Secure Design of Discrete Log Signature Schemes

Jacques Stern
Jacques.Stern@ens.fr

Département d'Informatique
École Normale Supérieure
45, rue d'Ulm
75230 PARIS CEDEX 05

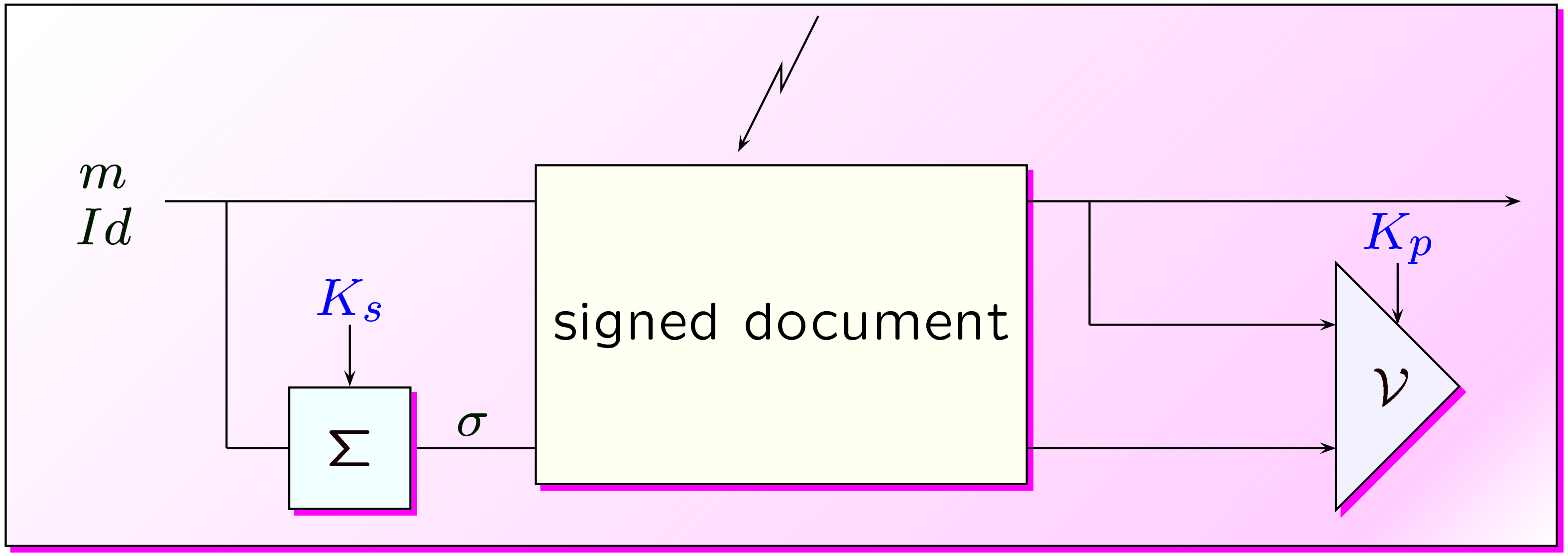
joint work with David Pointcheval,
Guillaume Poupard and David Naccache

Summary

References

- Pointcheval–Stern, Eurocrypt 96 (forking lemma)
- Pointcheval–Stern, Asiacrypt 96 (blind signatures)
- Poupard–Stern, Eurocrypt 98 (on-the-fly signatures)
- Naccache–Stern, work in progress (message recovery)

Signature schemes

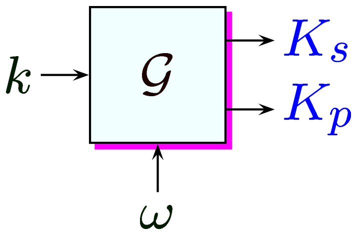
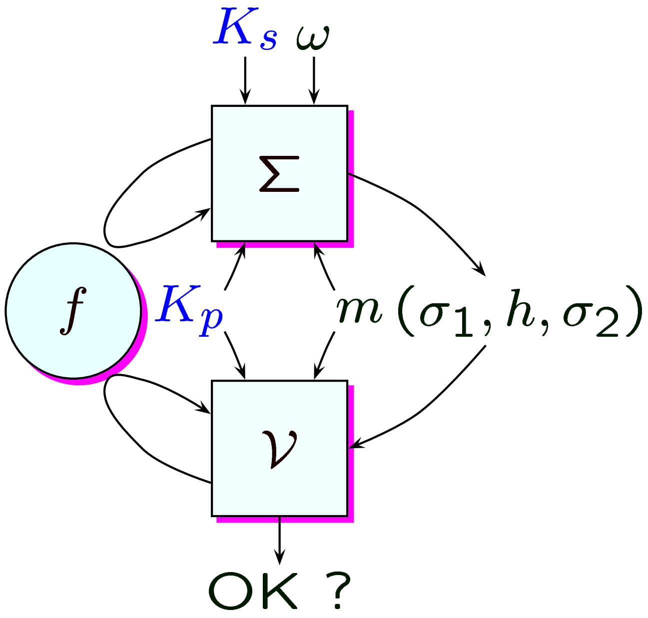


Proof of origin of a message.

