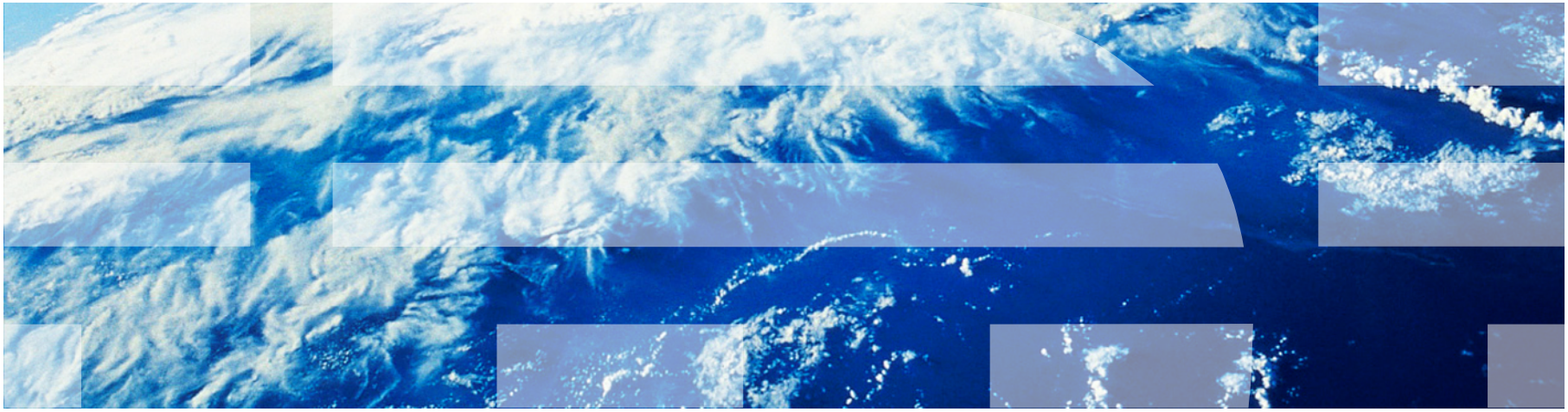
Gregory Neven, IBM Research – Zurich
Cryptography summer school, July 21-24, 2014, Bucharest

Digital signatures I



Overview

- Digital signatures I
  - Signatures based on RSA:
    RSA-FDH and the random-oracle model
  - Zero-knowledge proofs
  - Schnorr protocol

- Digital signatures II
  - Schnorr signatures and the forking lemma
  - Signatures based on one-way functions:
    Lamport one-time signatures
  - Signatures based on strong RSA:
    Camenisch-Lysyanskaya signatures
  - Signatures with protocols

Overview

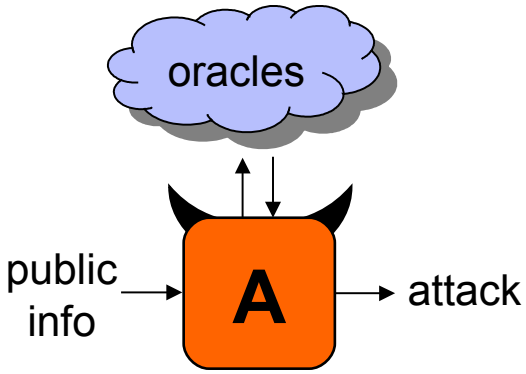
- Digital signatures I
  - **Signatures based on RSA:**
    RSA-FDH and the random-oracle model
  - Zero-knowledge proofs
  - Schnorr protocol

- Digital signatures II
  - Schnorr signatures and the forking lemma
  - Signatures based on one-way functions:
    Lamport one-time signatures
  - Signatures based on strong RSA:
    Camenisch-Lysyanskaya signatures
  - Signatures with protocols

Provable security in a nutshell

1. **Security model:** game with adversary

   concrete security: no adversary A running in time t making q oracle queries has advantage more than ε

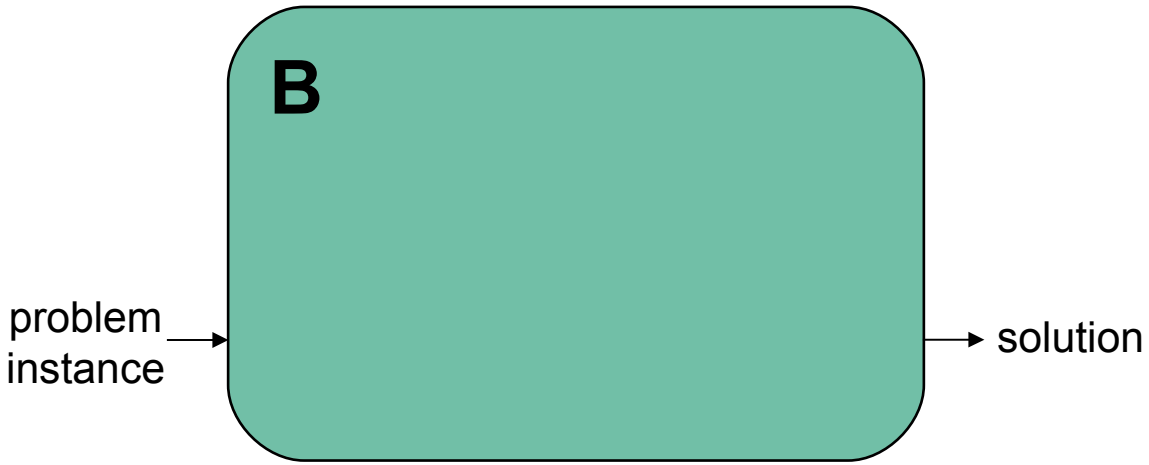


Provable security in a nutshell

1. **Security model:** game with adversary

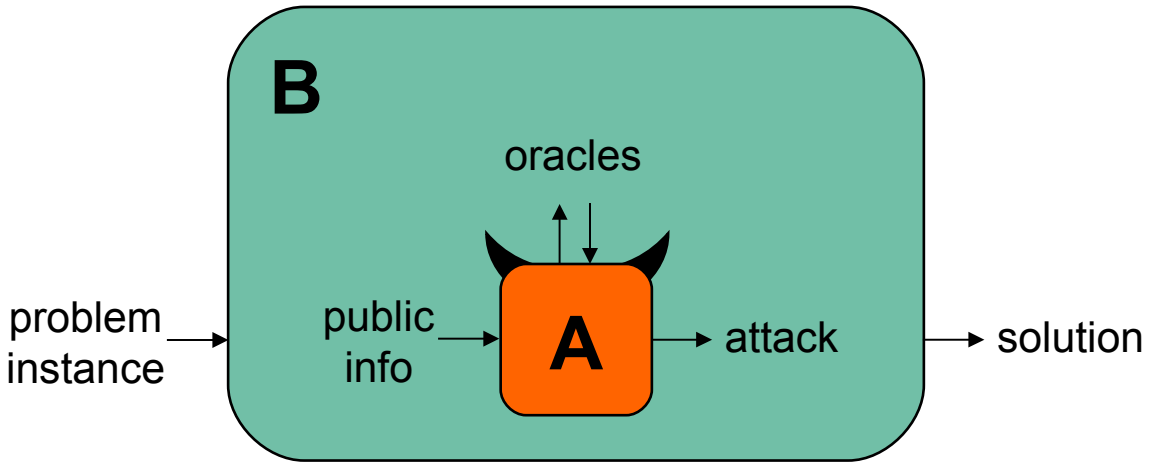
   concrete security: no adversary A running in time t has advantage > ε

2. **Security assumption:** hardness of math/crypto problem



Provable security in a nutshell

1. **Security model:** game with adversary
   concrete security: no adversary A running in time t has advantage > ε
2. **Security assumption:** hardness of math/crypto problem
3. **Security proof:** scheme is secure if assumption holds
   by reduction: given A breaking scheme, build B breaking assumption



Syntax of digital signatures

- Digital signature scheme DS = (Kg, Sign, Vf) where
  - Key generation: $(pk, sk) \leftarrow_{\$} Kg$
  - Signing: $\sigma \leftarrow_{\$} Sign(sk, M)$
  - Verification: $0/1 \leftarrow Vf(pk, M, \sigma)$

