponent

If one can break the scheme within time $T$, one can invert RSA within time

$$T' \leq 2^{75} + 2^{56} K^2$$

RSA:
- 1024 bits $\rightarrow 2^{76}$ (NFS: $2^{80}$)
- 2048 bits $\rightarrow 2^{78}$ (NFS: $2^{111}$)
- 4096 bits $\rightarrow 2^{80}$ (NFS: $2^{149}$)

Summary

- Introduction
- Signature
  - FDH
  - PSS
  - Forking Lemma
  - Generic Model
- Conclusion
**Schnorr Signature (1989)**

\[ G, g \text{ and } q: \text{common elements} \]

\[ x: \text{private key} \]

\[ y = g^x: \text{public key} \]

**Signing** \( m \):

choose \( k \in \mathbb{Z}_q \) and compute \( r = g^k \)
as well as \( e = H(m, r) \)
and \( s = k - xe \mod q \)

\[ \sigma = (e, s) \]

**Verifying** \((m, \sigma)\):

\[ u = g^s y^e \quad ( = g^{k-xe} g^{xe} ) \]

\[ \text{test if } e = H(m, u) \]

---

**Security Proof** Pointcheval-Stern ‘96

**Existential Forgery = DL problem**

**Idea:** forking lemma

\[ A \xrightarrow{H(m, r)} e \xrightarrow{(e, s)} \] \[ e', (e', s') \]

\[ g^s y^e = r = g^{s'} y^{e'} \]

\[ g^{s-s'} = y^{e'-e} \]

Let \( \alpha = (s - s')/(e' - e) \mod q \)

Then \( y = g^\alpha \)
A asks $q_H$ queries $(m_i, r_i): h_i = H(m_i, r_i)$ and outputs $(m^*, r^*, e^*, s^*)$ such that

- $m^* = m_j$
- $e^* = H(m^*, r^*)$
- $V(m^*, r^*, e^*, s^*) = 1$

with probability $\varepsilon$

- $\varepsilon = \Pr[\text{Success}]$
- $\varepsilon_i = \Pr[\text{Success} \land m^* = m_i]$
- $\varepsilon = \sum \varepsilon_i$

For any $i$, one defines

$\Omega = \{(\omega, h_1, \ldots, h_{i-1}, h_i, \ldots h_{qH})\} = X_i \times Y_i$

- $x_i = (\omega, h_1, \ldots, h_{i-1})$
- $y_i = (h_i, \ldots h_{qH})$

$\varepsilon_i = \Pr_{X_i \times Y_i} [\text{Success} \land m^* = m_i]$

$Z_i = \left\{ x_i \in X_i \mid \Pr_{Y_i} [\text{Success} \land m^* = m_i] \geq \alpha_i \right\}$
Splitting Lemma

Assume $\Pr[x_i \in Z_i] < \varepsilon_i - \alpha_i$

$$\varepsilon_i = \Pr[S_i] = \Pr[S_i | x_i \in Z_i] \Pr[x_i \in Z_i] + \Pr[S_i | x_i \not\in Z_i] \Pr[x_i \not\in Z_i]$$

$$< 1 \times (\varepsilon_i - \alpha_i) + \alpha_i \times 1 = \varepsilon_i$$

$$\Pr[x_i \in X_i | S_i] \geq 1 - \frac{\alpha_i}{\varepsilon_i}$$

\[
\Pr[X_i | S_i] = 1 - \Pr[x_i \not\in X_i | \neg S_i]
\]

$$\geq 1 - \alpha_i \times 1 / \varepsilon_i$$

Forking Lemma - 3

- Run $A$ once: for any $i$
  - success and $m^* = m_i$ with probability greater than $\varepsilon_i$
  - $x_i \in Z_i$ with probability greater than $1 - \alpha_i / \varepsilon_i$

- Run $A$ a second time with same $x_i$ but random $y_i$
  - new success with probability greater than $\alpha_i$

$$p = \sum \varepsilon_i \times \left(1 - \frac{\alpha_i}{\varepsilon_i}\right) \times \alpha_i = \sum (\varepsilon_i - \alpha_i) \times \alpha_i$$
Forking Lemma - 4

With $\alpha_i = \rho \varepsilon_i$

\[
p = \sum \varepsilon_i \times \left(1 - \frac{\alpha_i}{\varepsilon_i}\right) \times \alpha_i = \sum (\varepsilon_i - \alpha_i) \times \alpha_i
\]
\[
= \sum (\varepsilon_i^2 (1 - \rho) \times \rho) = (1 - \rho) \rho \times \sum \varepsilon_i^2
\]
\[
\geq (1 - \rho) \rho \times (\sum \varepsilon_i)^2 / q_H = (1 - \rho) \rho \times \varepsilon^2 / q_H
\]