Security

No one can forge a valid pair (m, σ) = no existential forgery

The framework

<table>
<tr><th>Key generation</th><th>Signature and Verification</th></tr>
<tr><td>k is the security parameter

$n = |K_p|$</td><td></td></tr>
</table>

- $\mathcal{G}$ and Σ are probabilistic algorithms : random tape ω
- $\mathcal{V}$ is deterministic
- f is a hash function

Example 1 : RSA

$\mathcal{G}$: $N = pq$ such that $|N| = n$
public key : N, e, with e prime to $\phi(N)$
secret key : d such that $ed = 1 \bmod \phi(N)$

Σ
$\sigma_1 = \emptyset$
$h = f(m, \sigma_1)$
$\sigma_2 = h^d \bmod N$

Signature : $\left(m, \sigma_2\right)$

$\mathcal{V}$: $(\sigma_2)^e \stackrel{?}{=} f(m) \bmod N$

Example 2 : DSA

$\mathcal{G}$: p, r prime with $r|(p-1)$ and g of order r in $(\mathbb{Z}/p\mathbb{Z})^\star$
secret key : $s \in_R [1,(r-1)]$
public key : $w = g^s \bmod p$

Σ : $u \in_R [1,(r-1)]$
$\sigma_1 = v = g^u \bmod p$
$c = v \bmod r$; if $c = 0$ abort
$h = f(m)$
solve $ud = h + sc \bmod r$; if $d = \sigma_2 = 0$ abort

Signature : (m, c, d)

$\mathcal{V}$: $c,\ d \in [1,(r-1)]$
$(g^{\frac{f(m)}{d} \bmod r} w^{\frac{c}{d} \bmod r} \bmod p) = c \bmod r$

Example 3 : ECDSA

$\mathcal{G}$: an elliptic curve E, a prime r, an element G of E of order r
secret key : $s \in_R [1,(r-1)]$
public key : $W = s.G$

Σ : $u \in_R [1,(r-1)]$
$V = u.G$
turn V into an integer $i = R(V)$
$\sigma_1 = c = i \bmod r$; if $c = 0$ abort
$h = f(m)$
solve $ud = h + sc \bmod r$; if $\sigma_2 = d = 0$ abort

Signature : (m, c, d)

$\mathcal{V}$: $c,\ d \in [1,(r-1)]$
$R((\frac{f(m)}{d} \bmod r).G + (\frac{c}{d} \bmod r).W) = c \bmod r$

A group-theoretic framework

$\mathcal{G}$: a group E, a prime r, an element G of E of order r
secret key : $s \in_R [1,(r-1)]$
public key : $W = s.G$

Σ : similar to ECDSA using a function R
which encodes an element V of E into an integer i
Signature : (m,c,d)

$\mathcal{V}$: similar to ECDSA

The random oracle model

- Σ and $\mathcal{V}$ both use a hash function f with $f \in_R \{0,1\}^{\ell} \rightarrow \{0,1\}^{k}$, seen as a random oracle. (refer to Bellare & Rogaway ACM CCCS'93)