- Correctness

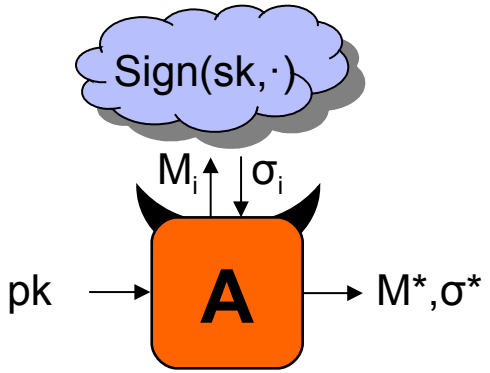
  $Vf(pk, M, Sign(sk, M)) = 1$

Security for digital signatures

- Desirable security properties
  - Given pk, hard to compute sk
  - Given M, hard to compute σ such that Vf(pk, M, σ) = 1
  - Hard to compute σ, M such that Vf(pk, M, σ) = 1
  - Given σ_1 for M_1, hard to compute σ_2 for M_2
  - …

Security model for digital signatures

Unforgeability under chosen-message attack [GMR88]



$(pk,sk) \leftarrow_{\$} Kg$

$(M^*,\sigma^*) \leftarrow_{\$} A^{Sign(sk,\cdot)}(pk)$

A wins iff

$Vf(pk,M^*,\sigma^*)=1$ and $M' \in \{M_1,\ldots,M_N\}$

Advantage $\varepsilon = Pr[\text{ A wins }]$

Some number theory

$N = pq$ where p,q primes, $|p| = |q| = k$

$Z^*_N = \{ x \in Z_N : \gcd(x,N) = 1 \}$

group under multiplication mod N

group order $\varphi(N) = (p-1)(q-1)$

e, d such that $e \cdot d = 1 \bmod \varphi(N)$

meaning $e = d^{-1} \bmod \varphi(N)$, $\gcd(e,N) = \gcd(d,N) = 1$

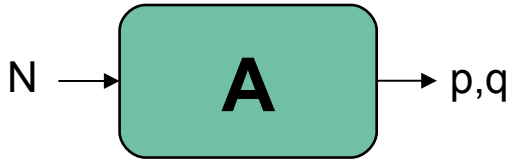
$RSA_{N,e}(x) = x^e \bmod N$

$RSA^{-1}_{N,e}(y) = y^d \bmod N$

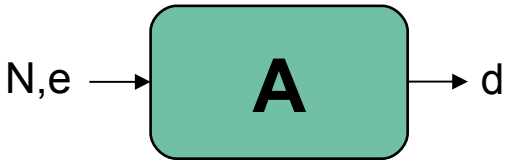
$RSA^{-1}(RSA(x)) = (x^e)^d \bmod N = x^{ed} \bmod \varphi(N) = x$

Assumptions related to RSA

- Factoring is hard:

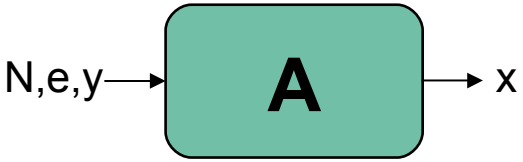


- Finding inverse exponent is hard



$e = d^{-1} \bmod (p-1)(q-1)$

- Finding RSA inverse is hard = one-wayness of RSA



$(N,e,d) \leftarrow_{\$} Kg_{RSA}(1^k)$

$y \leftarrow_{\$} Z^*_N$

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

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

“Textbook” RSA signatures

IBM

Kg:

$N = pq$ where p,q primes, $|p| = |q| = k$

e,d such that $e \cdot d = 1 \bmod \varphi(N)$

$pk \leftarrow (N,e)$

$sk \leftarrow (N,d)$

Sign(sk,M):

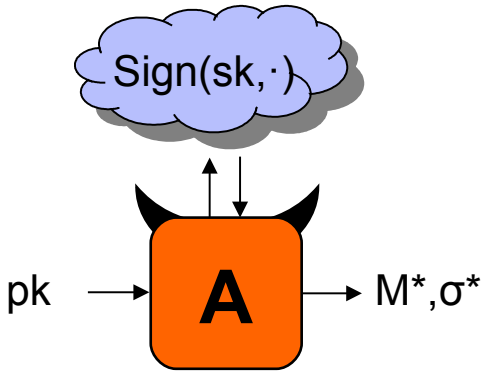
(assume $M \in Z^*_N$)

$\sigma \leftarrow M^d \bmod N$

Vf(pk,M,σ):

Check that $\sigma^e = M \bmod N$

Are these uf-cma secure?



“Textbook” RSA signatures

IBM

Kg:

$N = pq$ where p,q primes, $|p| = |q| = k$

e,d such that $e \cdot d = 1 \bmod \varphi(N)$

$pk \leftarrow (N,e)$

$sk \leftarrow (N,d)$

Sign(sk,M):

(assume $M \in Z^*_N$)

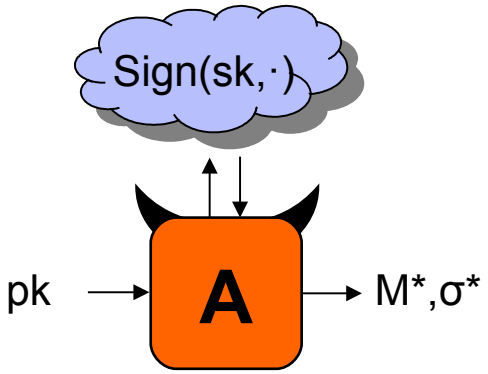
$\sigma \leftarrow M^d \bmod N$

Vf(pk,M,σ):

Check that $\sigma^e = M \bmod N$

Are these uf-cma secure? **No!**

- $\sigma^* = 1$ always valid for $M^* = 1$
- choose σ^*, compute $M^* \leftarrow \sigma^e \bmod N$
- homomorphism:
  (M_1,σ_1) and $(M_2,\sigma_2) \rightarrow (M_1M_2 , \sigma_1\sigma_2)$



"Textbook" RSA signatures

IBM

Kg:

$N = pq$ where p,q primes, $|p| = |q| = k$

e,d such that $e \cdot d = 1 \bmod \varphi(N)$

$pk \leftarrow (N,e)$

$sk \leftarrow (N,d)$

Sign(sk,M):

(assume $M \in Z^*_N$)

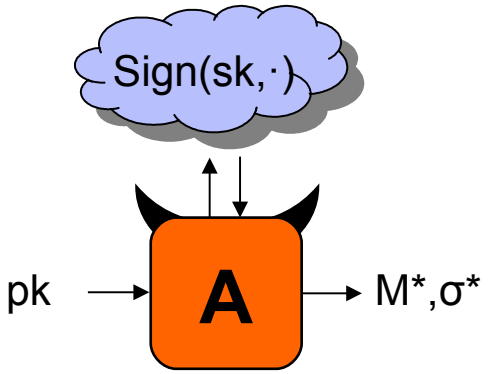
$\sigma \leftarrow M^d \bmod N$

Vf(pk,M,σ):

Check that $\sigma^e = M \bmod N$

Are these uf-cma secure? **No!**

- homomorphism:
  (M_1,σ_1) and $(M_2,\sigma_2) \rightarrow (M_1M_2 , \sigma_1\sigma_2)$



Exercise: Can you forge signature for **any** M?

Hint: Use signing oracle and homomorphism

“Textbook” RSA signatures

IBM

Kg:

$N = pq$ where p,q primes, $|p| = |q| = k$

e,d such that $e \cdot d = 1 \bmod \varphi(N)$

$pk \leftarrow (N,e)$

$sk \leftarrow (N,d)$

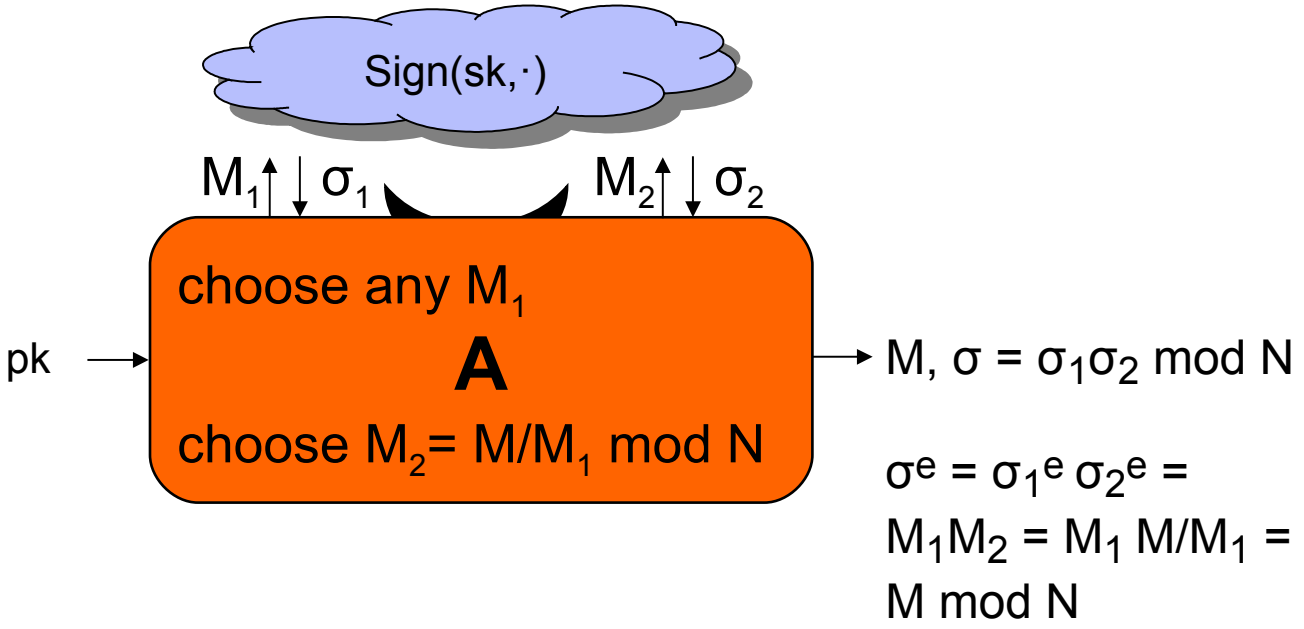
Sign(sk,M):

(assume $M \in Z^*_N$)

$\sigma \leftarrow M^d \bmod N$

Vf(pk,M,σ):

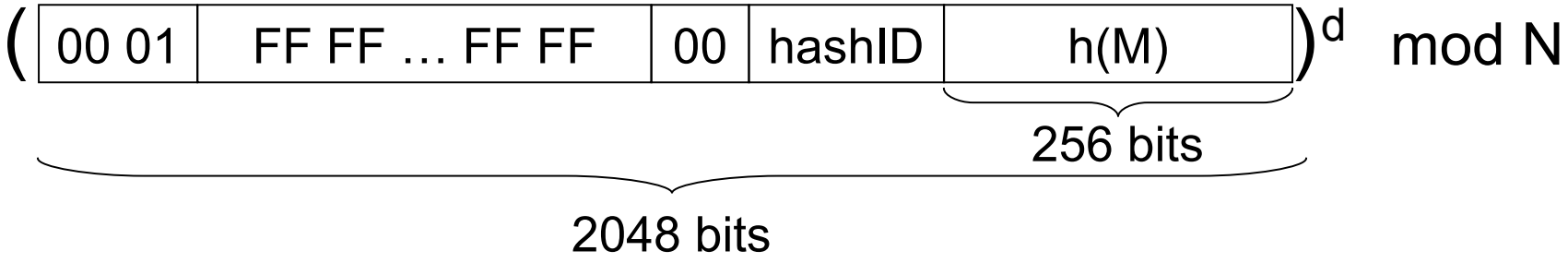
Check that $\sigma^e = M \bmod N$



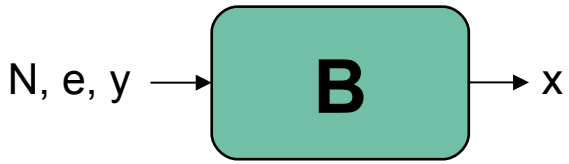
RSA PKCS#1 v1.5 signatures

- Industry standard:

  Public Key Cryptography Standards (PKCS) by RSA Labs:



- Seems to prevent attacks, but provably secure?
- Candidate assumption: one-wayness of RSA



$y \leftarrow_{\$} Z^*_N$

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

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

- Padded messages are only fraction $1/2^{1792}$ of Z^*_N

  So RSA may be one-way on average yet invertible for padding above!

Full-domain hash (RSA-FDH)

IBM

Kg:

$N = pq$ where p,q primes, $|p| = |q| = k$

e,d such that $e \cdot d = 1 \bmod \varphi(N)$

$pk \leftarrow (N,e)$; $sk \leftarrow (N,d)$

Sign(sk,M):

$\sigma \leftarrow H(M)^d \bmod N$

Vf(pk,M,σ):

Check that $\sigma^e = H(M) \bmod N$

where H is "full-domain" hash function $H : \{0,1\}^* \rightarrow Z_N^*$

What do we need/expect/hope to get from H?

- preimage of 1 hard to find
- one-wayness: hard to choose σ, compute $M \leftarrow H^{-1}(\sigma^e)$
- collision-resistance: hard to find M, M' such that $H(M) = H(M')$
- destroy algebraic structure: hard to find M_1, M_2, M_3 such that $H(M_1) \cdot H(M_2) = H(M_3) \bmod N$

These are **necessary** properties, but are they **sufficient**?

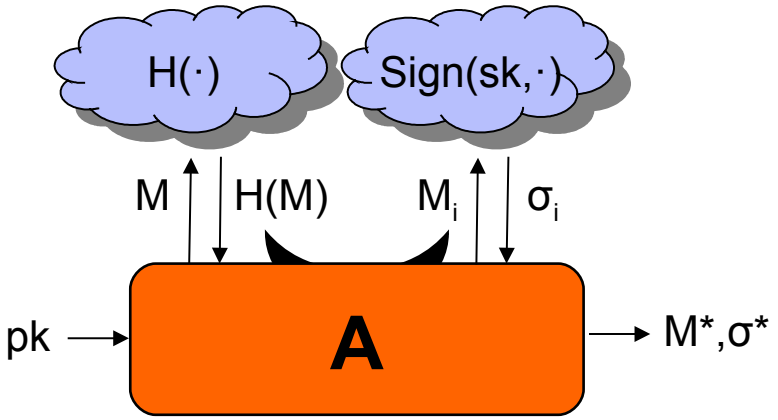
Random oracle model

Theory: give all parties (good & bad) access to ideal hash function

"random oracle" = truly random function H: $\{0,1\}^* \rightarrow Z_N^*$

consistent with previous queries (≈ dynamically built table)

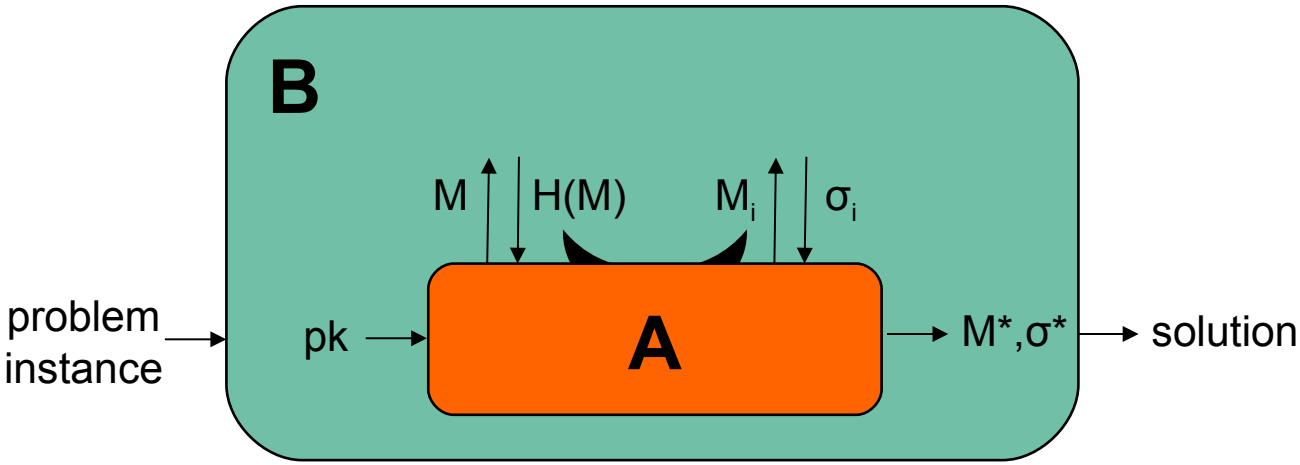
Practice: replace random oracle with hash function



Random oracle model

Random oracle model is stronger than

- collision-resistant hash function
  hash: computable ↔ RO: unpredictable until queried
- pseudo-random function:
  PRF: secret key unknown to A ↔ RO: publicly accessible



