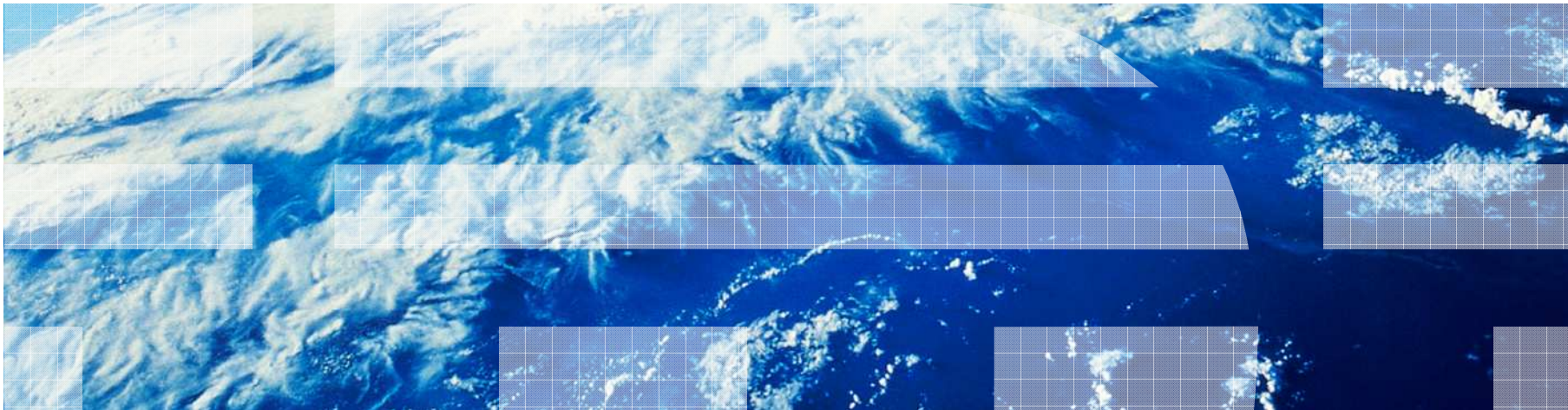


Gregory Neven, IBM Research – Zurich

ECRYPT II Summer School on Provable Security, Sept 7-11, 2009, Barcelona



Digital signatures II : Standard model



- Digital signatures I: Random oracle model
 - Signatures based on RSA:
RSA-FDH and the random oracle model
 - Signatures based on discrete logs:
Schnorr signatures and the forking lemma

- Digital signatures II: Standard model
 - **Signatures based on one-way functions:**
Lamport one-time signatures
 - Signatures based on strong RSA:
Cramer-Shoup signatures
 - Signatures with protocols:
Camenisch-Lysyanskaya signatures

- Intuitively
 - functions that are easy to compute but hard to invert
 - considered most basic primitive in cryptography
- Security model for one-way function $f : D \rightarrow R$



$$x' \leftarrow_{\$} D ; y \leftarrow f(x')$$

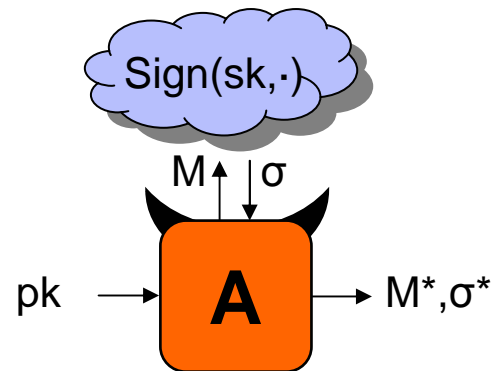
$$x \leftarrow_{\$} B(f,y)$$

$$\text{Avantage } \varepsilon = \Pr [y = f(x)]$$

Lamport one-time signature scheme



- One-time signature = only one signature per key pair



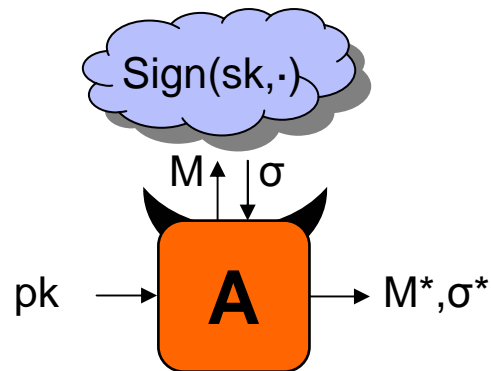
- Lamport one-time signature scheme

$$sk = \begin{pmatrix} x_{1,0} & x_{2,0} & \cdots & x_{n,0} \\ x_{1,1} & x_{2,1} & \cdots & x_{n,1} \end{pmatrix}$$