  $\longrightarrow$ validates cryptodesign
  $\longrightarrow$ spots design errors

- $k(n) \gg \log n$

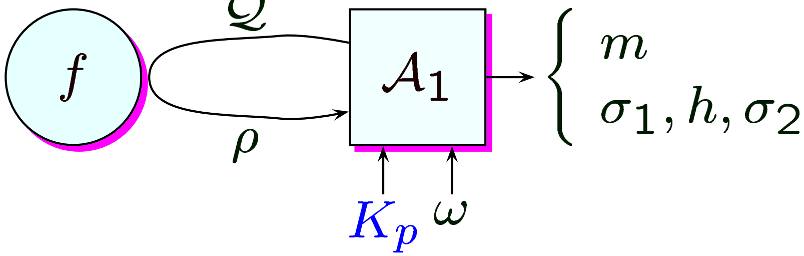
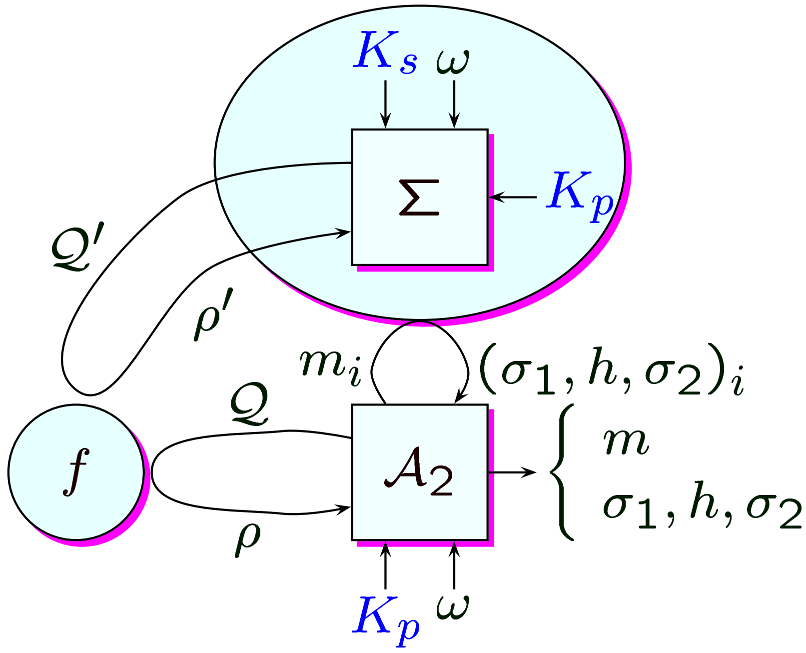
- Existential forgery :

  there is an attacker $\mathcal{A}$ which outputs proper signatures with probability $\varepsilon \geq \frac{1}{poly(n)}$ for infinitely many n's

Attacks

We only consider

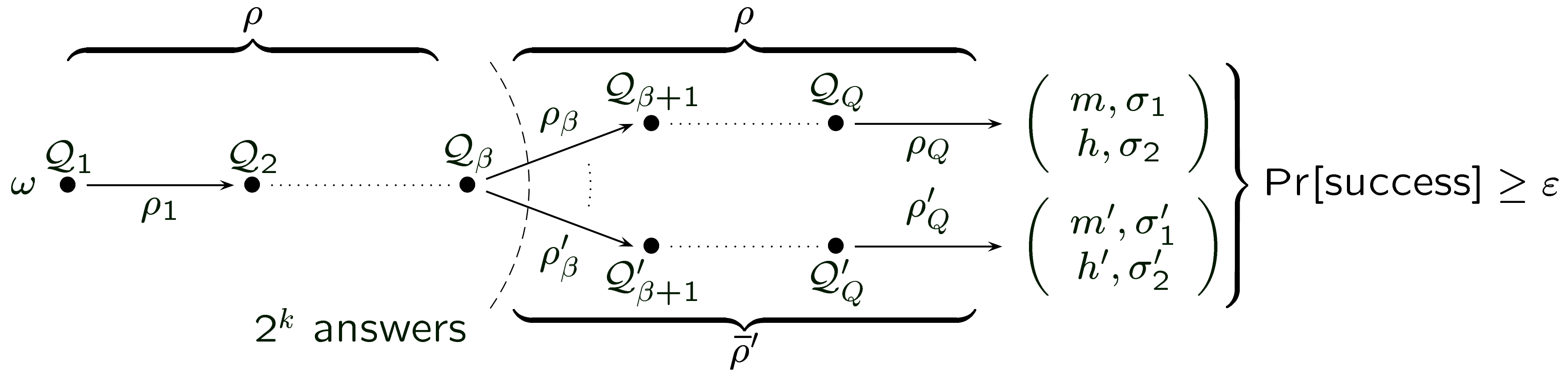
- No-message attacks
- Adaptively chosen message attacks

Attack I no-message attack	**Attack II** adaptively chosen message attack
	

Motivation

To provide proofs of security for signature schemes relatively to well-established difficult problems :
Existential forgery under such attacks allows to solve difficult problems.

The forking lemma



$\mathcal{A}$ is an attacker with probability of success, over ω, f and possibly K_p, greater that ε.