Random oracle model: pros & cons

- Pros
  - efficient, practical schemes
  - clear security notion, “some” security guarantee
    (much better than ad-hoc design without security proof)
  - excludes *generic* attacks
    (if scheme and hash function are “independent”)
- Cons
  - weaker security guarantee than standard model
  - (contrived) counterexamples exist [CGH98]

Security of RSA-FDH

Theorem:

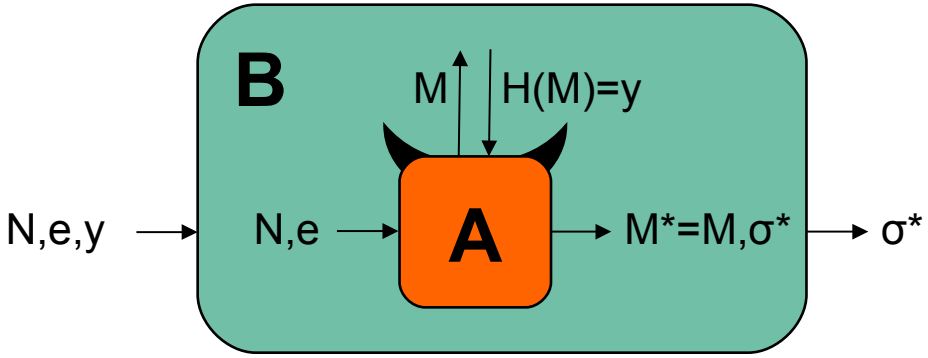
If RSA is (t,ε) one-way, then RSA-FDH signatures are $(t', q_H, q_S, \varepsilon')$ unforgeable in the ROM for

$$t' = t - (q_H + q_S)\, t_{exp}$$

$$\varepsilon' = (q_H + q_S + 1)\, \varepsilon$$

Proof of RSA-FDH

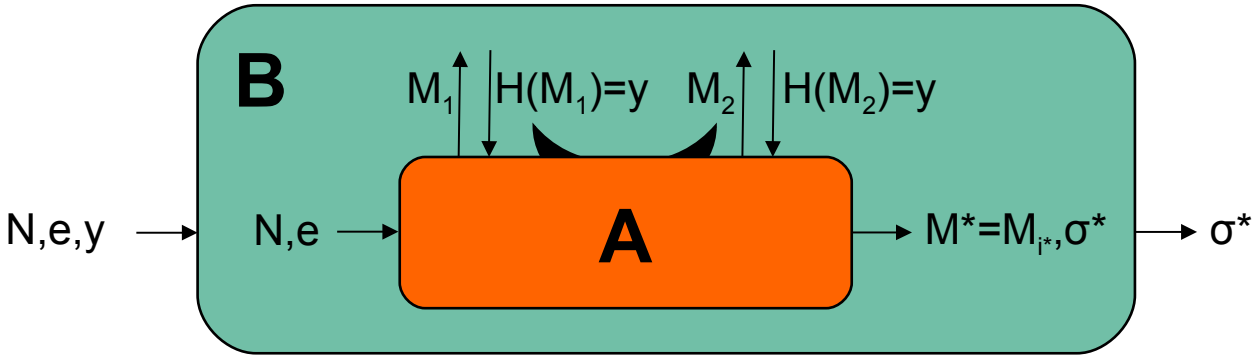
Step 1: Assume A makes 1 query H(M), no Sign queries, forges on M



- Check: inputs & oracles look like real attack?
  → yes, (N,e) correctly distributed, y uniform over Z^*_N
- Check: solution correct?
  → yes, $\sigma^{*e} = H(M) = y \bmod N$
- Advantage: $\varepsilon = \varepsilon'$

Proof of RSA-FDH

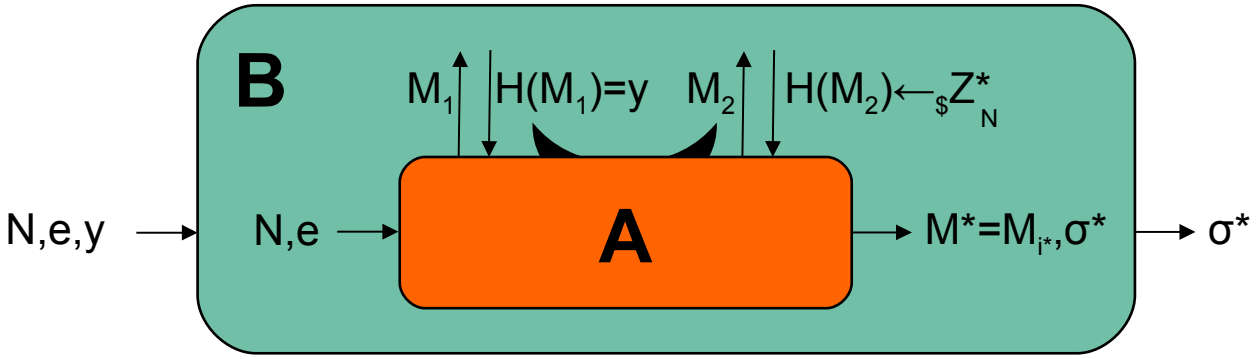
Step 2: Assume A makes 2 queries $H(M_1)$, $H(M_2)$, no Sign queries, forges on one of M_1, M_2



- Check: inputs & oracles look like real attack?
  → **no!** Pr [$H(M_1) = H(M_2)$] = $1/2^{2048}$, so normally doesn't happen

Proof of RSA-FDH

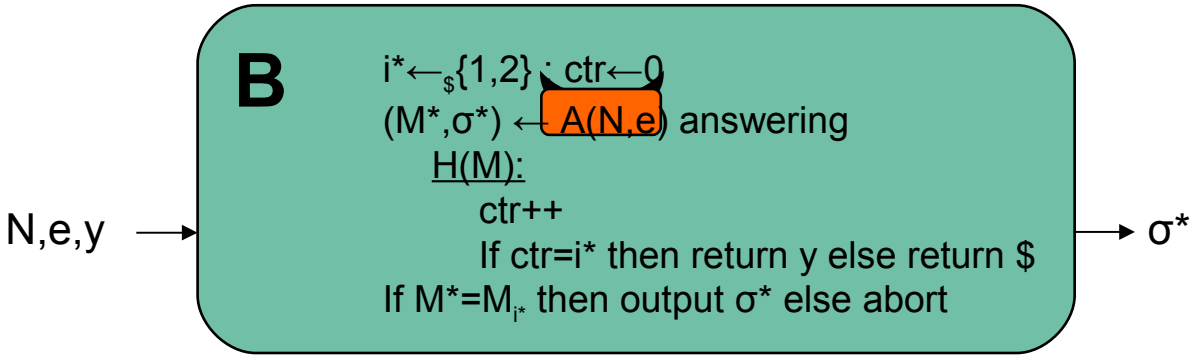
Step 2: Assume A makes 2 queries $H(M_1)$, $H(M_2)$, no Sign queries, forges on one of M_1,M_2



- Check: inputs & oracles look like real attack?
  → yes, $H(M_1)$, $H(M_2)$ independently random
- Check: solution correct?
  → **no!** what if A always forges M_2?

Proof of RSA-FDH

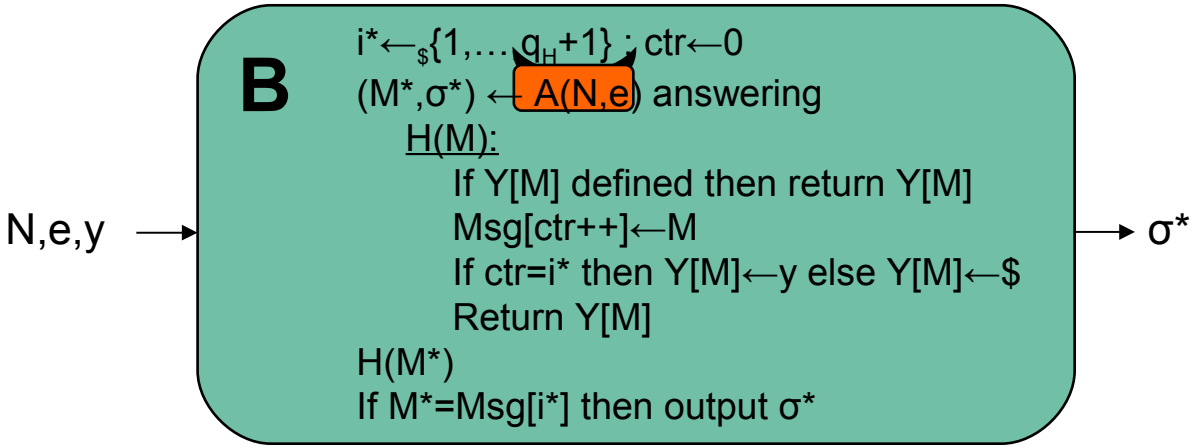
Step 2: Assume A makes 2 queries $H(M_1)$, $H(M_2)$, no Sign queries, forges on one of M_1,M_2