$$pk = \begin{pmatrix} y_{1,0} & y_{2,0} & \cdots & y_{n,0} \\ y_{1,1} & y_{2,1} & \cdots & y_{n,1} \end{pmatrix}$$

where $x_{i,j} \leftarrow_{\$} D$; $y_{i,j} \leftarrow f(x_{i,j})$

- One-time signature = only one signature per key pair



- Lamport one-time signature scheme

$$sk = \begin{pmatrix} x_{1,0} & x_{2,0} & \dots & x_{n,0} \\ x_{1,1} & x_{2,1} & \dots & x_{n,1} \end{pmatrix} \quad pk = \begin{pmatrix} y_{1,0} & y_{2,0} & \dots & y_{n,0} \\ y_{1,1} & y_{2,1} & \dots & y_{n,1} \end{pmatrix}$$

Sign: $\sigma \leftarrow (x_{1,M_1}, x_{2,M_2}, \dots, x_{n,M_n})$ where $M = M_1 M_2 \dots M_n \in \{0,1\}^n$

Verify: Check $f(\sigma_i) = y_{i,M_i}$ for $i=1, \dots, n$

- Extensions to multi-message signing possible using tree structure

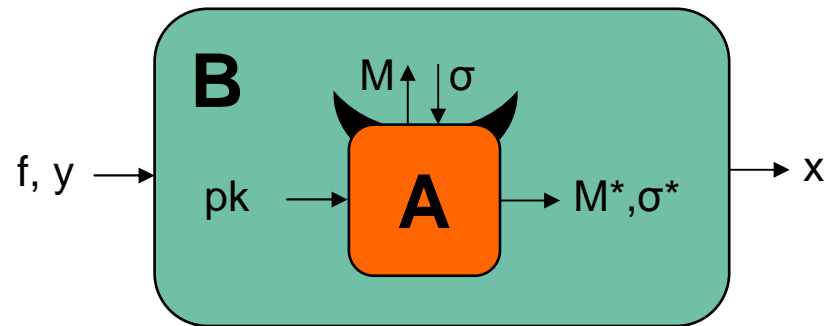
Security of Lamport signatures



Theorem:

If f is (t, ϵ) one-way,
 then Lamport signatures are $(t - 2nt_f, 2n\epsilon)$ unforgeable.

Proof:



B guesses position (i^*, j^*) in pk to embed own challenge y

$$sk = \begin{pmatrix} x_{1,0} & \dots & x_{i^*,0} & \dots & x_{n,0} \\ x_{1,1} & \dots & ? & \dots & x_{n,1} \end{pmatrix} \quad pk = \begin{pmatrix} y_{1,0} & \dots & y_{i^*,0} & \dots & y_{n,0} \\ y_{1,1} & \dots & y & \dots & y_{n,1} \end{pmatrix}$$

Sign(M): hope that $M[i^*] = 1 - j^*$ (probability = $1/2$)

Solve $f^{-1}(y)$: hope that $M^*[i^*] = j^*$ (probability $\geq 1/n$)

- Digital signatures I: Random oracle model
 - Signatures based on RSA:
RSA-FDH and the random oracle model
 - Signatures based on discrete logs:
Schnorr signatures and the forking lemma

- Digital signatures II: Standard model
 - Signatures based on one-way functions:
Lamport one-time signatures
 - **Signatures based on strong RSA:**
Cramer-Shoup signatures
 - Signatures with protocols:
Camenisch-Lysyanskaya signatures

Scheme	Assumption	Signature size (bits)	Remarks
[GHR99]	strong RSA	2432	prime-output chameleon hash
[CS99]	strong RSA	4352	collision-resistant hash optimizations exist
[CL02]	strong RSA	4992	efficient ZK protocols
[HK08]	strong RSA	2304	programmable hash functions
[HW09]	RSA	4096	chameleon hash
[BB04]	pairings (q-SDH)	512	
[W05]	pairings (CDH)	512	long public key (>65536 bits)
[HK08]	pairings (q-SDH)	356	programmable hash functions