Oracle replay :

- play the attack with random ω and f
- select β at random
- replay the attack with the same ω and same $\beta - 1$ first answers, others are given at random

Conditioning

Let $A \subset X \times Y$ such that Pr$[A(x,y)] \geq \varepsilon$
Then there exists $U \subset X$ such that

- Pr$[x \in U] \geq \dfrac{\varepsilon}{2}$
- whenever $a \in U$, Pr$[A(a,y)] \geq \dfrac{\varepsilon}{2}$

- there is a β such that Pr[success and output queried at β] $\geq \varepsilon/Q$
- using the previous lemma, we get a set Ω such that
  - Pr$[(\omega,\rho) \in \Omega] \geq \varepsilon/2Q$
  - whenever $(\omega,\rho) \in \Omega$, Pr$_{\bar{\rho}}$[success and output queried at β] $\geq \varepsilon/2Q$

Using the forking lemma

With non-negligible probability, one gets index β and $(\omega, \rho) \in \Omega$

Then, with random choice of $\bar{\rho}$ and $\bar{\rho}'$, with non-negligible probability :

- with answers $(\rho, \bar{\rho})$, the attacker outputs $(m, \sigma_1, h, \sigma_2)$
- with answers $(\rho, \bar{\rho}')$, the attacker outputs $(m', \sigma_1', h', \sigma_2')$,

- when both m and σ_1 are submitted to the oracle i.e. $h = f(m, \sigma_1)$ $m = m'$ and $\sigma_1 = \sigma_1'$
- With probability larger than $1 - 2^{-k(n)}$, $h \neq h'$
- Using verifying equation, infer information on secret data.

El Gamal

$\mathcal{G}$: p prime, and g generator of $(\mathbb{Z}/p\mathbb{Z})^\star$
secret key : $x \in_R (\mathbb{Z}/(p-1)\mathbb{Z})^\star$
public key : $y = g^x \bmod p$

Σ : $k \in_R (\mathbb{Z}/(p-1)\mathbb{Z})^\star$
$r = g^k \bmod p$
solve $m = xr + ks \bmod (p-1)$

Signature : (m, r, s)

$\mathcal{V}$: $g^m \stackrel{?}{=} y^r r^s \bmod p$

Existential forgery

$$\begin{array}{ll} \text{choose} & e \in \mathbb{Z}/(p-1)\mathbb{Z} \\ & v = (\mathbb{Z}/(p-1)\mathbb{Z})^\star \\ \text{let} & r = g^e y^v \bmod p \\ & s = -rv^{-1} \bmod (p-1) \end{array}$$

(r, s) is a valid signature of the message

$$m = es \bmod (p-1)$$

Modified El Gamal Signature

$\mathcal{G}$: p prime, and g generator of $(\mathbb{Z}/p\mathbb{Z})^\star$
secret key : $x \in_R (\mathbb{Z}/(p-1)\mathbb{Z})^\star$
public key : $y = g^x \bmod p$

Σ : $k \in_R (\mathbb{Z}/(p-1)\mathbb{Z})^\star$
$r = g^k \bmod p$
solve $f(m,r) = xr + ks \bmod (p-1)$

Signature : $(m, r, f(m,r), s)$

$\mathcal{V}$: $g^{f(m,r)} \stackrel{?}{=} y^r r^s \bmod p$

First Result

For fixed α, an α-hard prime p is a prime p such that $p-1 = QR$ with Q prime and $R \leq |p|^{\alpha}$.

Existential forgery of the Modified El Gamal signature scheme, under a no-message attack, is equivalent to discrete logarithms with α-hard primes.

Proof

By the forking lemma, we get (m, r, h, s) and (m, r, h', s') such that

$$h \neq h' \text{ and } \begin{cases} g^h & = & y^r r^s \bmod p \\ g^{h'} & = & y^r r^{s'} \bmod p \end{cases}$$

$$g^{hs'-h's} = y^{r(s'-s)} \bmod p$$
$$g^{h-h'} = r^{s-s'} \bmod p$$

There are x and t such that $y = g^x$ and $r = g^t$, so

$$hs' - h's = xr(s' - s) \bmod (p-1)$$
$$h - h' = t(s - s') \bmod (p-1)$$

Proof (cont.)

h and h' come from the random oracle, we may assume
$h - h'$ prime to Q hence $s - s'$ prime to Q.