- Check: inputs & oracles look like real attack?
  → yes, $H(M_1)$, $H(M_2)$ independently random
- Check: solution correct?
  → yes, **if $M^*=M_{i^*}$**
- Advantage: $\varepsilon = \varepsilon'/2$

Proof of RSA-FDH

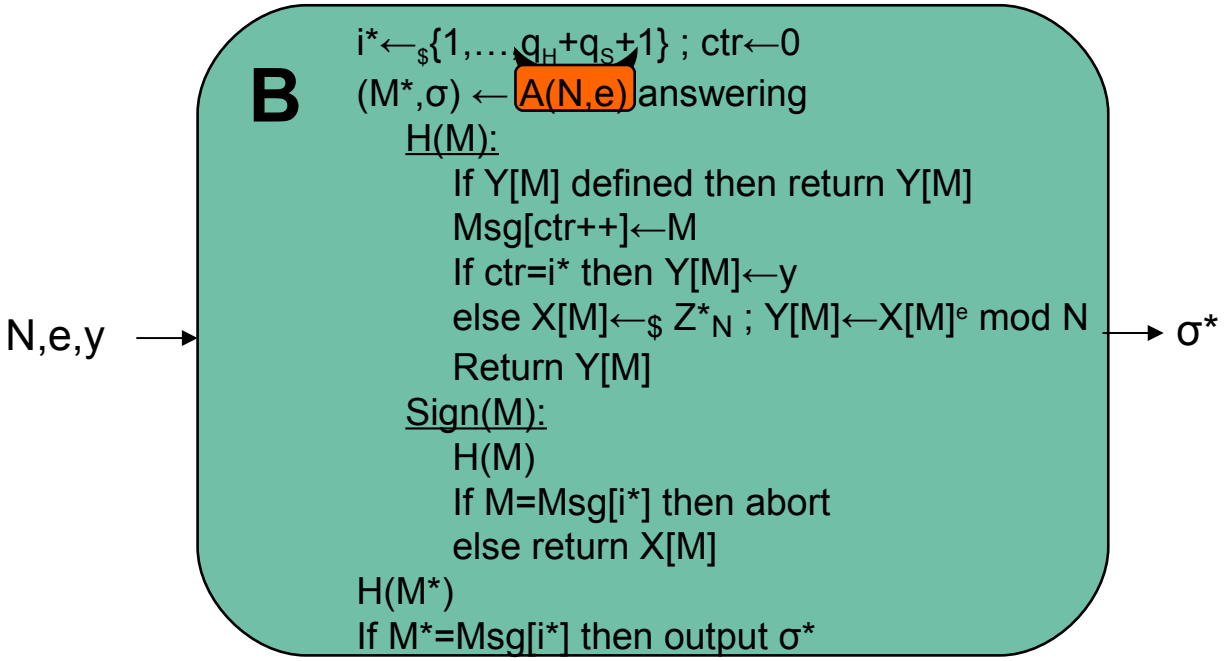
Step 3: Assume A makes q_H H(·) queries, no Sign queries, forges on any message



- Check: inputs & oracles look like real attack? → yes
- Check: solution correct? → yes, if M*=Msg[i*]
- Advantage: $\varepsilon = \varepsilon'/(q_H+1)$

Proof of RSA-FDH

Step 4: Assume A makes q_H H(·) queries, q_S Sign(·) queries, forges on any message

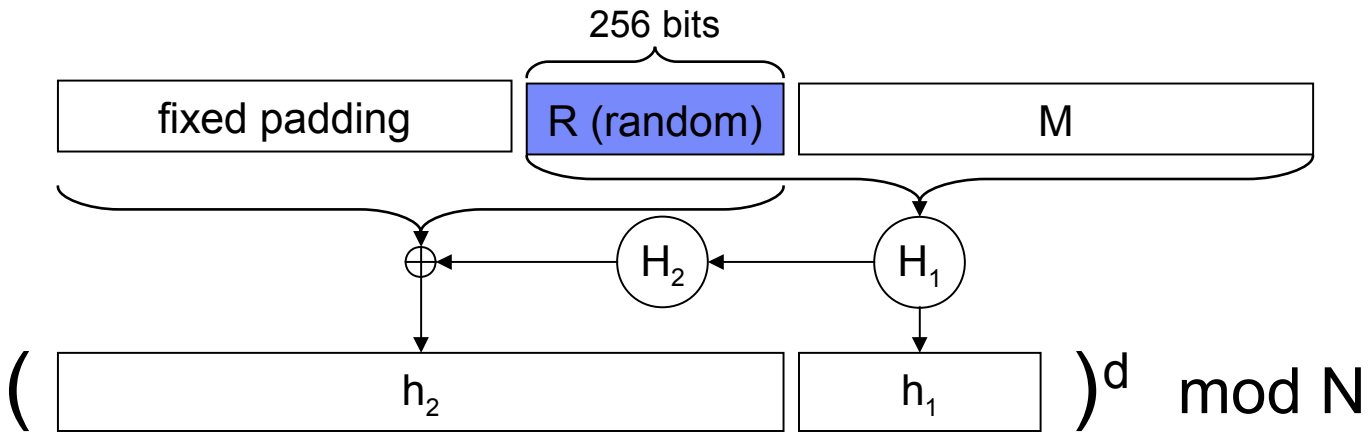


- Check: inputs & oracles look like real attack? → yes
- Check: solution correct? → yes, if M*=Msg[i*]
- Advantage: $\varepsilon = \varepsilon'/(q_H+q_S+1)$

Probabilistic signing: RSA-PSS

- Advantage: $\varepsilon = \varepsilon'/(q_H+q_S+1)$ → what does this mean?
  - theoretically: forging signatures may be easier than inverting RSA
  - practically: debatable…

- PKCS#1 v2.1 recommends using RSA-PSS instead:

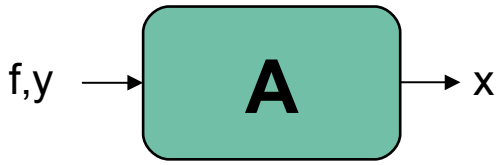


has “tight” security proof, i.e., $\varepsilon \approx \varepsilon'$

Homework 1

- Trapdoor one-way permutation $f : D \rightarrow D$
  $(f,tr) \leftarrow_{\$} Kg(1^k)$, $y \leftarrow f(x)$, $x \leftarrow Inv(tr,y)$

  is (t,ε) one-way iff for all A running in time t



$(f,t) \leftarrow_{\$} Kg(1^k)$

$y \leftarrow_{\$} D$

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

Avantage $\varepsilon = Pr [y = f(x) \bmod N]$

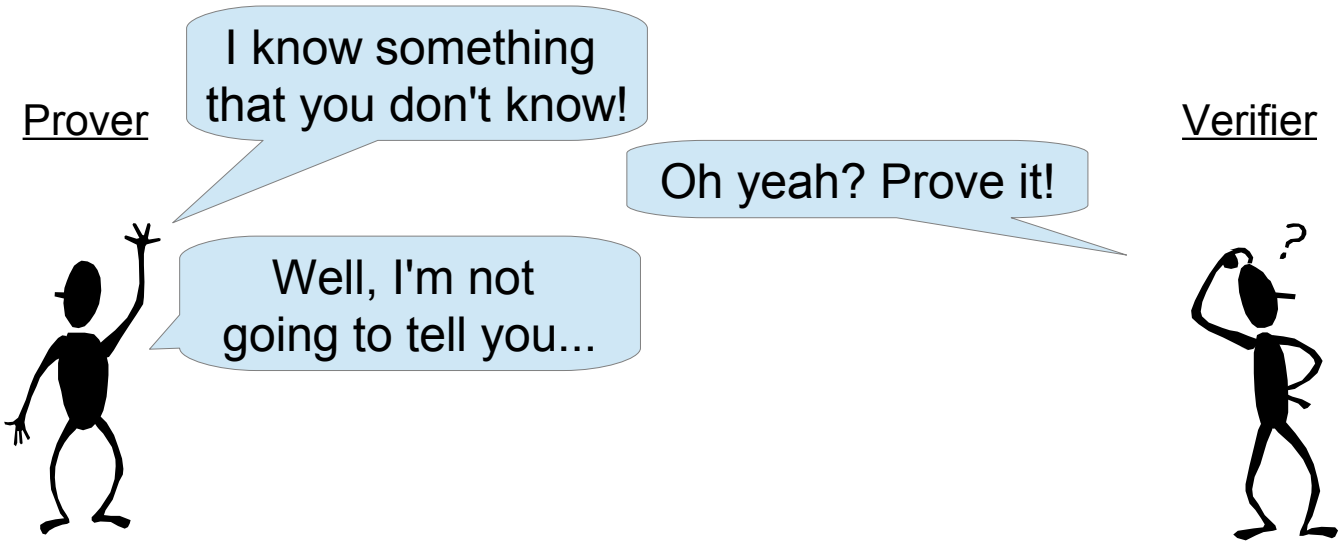
- Design a signature scheme for messages $M \in D$ and prove it uf-cma secure in the random-oracle model