Practical signatures in standard model



Scheme	Assumption	Signature size (bits)	Remarks
[GHR99]	strong RSA	2432	prime-output chameleon hash
[CS99]	strong RSA	4352	collision-resistant hash optimizations exist
[CL02]	strong RSA	4992	efficient ZK protocols
[HK08]	strong RSA	2304	programmable hash functions
[HW09]	RSA	4096	chameleon hash
[BB04]	pairings (q-SDH)	512	
[W05]	pairings (CDH)	512	long public key (>65536 bits)
[HK08]	pairings (q-SDH)	356	programmable hash functions

Kg:

Choose random primes p, p', q, q'
such that $p=2p'+1$, $q=2q'+1$

$N \leftarrow pq$

$h, x \leftarrow_{\$} QR_N$

Choose random 257-bit prime e'

$pk \leftarrow (N, e', h, x)$; $sk \leftarrow (p, q)$

Sign(sk, M):

Choose random 257-bit prime $e \neq e'$

$y' \leftarrow_{\$} QR_N$

$x' \leftarrow y'^{e'} / h^{H(M)} \bmod N$

or: $y'^{e'} = x' h^{H(M)} \bmod N$

$y \leftarrow (x h^{H(x')})^{1/e} \bmod N$

or: $y^e = x h^{H(x')} \bmod N$

$\sigma \leftarrow (e, y, y')$

Strong prime $p = 2p'+1$ where p, p' prime

QR_N = set of quadratic residues mod N

$N = pq$ where p, q strong primes

$\rightarrow \varphi(N) = 4p'q'$

$\rightarrow QR_N$ is cyclic subgroup of order $p'q'$

Vf(pk, M, σ):

Check e is odd, 257-bit, $e \neq e'$

$x' \leftarrow y'^{e'} / h^{H(M)} \bmod N$

Check $y^e = x h^{H(x')} \bmod N$

Kg:

Choose random primes p, p', q, q'
such that $p=2p'+1$, $q=2q'+1$

$N \leftarrow pq$

$h, x \leftarrow_{\$} QR_N$

Choose random 257-bit prime e'

$pk \leftarrow (N, e', h, x)$; $sk \leftarrow (p, q)$

Sign(sk, M):

Choose random 257-bit prime $e \neq e'$

$y' \leftarrow_{\$} QR_N$

$x' \leftarrow y'^{e'} / h^{H(M)} \bmod N$

or: $y'^{e'} = x' h^{H(M)} \bmod N$

$y \leftarrow (x h^{H(x')})^{1/e} \bmod N$

or: $y^e = x h^{H(x')} \bmod N$

$\sigma \leftarrow (e, y, y')$

Strong prime $p = 2p'+1$ where p, p' prime

QR_N = set of quadratic residues mod N

$N = pq$ where p, q strong primes

$\rightarrow \varphi(N) = 4p'q'$

$\rightarrow QR_N$ is cyclic subgroup of order $p'q'$

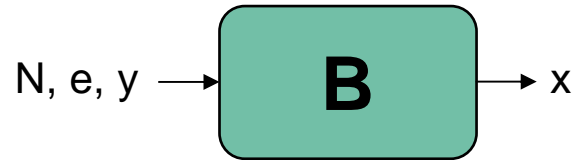
Vf(pk, M, σ):

Check e is odd, 257-bit, $e \neq e'$

$x' \leftarrow y'^{e'} / h^{H(M)} \bmod N$

Check $y^e = x h^{H(x')} \bmod N$

- RSA assumption

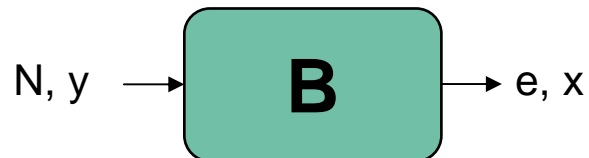


$$y \leftarrow_{\$} Z_N^*$$

$$x \leftarrow_{\$} B(N, e, y)$$

$$\text{Avantage } \varepsilon = \Pr [y = x^e \bmod N]$$

- Strong RSA assumption



$$y \leftarrow_{\$} Z_N^*$$

$$(e, x) \leftarrow_{\$} B(N, y)$$

$$\text{Avantage } \varepsilon = \Pr [y = x^e \bmod N \wedge e \neq 1]$$

Given $x, y \in \mathbb{Z}_N^*$, $a, b \in \mathbb{Z}$ such that $\gcd(a, b) = 1$ and $x^a = y^b \pmod{N}$
one can compute $w = y^{1/a} \pmod{N}$.