1. r also prime to Q $\Longrightarrow$ $x \mod Q$ $\Longrightarrow$ x
2. $r = bQ$ with b small $\Longrightarrow$ $t \mod Q$ $\Longrightarrow$ t

1. $\Pr[\mathcal{M}(g, y) \to x] \geq \frac{1}{poly(n)}$ $\Longrightarrow$ OK
2. $\Pr[\mathcal{M}(g, y) \to (b, t)] \geq \frac{1}{poly(n)}$ $\Longrightarrow$ bad case

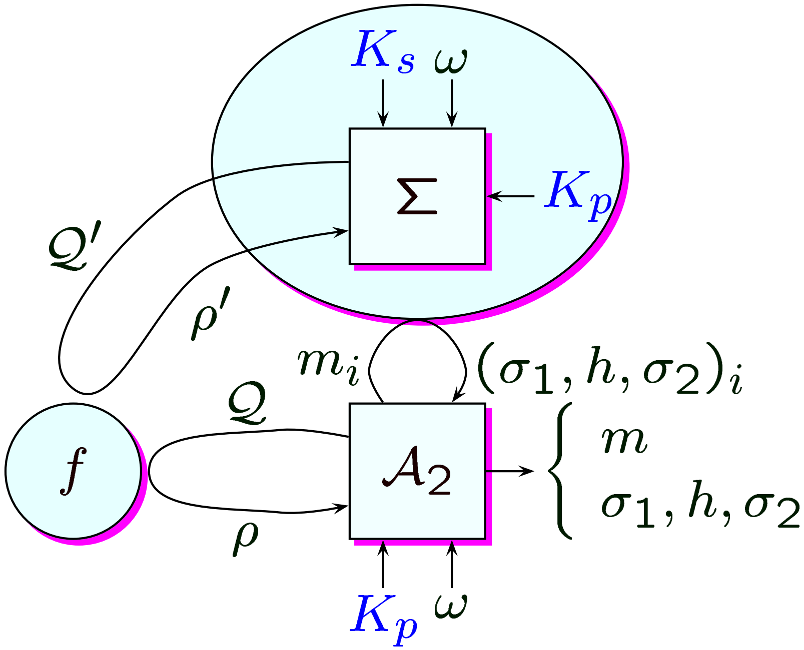
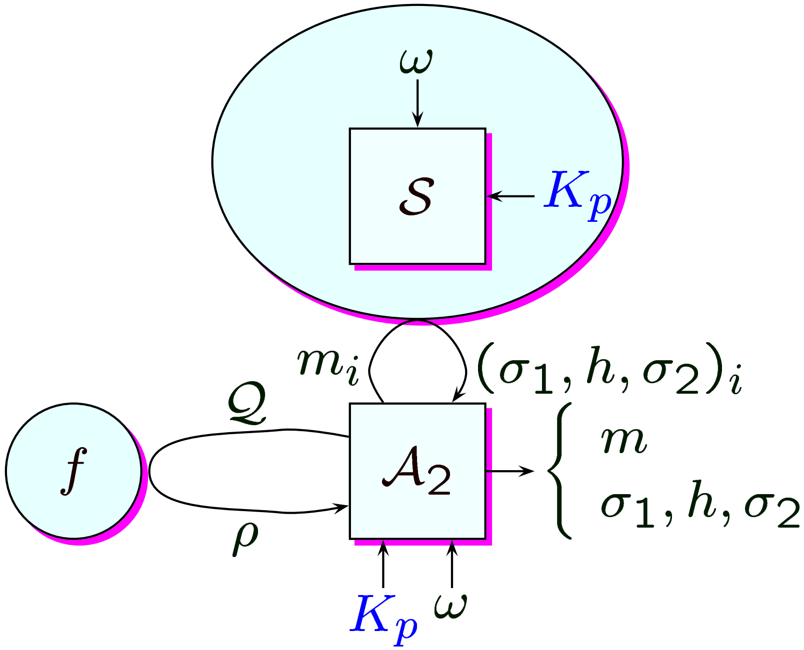
End of proof

- Since b is small, we treat it as constant b_0, by reducing the success probability.
- Bad case produces DL of fixed $b_0 Q$ in base g.
- By having g vary as well, this solves DL problem

Note that proof assumes that g is random.

This matches up with an attack described by Bleichenbacher at Eurocrypt 96, which uses specific choice for g.

Adaptively Chosen Message Attack

Attacker II + Signer (Σ)	Attacker II + Simulator ($\mathcal{S}$)
	

We suppose $f(m_i, (\sigma_1)_i) = h_i \ \forall i$

Simulation

1. Attacker and Oracle can collude to produce signatures that cannot be distinguished from regular signatures.
2. Uses the freedom to provide answers to the oracle queries
3. Simulation mimicks existential forgery.

Main Result

In the random oracle model, existential forgery of the Modified El Gamal signature scheme for an adaptively chosen message attack allows to solve the discrete logarithm problem with α-hard primes.