Overview

- Digital signatures I
  - Signatures based on RSA:
    RSA-FDH and the random-oracle model
  - **Zero-knowledge proofs**
  - Schnorr protocol

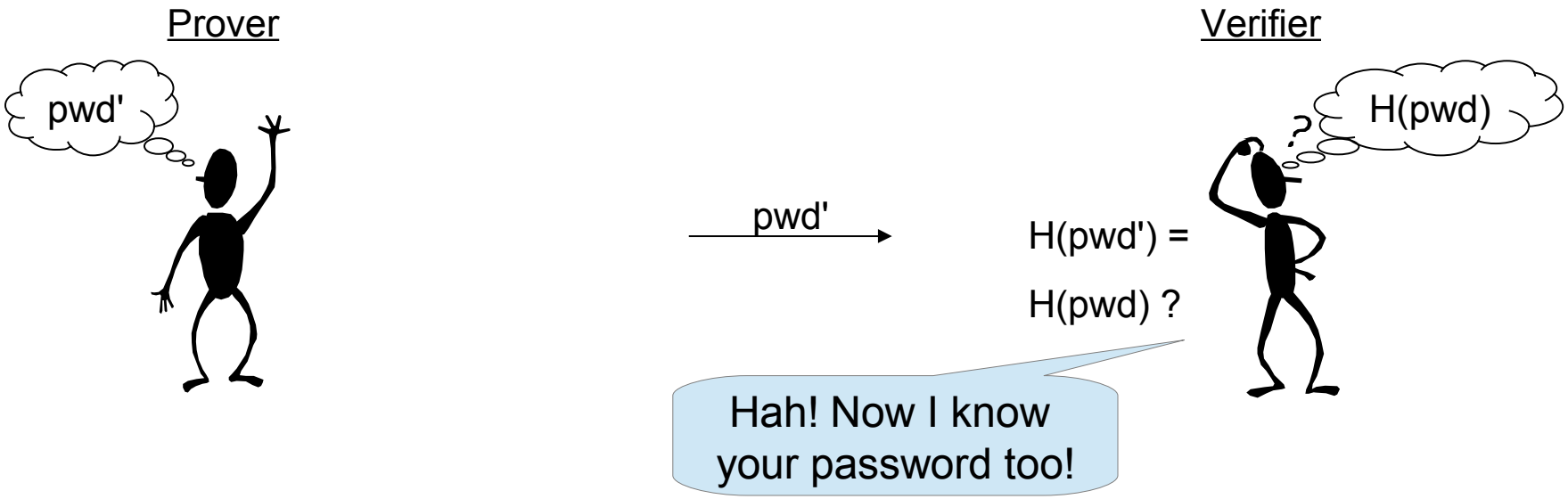
- Digital signatures II
  - Schnorr signatures and the forking lemma
  - Signatures based on one-way functions:
    Lamport one-time signatures
  - Signatures based on strong RSA:
    Camenisch-Lysyanskaya signatures
  - Signatures with protocols