$\gcd(a, b) = 1 \rightarrow$ find $a', b' \in \mathbb{Z} : aa' + bb' = 1$ by extended Euclidean

$$x^{ab'} = y^{bb'} = y^{1-aa'} \pmod{N}$$

$$x^{ab'} y^{aa'} = y \pmod{N}$$

$$(x^{b'} y^{a'})^a = y \pmod{N}$$

$$w \leftarrow x^{b'} y^{a'} \pmod{N}$$

$pk = (N, e', x, h)$, $\sigma = (e, y, y')$ such that

- $y^e = x h^{H(x')}$ mod N
- $y'^{e'} = x' h^{H(M)}$ mod N

- Let i^{th} signing query be $M_i \rightarrow \sigma_i = (e_i, y_i, y'_i)$, let $x'_i \leftarrow y_i^{e_i} h^{-H(M)}$
- Let forgery be M^* , $\sigma^* = (e^*, y^*, y'^*)$, let $x'^* \leftarrow (y'^*)^{e^*} h^{-H(M^*)}$

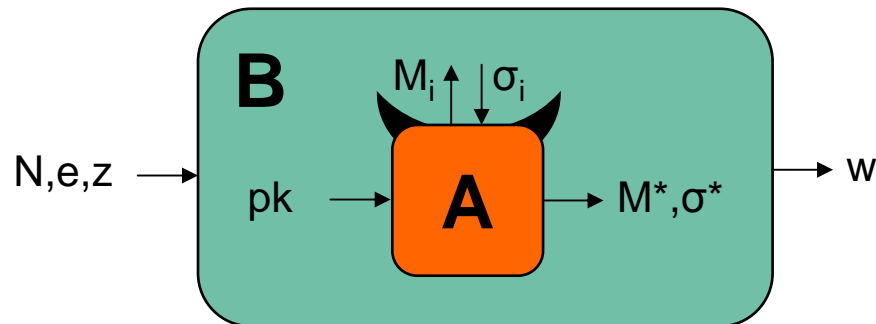
- Type-I forgery: For some $1 \leq j \leq q_S$: $e^* = e_j$ and $x'^* = x'_j$
- Type-II forgery: For some $1 \leq j \leq q_S$: $e^* = e_j$ and $x'^* \neq x'_j$
- Type-III forgery: For all $1 \leq i \leq q_S$: $e^* \neq e_i$

Type-I forgery

$pk = (N, e', x, h)$, $\sigma = (e, y, y')$ such that

- $y^e = x h^{H(x')} \bmod N$
- $y'^{e'} = x' h^{H(M)} \bmod N$

Type-I forgery: For some $1 \leq j \leq q_S$: $e^* = e_j$ and $x'^* = x'_j$
→ break one-wayness of RSA



$pk = (N, e', x, h)$, $\sigma = (e, y, y')$ such that

- $y^e = x h^{H(x')} \pmod N$
- $y'^{e'} = x' h^{H(M)} \pmod N$

Type-I forgery: For some $1 \leq j \leq q_S$: $e^* = e_j$ and $x'^* = x'_j$

→ break one-wayness of RSA: given (N, e, z) compute w such that $w^e = z$

Let $E = \prod_i e_i$ where e_i random primes

Simulate pk :

$$e' \leftarrow e ; r \leftarrow_{\$} Z_N^* ; x \leftarrow r^{2E} ; h \leftarrow z^{2E}$$

Simulate σ_i :

$$y'_i \leftarrow_{\$} QR_N ; x'_i \leftarrow y'^{e'}_i h^{-H(M_i)} ; y_i \leftarrow (x h^{H(x'_i)})^{1/e_i} = r^{2E/e_i} (z^{2E/e_i})^{H(x'_i)}$$