Blind Signatures

an authority helps a user to get a valid signature

the message and the signature
must remain unknown for the authority

Schnorr Blind Scheme

Bank		Alice
public : p, q, g, y		
secret : $y = g^{-x} \bmod p$		
$k \in (\mathbb{Z}/q\mathbb{Z})^\star$		
$r = g^k \bmod p$	$\xrightarrow{\quad r \quad}$	
		$\alpha, \beta \in \mathbb{Z}/q\mathbb{Z}$
		$r' = r g^\alpha y^\beta \bmod p$
		$e' = H(m, r')$
	$\xleftarrow{\quad e \quad}$	$e = e' - \beta \bmod q$
$s = k + ex \bmod q$	$\xrightarrow{\quad s \quad}$	
		$g^s y^e \stackrel{?}{=} r \bmod p$
		$s' = s + \alpha \bmod q$

(r', s') is an unknown signature
of the unknown message m.

Security Properties

- $(\ell, \ell+1)$-forgery : after ℓ interactions with the Bank the attacker can forge $\ell+1$ message–signature valid pairs.
- One-more forgery : an $(\ell, \ell+1)$-forgery for some integer ℓ.

Attacks

- sequential attack : the attacker interacts sequentially with the signer.
- parallel attack : the attacker can initiate several interactions at the same time with the signer.

Okamoto–Schnorr Blind Scheme

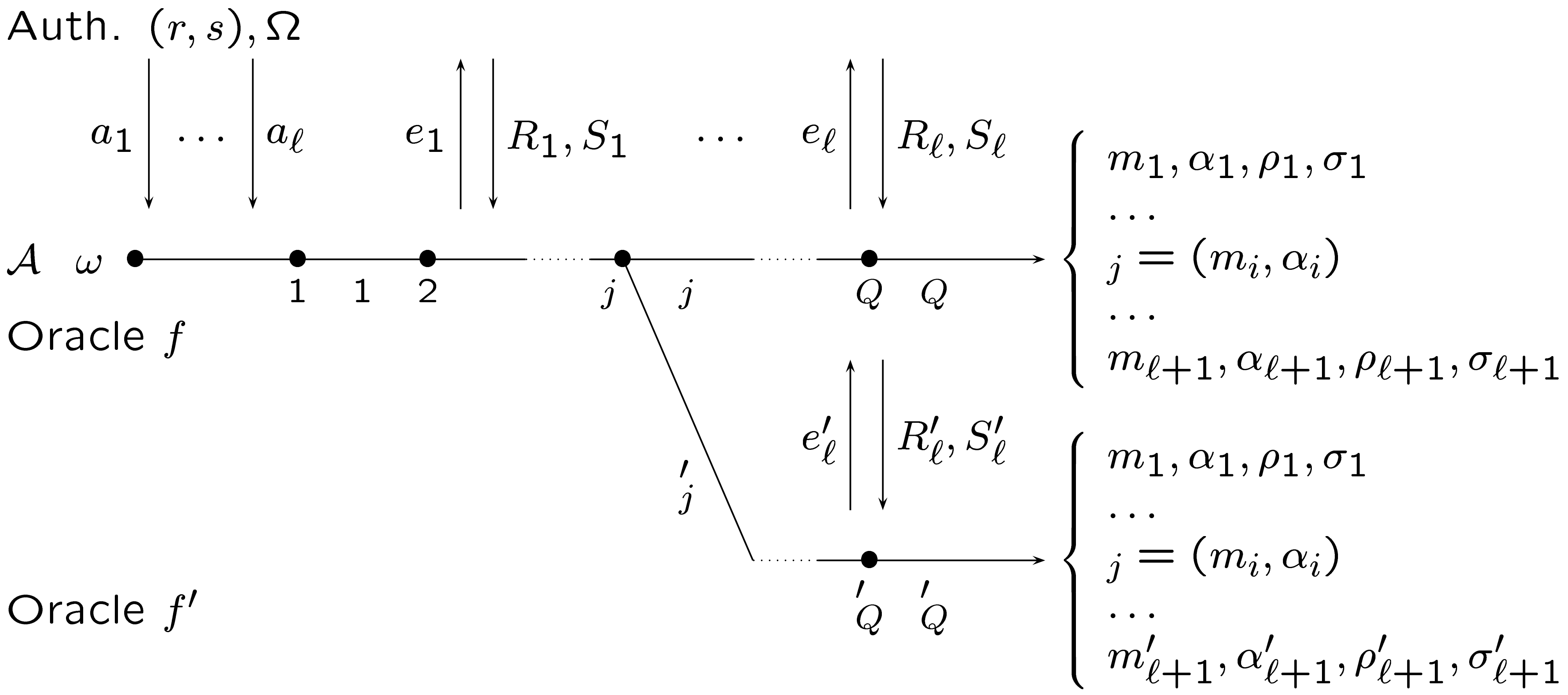
Authority		Alice
public : p, q, g, h, y secret : $y = g^{-r}h^{-s}$ mod p $t, u \in (\mathbb{Z}/q\mathbb{Z})^\star$ $a = g^t h^u$ mod p	$\xrightarrow{\quad a \quad}$	
		$\beta, \gamma, \delta \in \mathbb{Z}/q\mathbb{Z}$ $\alpha = ag^\beta h^\gamma y^\delta$ mod p $\varepsilon = H(m, \alpha)$ $e = \varepsilon - \delta$ mod q
	$\xleftarrow{\quad e \quad}$	
$R = t + er$ mod q $S = u + es$ mod q	$\xrightarrow{\quad R, S \quad}$	
		$g^R h^S y^e \stackrel{?}{=} a$ mod p $\rho = R + \beta$ mod q $\sigma = S + \gamma$ mod q