Proof protocols



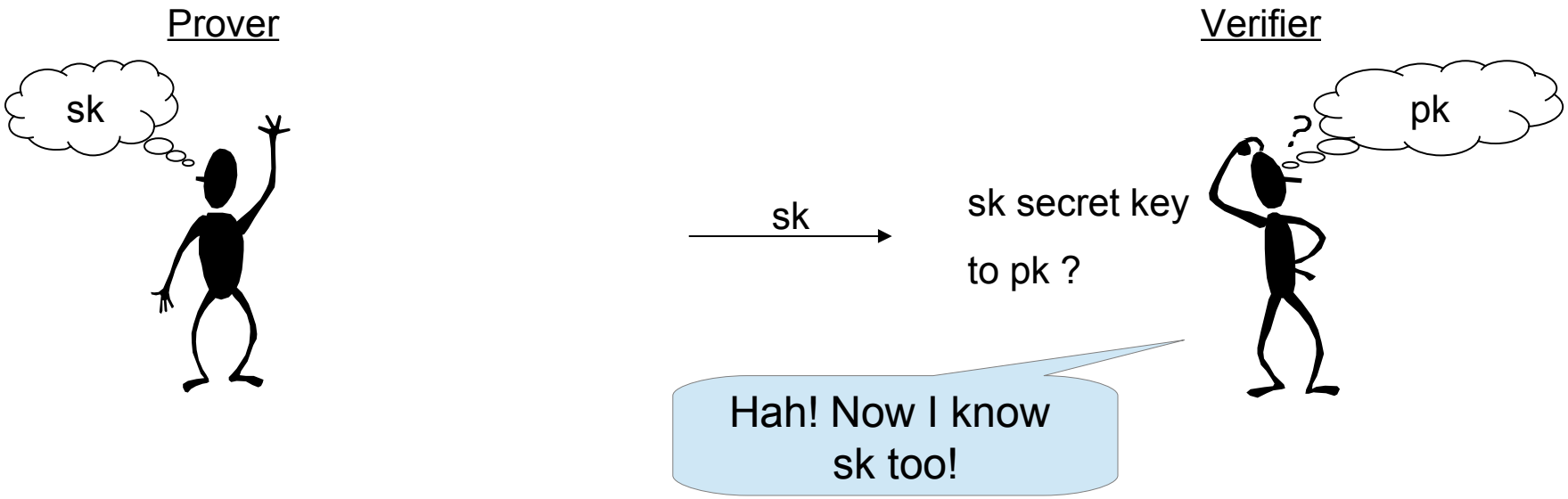
Password authentication

Prover wants to convince Verifier that he knows password



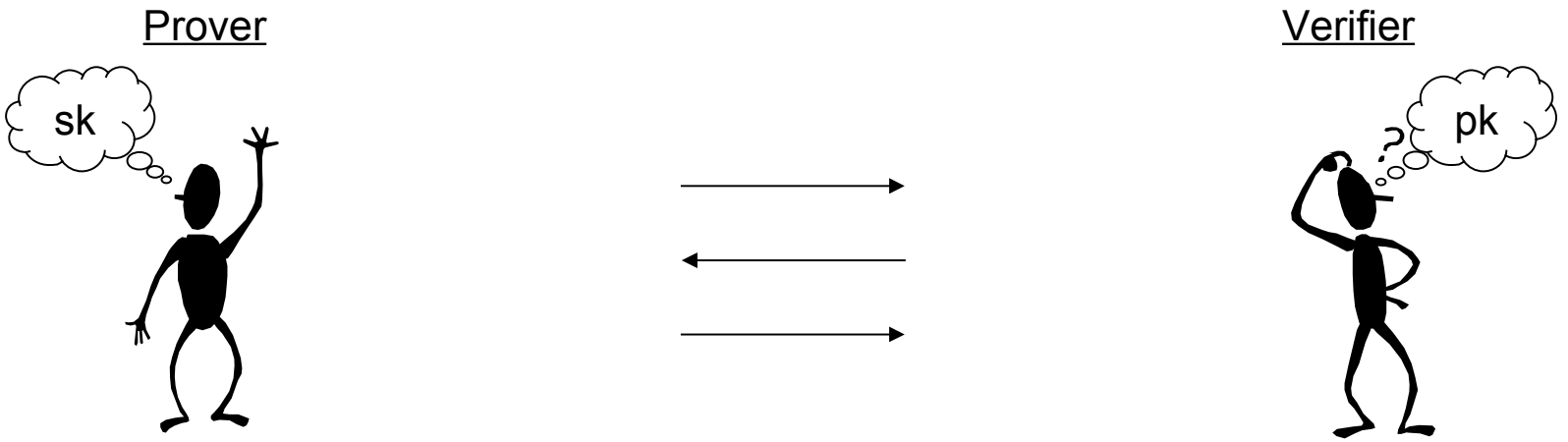
Public-key authentication

Prover wants to convince Verifier that he knows secret key



Zero-knowledge proofs of knowledge

Prover wants to convince Verifier that he knows sk



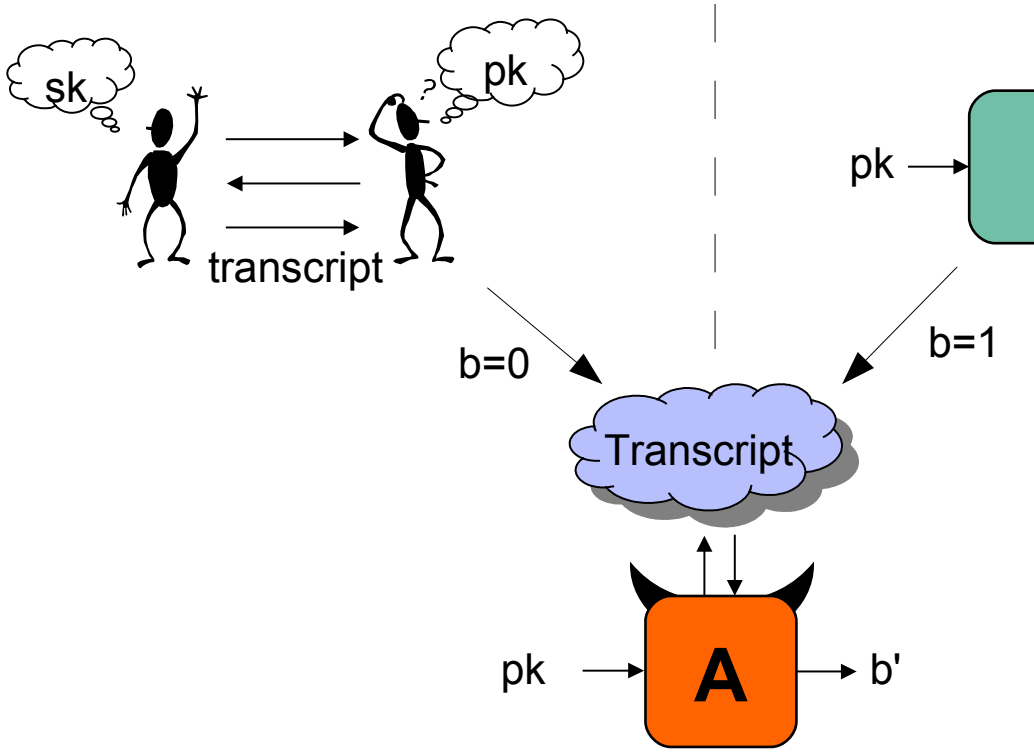
such that

- Verifier doesn't learn anything about sk
- Prover must know sk to succeed

Honest-verifier zero knowledge

"Verifier doesn't learn anything about pk"

(Honest-verifier) zero knowledge: there exists a simulator Sim s.t.



$(pk,sk) \leftarrow_{\$} Kg$

$b' \leftarrow_{\$} A^{Transcript}(pk)$

where Transcript() =

(P(sk) ↔ V(pk)) if b=0

Sim(pk) if b=1

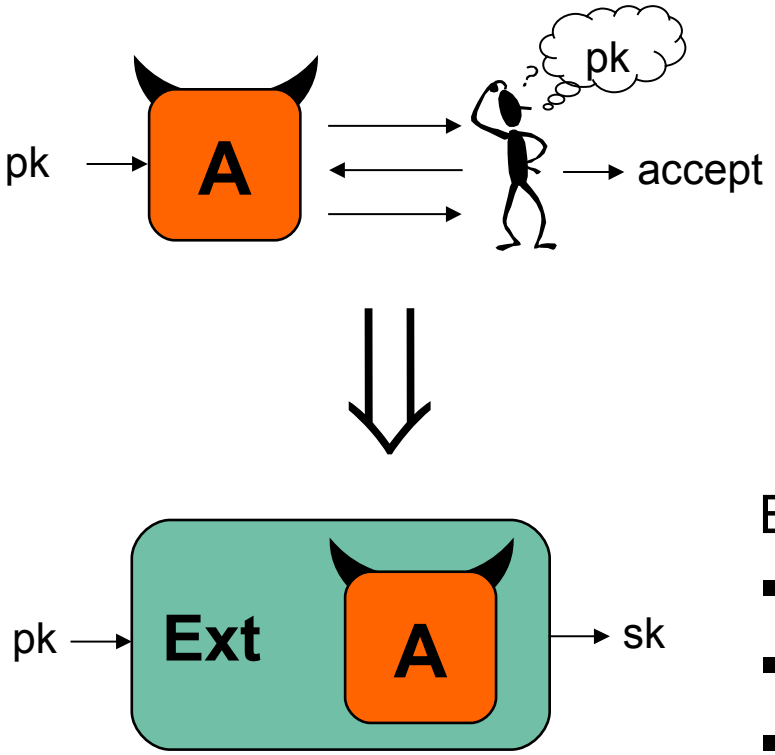
A wins iff b'=b

Advantage $\varepsilon = Pr[\text{ A wins }]$

Proof of knowledge

"Prover must know sk to succeed"

Proof of knowledge: if successful A exists, then extractor Ext_A exists s.t.



Ext has "extra powers" such as

- steering public parameters
- steering random oracle
- rewinding

Overview

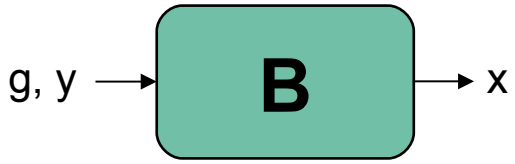
- Digital signatures I
  - Signatures based on RSA:
    RSA-FDH and the random-oracle model
  - Zero-knowledge proofs
  - **Schnorr protocol**

- Digital signatures II
  - Schnorr signatures and the forking lemma
  - Signatures based on one-way functions:
    Lamport one-time signatures
  - Signatures based on strong RSA:
    Camenisch-Lysyanskaya signatures
  - Signatures with protocols

Discrete logarithm problem

- Cyclic group G of prime order q, generator g
  - prime-order subgroup of Z^*_p where q|p-1
  - elliptic curve group

- Discrete logarithm problem (DLP):