Recover $w = z^{1/e}$:

$$y'^{e'} = x'^* h^{H(M^*)} \text{ and } y'^{e'}_j = x'_j h^{H(M_j)} \text{ and } H(M^*) \neq H(M_j)$$
$$(y'^*/y'_j)^{e'} = h^{H(M^*) - H(M_j)} = z^{2E(H(M^*) - H(M_j))}$$

use Shamir's trick to recover $w = z^{1/e'}$

$pk = (N, e', x, h)$, $\sigma = (e, y, y')$ such that

- $y^e = x h^{H(x')} \pmod N$
- $y'^{e'} = x' h^{H(M)} \pmod N$

Type-II forgery: For some $1 \leq j \leq q_s$: $e^* = e_j$ and $x'^* \neq x'_j$

→ break one-wayness of RSA: given (N, e, z) compute w such that $w^e = z$

Choose random e', e_i for $i \neq j$; $e_j \leftarrow e$; $E \leftarrow \prod_{i \neq j} e_i$

Simulate pk :

$$r, s \leftarrow_{\$} Z_N^* ; y_j \leftarrow r^{2E} ; x'_j \leftarrow s^{2e'} ; h \leftarrow z^{2e'E} ; x \leftarrow y_j^{e_j} h^{-H(x'_j)}$$

Simulate σ_i :

$$\underline{i \neq j}: y'_i \leftarrow_{\$} QR_N ; x'_i \leftarrow y_i^{e'} h^{-H(M_i)} ; y_i \leftarrow (x h^{H(x'_i)})^{1/e_i} = r^{2Ee_j/e_i} (z^{2e'E/e_i})^{H(x'_i)-H(x'_j)}$$

$$\underline{i = j}: y'_j \leftarrow (x'_j h^{H(M_j)})^{1/e'} = s^2 z^{2EH(M_j)}$$

Recover $w = z^{1/e}$:

$$y^{*e_j} = x h^{H(x'^*)} \text{ and } y_j^{e_j} = x h^{H(x'_j)} \text{ and } H(x'^*) \neq H(x'_j)$$

$$(y^*/y_j)^e = h^{H(x'^*)-H(x'_j)} = z^{2e'E(H(x'^*)-H(x'_j))}$$

use Shamir's trick to recover $w = z^{1/e}$

$pk = (N, e', x, h)$, $\sigma = (e, y, y')$ such that

- $y^e = x h^{H(x')} \pmod N$
- $y'^{e'} = x' h^{H(M)} \pmod N$

Type-III forgery: For all $1 \leq i \leq q_S : e^* \neq e_i$

→ break strong RSA: given (N, z) compute (e, w) such that $w^e = z$

Choose random e', e_i ; $E \leftarrow \prod_i e_i$

Simulate pk :

$$a \leftarrow_{\$} Z_{N^2}^* ; h \leftarrow z^{2e'E} ; x \leftarrow h^a$$

Simulate σ_i :

$$y'_i \leftarrow_{\$} QR_N ; x'_i \leftarrow y'^{e'}_i h^{-H(M_i)} ; y_i \leftarrow (x h^{H(M_i)})^{1/e'} = z^{2E(a+H(M_i))}$$

Recover $e, w = z^{1/e}$:

$$y^{*e^*} = x h^{H(x'^*)} = z^{2e'E(a+H(x'^*))} = z^m$$

$d = \gcd(e, m)$; use Shamir's trick to recover $(e/d)^{\text{th}}$ root $w = z^{d/e}$

$e/d=1$? No, $e \nmid m$ by randomness of a

- Digital signatures I: Random oracle model
 - Signatures based on RSA:
RSA-FDH and the random oracle model
 - Signatures based on discrete logs:
Schnorr signatures and the forking lemma

- Digital signatures II: Standard model
 - Signatures based on one-way functions:
Lamport one-time signatures
 - Signatures based on strong RSA:
Cramer-Shoup signatures
 - **Signatures with protocols:**
Camenisch-Lysyanskaya signatures

Kg:

Choose random primes p, p', q, q'
such that $p=2p'+1$, $q=2q'+1$
 $N \leftarrow pq$
 $a, b, c \leftarrow_{\$} QR_N$
 $pk \leftarrow (N, a, b, c)$; $sk \leftarrow (p, q)$