$(m, \alpha, \varepsilon, \rho, \sigma)$ s.t. $\alpha = g^\rho h^\sigma y^\varepsilon$ mod p with $\varepsilon = H(m, \alpha)$.

Main Result

If there exists a Probabilistic Polynomial Turing Machine
which can perform a one-more forgery,
with non-negligible probability,
even under a parallel attack,
then the Discrete Logarithm Problem
can be solved in Polynomial Time.

Forking Lemma



Using the forking Lemma

We play the attack with random (r,s), Ω, ω and f and replay to get a fork.

We get ith non-negligible probability,

$$\begin{aligned} \alpha_i &= g^{\rho_i} h^{\sigma_i} y^{\varepsilon_i} \bmod p \\ &= g^{\rho_i - r\varepsilon_i} h^{\sigma_i - s\varepsilon_i} \bmod p. \\ &= g^{\rho'_i - r\varepsilon'_i} h^{\sigma'_i - s\varepsilon'_i} \bmod p. \end{aligned}$$

This provides DL of h in base g provided the random variable $\chi_i = \rho_i - r\varepsilon_i \bmod q$. is sensitive to queries of index $> \beta$. This (technical) part uses the so-called witness-indistinguishability property.

The GPS "on-the-fly" signature scheme

$\mathcal{G}$: $N = pq$, g of maximal order $\lambda(N)$ in $(\mathbb{Z}/N\mathbb{Z})^\star$
secret key : $s \in_R [0, (S-1)]$
public key : $v = g^s \bmod p$

Σ : $x \in_R [0, (X-1)]$
$\sigma_1 = r = g^x \bmod N$
$h = c = f(m, r)$
solve $\sigma_2 = y = x + sc \bmod r$

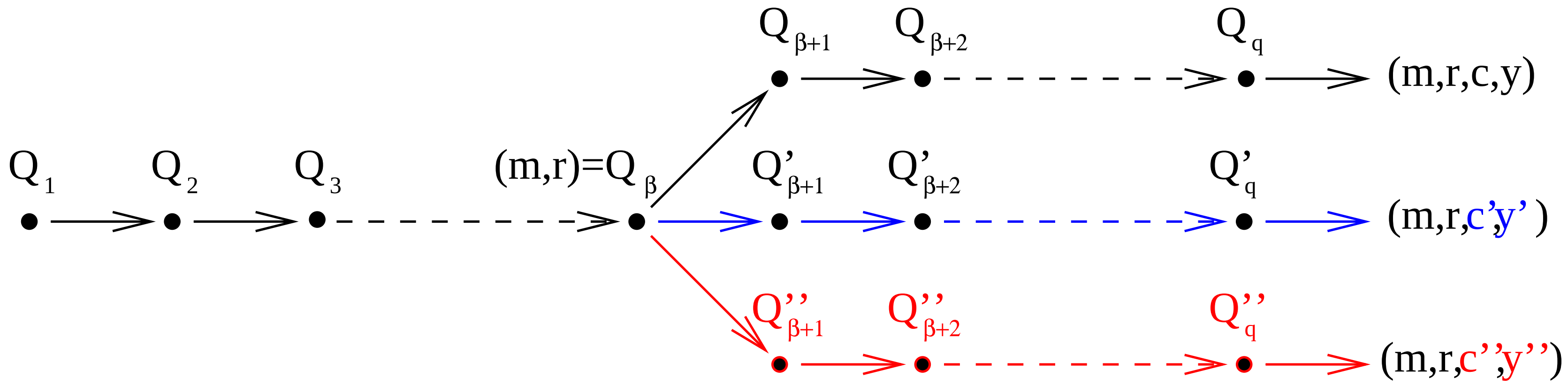
Signature : (m, c, y)