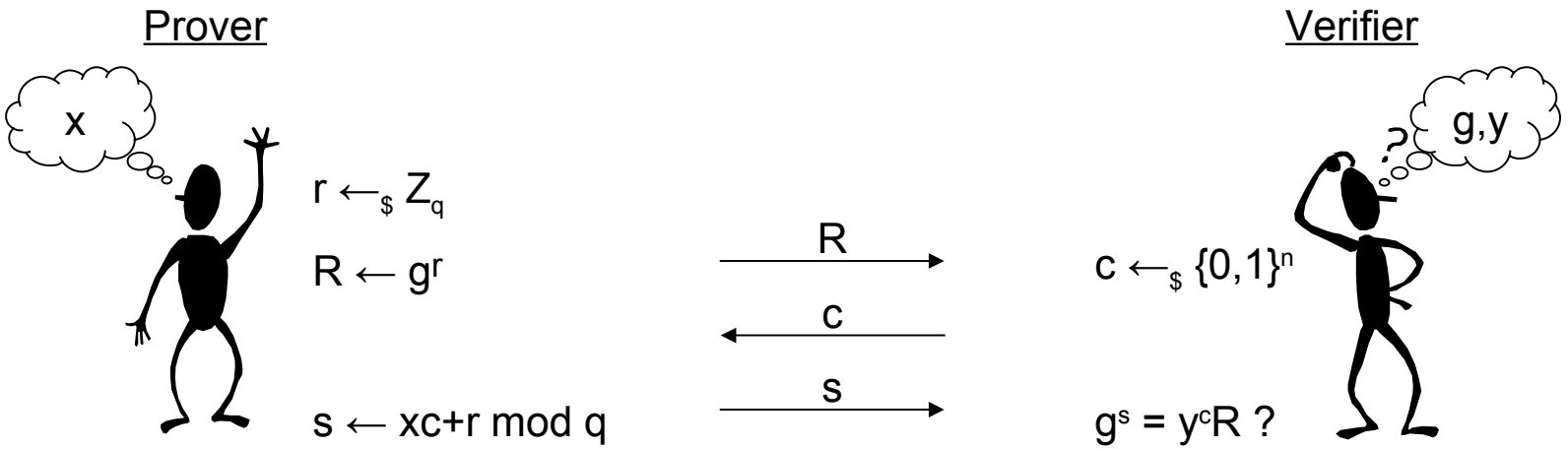
$y \leftarrow_{\$} G$

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

Avantage $\varepsilon = Pr [y = g^x]$

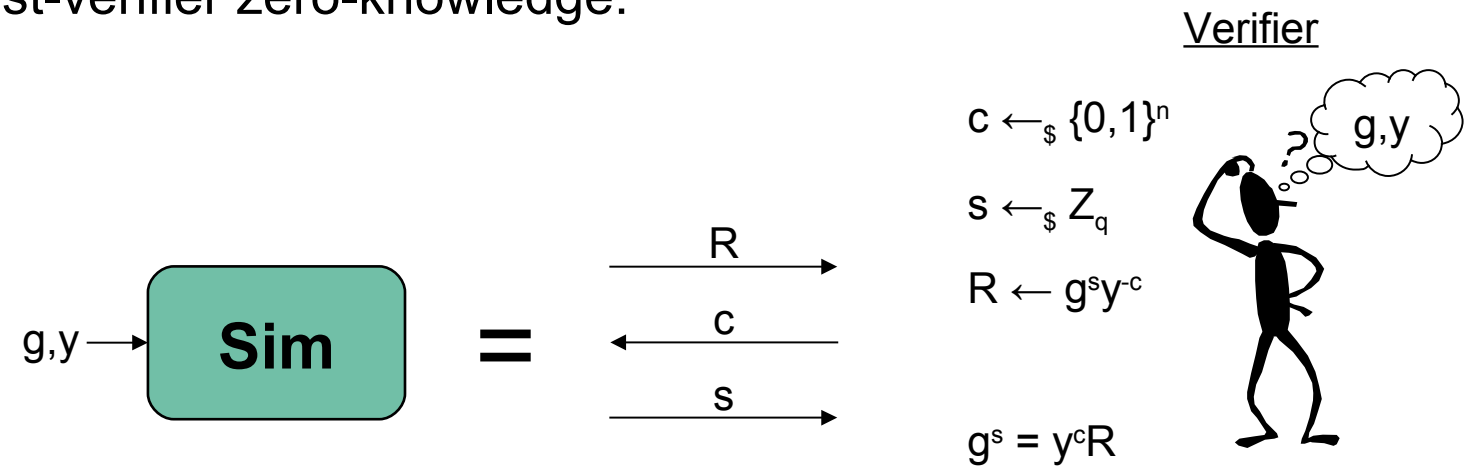
Schnorr protocol

Prover wants to convince Verifier that he knows x : $y = g^x$



Schnorr protocol is HVZK

Honest-verifier zero-knowledge:



Real proof:

R uniform over G because r uniform over Z_q

c uniform over $\{0,1\}^n$

s unique solution such that $g^s = y^c R$

Simulated proof:

c uniform over $\{0,1\}^n$

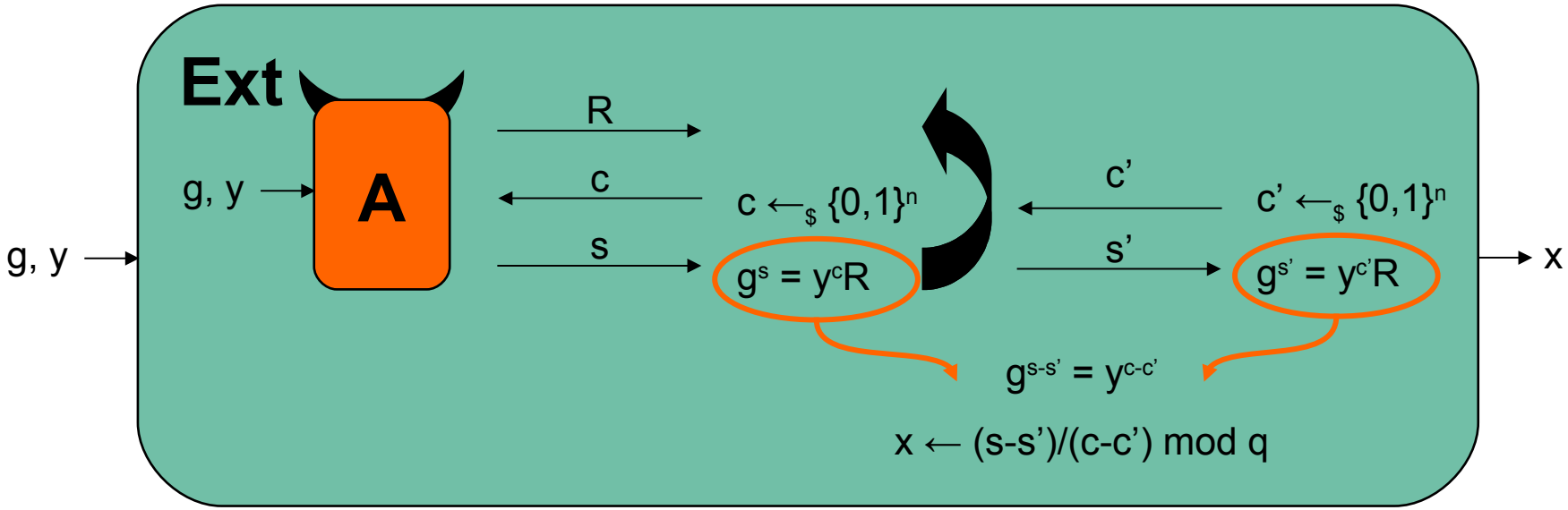
R uniform over G because s uniform over Z_q

s unique solution such that $g^s = y^c R$

Schnorr is proof of knowledge

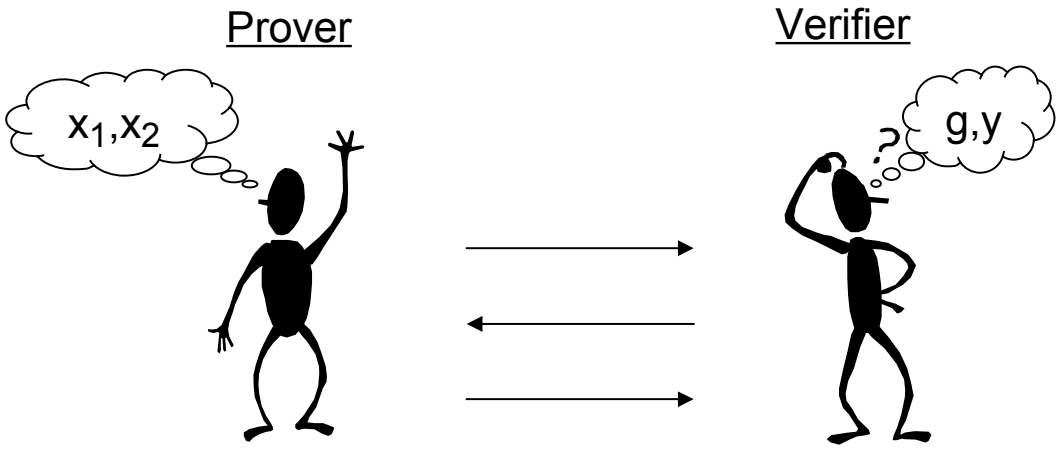
Proof of knowledge:

can extract x from any successful prover



Homework 2

$pk = (g_1, g_2, y)$, $sk = (x_1, x_2)$ such that $y = g_1^{x1}\, g_2^{x2}$



Design a honest-verifier zero-knowledge proof of knowledge, i.e.,

1. Design prover and verifier algorithms
2. Describe simulator
3. Describe extractor

Bibliography

- [BR93] M. Bellare, P. Rogaway: Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. ACM CCS 1993: 62-73.
- [BR96] M. Bellare, P. Rogaway: The Exact Security of Digital Signatures - How to Sign with RSA and Rabin. EUROCRYPT 1996: 399-416.
- [C00] J.-S. Coron: On the Exact Security of Full Domain Hash. CRYPTO 2000: 229-235.
- [CGH98] R. Canetti, O. Goldreich, S. Halevi: The Random Oracle Methodology, Revisited. STOC 1998: 209-218.
- [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).