Sign(sk, M):

Choose random prime $2^{255} < e < 2^{256}$
 $s \leftarrow_{\$} Z_{2^{2688}}$
 $v \leftarrow (a^M b^{sc})^{1/e} \bmod N$
 $\sigma \leftarrow (e, s, v)$

Vf(pk, M, σ):

Check $e > 2^{255}$
Check $v^e = a^M b^{sc} \bmod N$

Secure under strong RSA assumption
Proof somewhat similar to Cramer-Shoup

- Observation 1

CL signatures are “somewhat” re-randomizable

If $\sigma = (e, s, v)$ is valid signature on M , meaning $v^e = a^M b^{sc} \pmod N$

Then $\sigma = (e, s' = s + er, v' = vb^r)$ is also valid signature on M :

$$v'^e = v^e b^{er} = a^M b^{sc} b^{er} = a^M b^{s+er} c \pmod N$$

- Observation 2

Generalized Schnorr proofs [CKY09]: prove any statements of the form

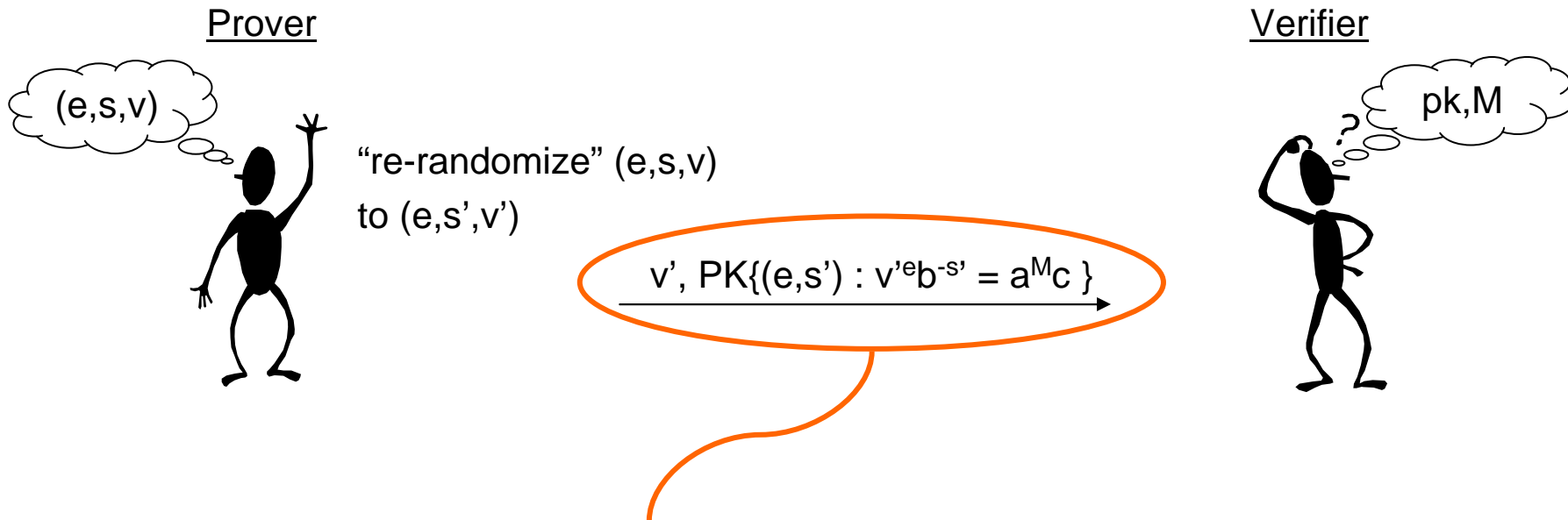
$$\text{PK}\{ (x_1, \dots, x_n) : \bigwedge_i (\prod_j A_{i,j}^{x_j} = C_i \pmod{N_i}) \}$$

Well-suited for CL signatures as most signature values in exponent

Prove knowledge of a CL signature



- Prove knowledge of valid signature on M:

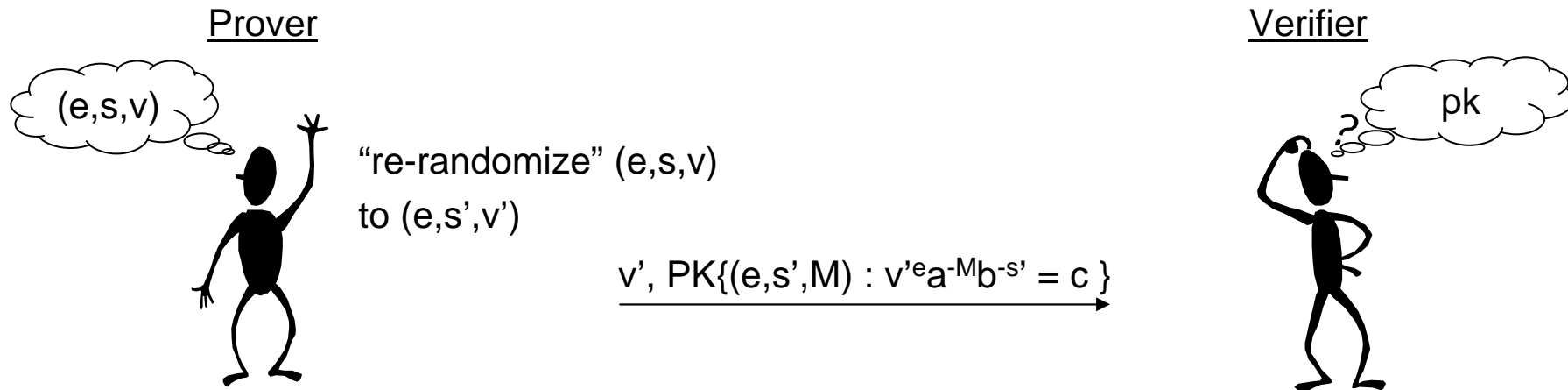


no information whatsoever on signature itself
→ prover remains “anonymous”

Prove knowledge of a CL signature



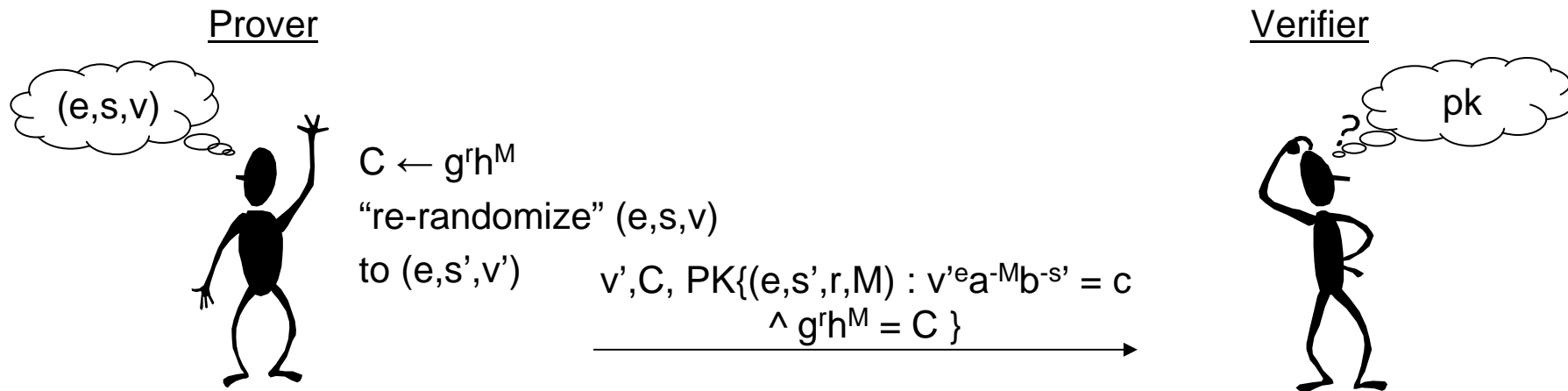
- Prove knowledge of valid signature on M
- Prove knowledge of valid signature on some undisclosed M



Prove knowledge of a CL signature



- Prove knowledge of valid signature on M
- Prove knowledge of valid signature on some undisclosed M
- Prove knowledge of valid signature on committed M



CL signatures for blocks of messages



Kg:

Choose random primes p, p', q, q'

such that $p=2p'+1, q=2q'+1$

$N \leftarrow pq$

$a_1, \dots, a_n, b, c \leftarrow_{\$} QR_N$

$pk \leftarrow (N, a_1, \dots, a_n, b, c) ; sk \leftarrow (p, q)$

Sign(sk, M):

Choose random prime $2^{255} < e < 2^{256}$

$s \leftarrow_{\$} Z_{2^{2688}}$

$v \leftarrow (a_1^{M_1} \dots a_n^{M_n} b^{sc})^{1/e} \bmod N$

$\sigma \leftarrow (e, s, v)$

Vf(pk, M, σ):

Check $e > 2^{255}$

Check $v^e = a_1^{M_1} \dots a_n^{M_n} b^{sc} \bmod N$

Secure under strong RSA assumption

- Prove knowledge of valid signature on M
- Prove knowledge of valid signature on some undisclosed M
- Prove knowledge of valid signature on committed M
- Prove knowledge of valid signature on M_i for $i \in S \subseteq \{1, \dots, n\}$
- Prove knowledge of valid signature on $M \in [a, b]$
- Signing committed message, blind signatures
- Anonymity revocation, group signatures
- Prove knowledge of multiple signatures with relations among messages
- ...
- Idemix anonymous credential system

- Standard model preferable over ROM:
real-world hash functions, rather than utopian idealizations
- But... performance cost (even if sometimes small)
- ROM seems to have been accepted in practice
- Mixed opinions in theoretic community

- Personal opinion:
ROM perfectly fine for real-world crypto primitives
but standard-model schemes do have merit as
 - theoretic achievements
 - building blocks for “fancier” schemes and protocols

- [B08] D. Bernstein: Proving Tight Security for Rabin-Williams Signatures. EUROCRYPT 2008: 70-87.
- [BB04] D. Boneh, X. Boyen: Short Signatures Without Random Oracles. EUROCRYPT 2004: 56-73.
- [BN06] M. Bellare, G. Neven: Multi-signatures in the plain public-key model and a general forking lemma. ACM Conference on Computer and Communications Security 2006: 390-399.
- [C00] J.-S. Coron: On the Exact Security of Full Domain Hash. CRYPTO 2000: 229-235.
- [C02] J.-S. Coron: Optimal Security Proofs for PSS and Other Signature Schemes. EUROCRYPT 2002: 272-287.
- [CGH98] R. Canetti, O. Goldreich, S. Halevi: The Random Oracle Methodology, Revisited. STOC 1998: 209-218.
- [CKY09] J. Camenisch, A. Kiayias, M. Yung: On the Portability of Generalized Schnorr Proofs. EUROCRYPT 2009.
- [CL02] J. Camenisch, A. Lysyanskaya: A Signature Scheme with Efficient Protocols. SCN 2002: 268-289.
- [CS99] R. Cramer, V. Shoup: Signature Schemes Based on the Strong RSA Assumption. ACM CCS 1999: 46-51.
- [DOP05] Y. Dodis, R. Oliveira, K. Pietrzak: On the Generic Insecurity of the Full Domain Hash. CRYPTO 2005: 449-466.
- [GHR99] R. Gennaro, S. Halevi, T. Rabin: Secure Hash-and-Sign Signatures Without the Random Oracle. EUROCRYPT 1999: 123-139.
- [GMR88] S. Goldwasser, S. Micali, R. Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks. SIAM J. Comput. 17(2): 281-308 (1988).
- [HK08] D. Hofheinz, E. Kiltz: Programmable Hash Functions and Their Applications. CRYPTO 2008: 21-38.
- [HW09] S. Hohenberger, B. Waters: Realizing Hash-and-Sign Signatures under Standard Assumptions. EUROCRYPT 2009: 333-350.
- [KW03] J. Katz, N. Wang: Efficiency improvements for signature schemes with tight security reductions. ACM Conference on Computer and Communications Security 2003: 155-164.
- [LN09] G. Leurent, P. Nguyen: How Risky is the Random-Oracle Model? CRYPTO 2009.
- [PS00] D. Pointcheval, J. Stern: Security Arguments for Digital Signatures and Blind Signatures. J. Cryptology 13(3): 361-396 (2000).
- [W05] B. Waters: Efficient Identity-Based Encryption Without Random Oracles. EUROCRYPT 2005: 114-127