$\mathcal{V}$: $y \in [1, (X-1)]$, $c \in [0, K-1]$, with $K = 2^k$
$f(m, g^y v^c \bmod N) = c$

Security of the signature scheme

Assume $X \gg KS$. If an existential forgery of the signature scheme under adaptatively chosen message attack has a non-negligible probability of success then the discrete logarithm problem modulo N with logarithms in the range $[0, S]$ can be solved in time polynomial in $|N|$.

The Forking Lemma



Given any positive integer B,

$$\Pr_{0 \leq c', c'' < K} \{\mathrm{gcd}(c - c', c - c'') > B\} < \frac{2.7}{B}$$

1. use $\mathcal{M}$ with the public key v

$$\Longrightarrow s_0 = s \bmod \frac{\lambda(N)}{\mathrm{gcd}(c - c', c - c'', \lambda(N))}$$

2. when $g^{s_0} = v \bmod N$, the algorithm is successful (good case)
3. if bad cases appear too often, then using $\mathcal{M}$ with $v_0 = g^{\gamma} \bmod n$, $\gamma \in_R [0, (S-1)]$

$$\Longrightarrow \lambda(N)$$

4. once $\lambda(N)$ is known, step 1 yields s as denominator is small

Message recovery : The NR scheme

$\mathcal{G}$: a group E, an encoding R of the elements of E into integers, a prime r
an element G of E of order r
secret key : $s \in_R [1,(r-1)]$
public key : $W = s.G$

Σ : $u \in_R [1,(r-1)]$
$V = u.G$
turn V into an integer $i = R(V)$
$\sigma_1 = c = i + m \bmod r$; if $c = 0$ abort
$h = f(m)$
solve $d = u - sc \bmod r$

Signature : (c, d)

$\mathcal{V}$: $c \in [1,(r-1)]$, $d \in [0,(r-1)]$
$m = c - R(d.G + c.W)$; check redundancy of m

A scheme with "partial" recovery

$\mathcal{G}$: a group E, an encoding R of the elements of E into integers,
a prime r, an element G of E of order r
secret key : $s \in_R [1, (r-1)]$
public key : $W = s.G$

Σ : $u \in_R [1, (r-1)]$
$V = u.G$; turn V into an integer $i = R(V)$
split m into m_1, m_2 ; add redundancy to m_1 to obtain f_1
$\sigma_1 = c = i + f_1 \bmod r$; if $c = 0$ abort
$f_2 = f(m_2)$
solve $ud = f_2 + sc \bmod r$. If $d = 0$ abort

Signature : (m_2, c, d)

$\mathcal{V}$: $c \in [1, (r-1)]$, $d \in [1, (r-1)]$
$f_1 = c - R((\frac{f(m_2)}{d} \bmod r).G + (\frac{c}{d} \bmod r).W)$
check redundancy of f_1 and extract m_1

Why such a scheme

1. allows to transmit only a part of the message (typically ten bytes can be saved) ; crucial in some environments
2. contrary to NR, no size constraint on the message

Security of the scheme

- security proof assumes also R comes from a random oracle ; presumably "more convincing" for EC crypto
- uses the forking lemma, noting that
  $P = (\frac{f(m_2)}{d} \bmod r).G + (\frac{c}{d} \bmod r).W$ is
  queried from R-oracle and m_2 is queried from f-oracle
  where (m_2, c, d) is the output signature
- Case 1 : with significant probability, m_2 is queried before P ; fork at P to obtain $P = (f_2 d^{-1}).G + (cd^{-1}).W = (f_2' d'^{-1}).G + (c'd'^{-1}).W$ which produces DL of W in base G, unless relation is trivial.
- Case 2 : P is queried first ; fork at m_2 to obtain similar relation
- exceptional cases appear with probability ϵ, where ϵ is the probability that a random element f_2 has correct redundancy

Conclusion

The forking lemma provides conceptually simple proofs of security for

- classical schemes (El Gamal, blind signatures)
- schemes tailored to specific needs (“on-the-fly” signatures, signatures with “partial” recovery

It thus appears as a very useful tool in the design of EC-based signature schemes.