Optimal for $\rho = 1/2 : p \geq \varepsilon^2 / 4 q_H$

---

Forking Lemma: Result

- Run $A$ once with random $(\omega, h_1, \ldots, h_{i-1}, h_i, \ldots, h_{q_H}) = (x_i, y_i)$
- In case of success:
  run $A$ again with same $x_i$ but random $y'_i$
- One gets two successes $(m_1, r_1, e_1, s_1)$ and $(m_2, r_2, e_2, s_2)$ such that
  \[
  (m_1, r_1) = (m_2, r_2)
  \]
  \[
  \mathbf{V}(m_1, r_1, e_1, s_1) = 1 \text{ and } \mathbf{V}(m_2, r_2, e_2, s_2) = 1
  \]
  with probability greater than $\varepsilon^2 / 4 q_H$
Forking Lemma - Improvement

- Run $A$ until one gets a success:
on average $= 1/\varepsilon$ iterations: for any $i$
  - $m^* = m_i$ with prob greater than $\Pr[S_i \mid S] \geq \varepsilon_i / \varepsilon$
  - $x_i \in Z_i$ with probability greater than $1 - \alpha_i / \varepsilon_i$
- Run $A$ again with same $x_i$, but random $y_i$
  until a success: on average $1 / \alpha_i$ times
- On average:

$$T = \frac{1}{\varepsilon} + \sum \frac{\varepsilon_i}{\varepsilon} \times \left(1 - \frac{\alpha_i}{\varepsilon_i}\right) \times \frac{1}{\alpha_i} = \frac{1}{\varepsilon} + \frac{1}{\varepsilon} \times \sum \frac{1 - \rho}{\rho} = \frac{q_H + 1}{\varepsilon}$$

Comments: Forking Lemma

Security bound: $2^{75}$, and $2^{55}$ hash queries
If one can break the scheme within time $T = t/\varepsilon$,
one can extract two tuples within time
$$T' \leq q_H t/\varepsilon = q_H T \leq 2^{130}$$
This is not a practical result:
4096 bit moduli are required
Chosen-Message Attacks

The random oracle provides an easy simulation of the signing oracle.

The forking lemma applies to:

- Fiat-Shamir
- Guillou-Quisquater
- Schnorr
- …

---

Summary

- Introduction
- Signature
  - FDH
  - PSS
  - Forking Lemma
  - Generic Model
- Conclusion
Generic Model: ECDSA

\[ G = \langle P \rangle \text{ and } q: \text{common elements} \]
\[ x: \text{private key} \quad Y = x.P: \text{public key} \]

Signing \( m \):
choose \( k \in \mathbb{Z}_q \) and compute \( R = k.P \)
as well as \( r = f(R) \) and \( e = H(m) \)
and \( s = (e + xr)/k \mod q \)

\[ \sigma = (r, s) \]

Verifying \( (m, r, s) \): first \( 0 < r, s < q \)
\[ R' = e s^{-1}.P + r s^{-1}.Y \]
test if \( r = f(R') \)

Non-Malleability: ECDSA

Under some assumptions about the function \( f \) and the hash function \( H \), one can show

*In the generic model,*
*one cannot break non-malleability of ECDSA with probability significantly greater than*

\[ (n+1)(n+q_s+1)/2q \]

- \( q_s \) is the number of signing queries
- \( n \) is the number of group law operations
Malleability: ECDSA

In the description of ECDSA: \( f(R) = x_R \) (the first coordinate of \( R \))

Thus \( f(-R) = f(R) \)

If \((m,r,s)\) is a valid signature:

\[
0 < r, s < q \text{ and } f(e s^{-1}P + r s^{-1}Y) = r
\]

Then \((m,r,q-s)\) is a valid signature too:

\[
s' = -s \mod q \text{ and } 0 < r, s' < q
\]

\[
f(e s'^{-1}P + r s'^{-1}Y) = f(-e s^{-1}P - r s^{-1}Y)
= f(e s^{-1}P + r s^{-1}Y) = r
\]

Comments: ECDSA

However, this function \( f \) satisfies the requirements of the security theorem!

\( \Rightarrow \) The problem comes from the generic model

Indeed, when one knows \( E(P) \), one usually knows \( E(-P) \):

they are not independent

Thus \( f(R) \) and \( f(-R) \) are not independent!

If \( f \) random oracle: provably secure relative to DL in the random oracle model only (KCDSA)
The generic model should thus be used with care: automorphisms in the group may break the genericity of the encoding.

Summary

- Introduction
- Signature
  - FDH
  - PSS
  - Forking Lemma
  - Generic Model
- Conclusion
Generic Constructions

FDH: trapdoor OW permutation
Bad reduction to EF-CMA: $T' \approx q_H T$
  – If many-to-one function: SO-CMA only
  – If random self-reducibility (RSR):
    better reduction: $T' \approx q_s T$

PSS: RSR trapdoor OW permutation
Tight reduction: $T' \approx T$ \textit{practical security}

Forking lemma: identification scheme
secure against passive attacks
Bad reduction: $T' \approx q_H T$

Ideal Models

Ideal models to be handled with care
  – Random oracle model:
    seems correct in practice
  – Generic model: less convincing