

# Short and Stateless Signatures from the RSA Assumption

Susan Hohenberger

JOHNS HOPKINS  
UNIVERSITY

Brent Waters

THE UNIVERSITY OF  
TEXAS  
AT AUSTIN™

# Signatures Today

Schemes mostly fall into one of two classes:

## Tree-Based Signatures

- [GMR85, G86, M89, DN89, BM90, NY94, R90, CD95, CD96, ...]
- tradeoff in size of signature and public key

## "Hash-and-Sign" Signatures

- [RSA78, E84, S91, O92, BR93, PS96, GHR99, CS00, CL01, BLS04, BB04, CL04, W05, GJKW07, GPV08, HW09a, ...]
- short signatures and short public keys
- what practitioners expect

# Focus on "Hash-and-Sign"

Again, most things fall into three classes:

## Random Oracle Model

- RSA [RSA78]
- Discrete logarithm [E84,S91]
- Lattices [GPV08]

## Strong Assumptions

- Strong RSA [GHR99, CS00]
- $q$ -Strong Diffie-Hellman [BB04]
- LRSW [CL04]
- Exponential hardness [MRV99]

## Stateful

- RSA, Computational Diffie-Hellman [HW09a]

### Exception?

Waters '05 sigs from CDH.

They are short, but PK needs  $O(k)$  elements for sec. parameter  $k$ .

# Our Main Result

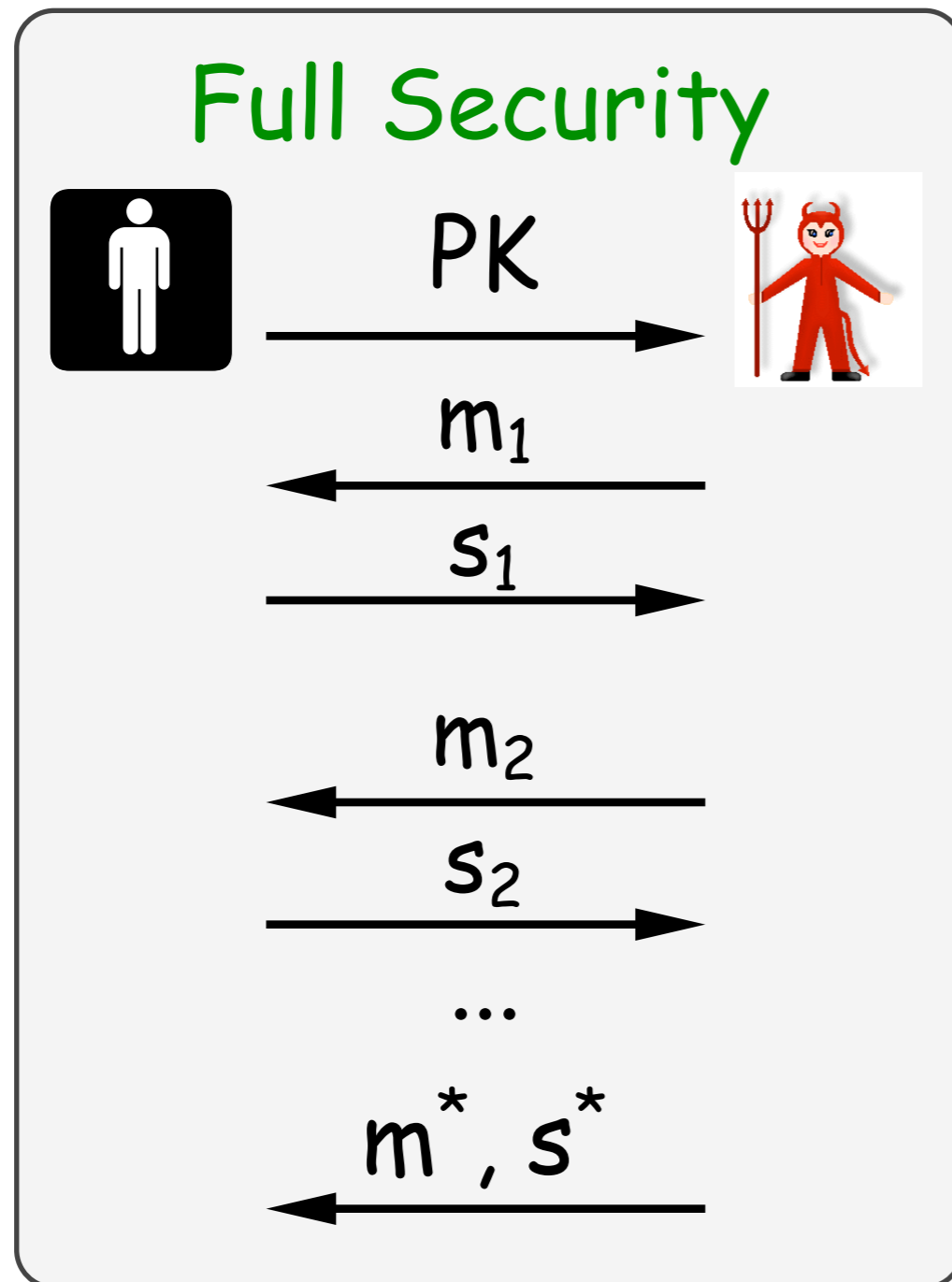
Immediate: a digital signature scheme:

- under the RSA assumption
- standard model
- stateless
- short signatures (1 element, 1 integer)
- short public keys (modulus, 1 element, hash parameters)

Longer-term: a technique for:

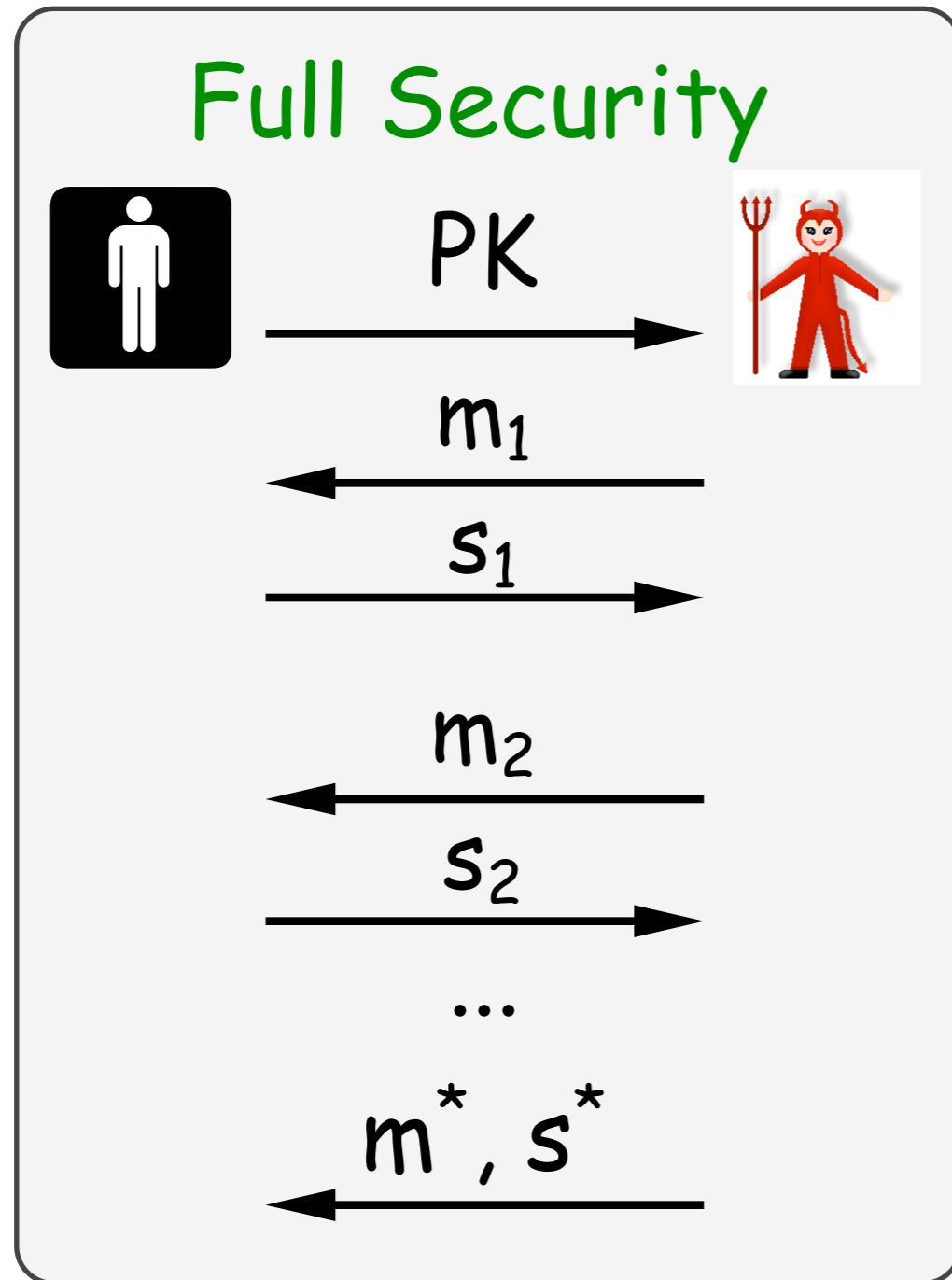
- designing short, standard model signatures
- non-generic path from selective to full security

# Goldwasser-Micali-Rivest Definition



Negligible probability that  $\text{Verify}(PK, m^*, s^*)=1$  and  $m^*$  is new.

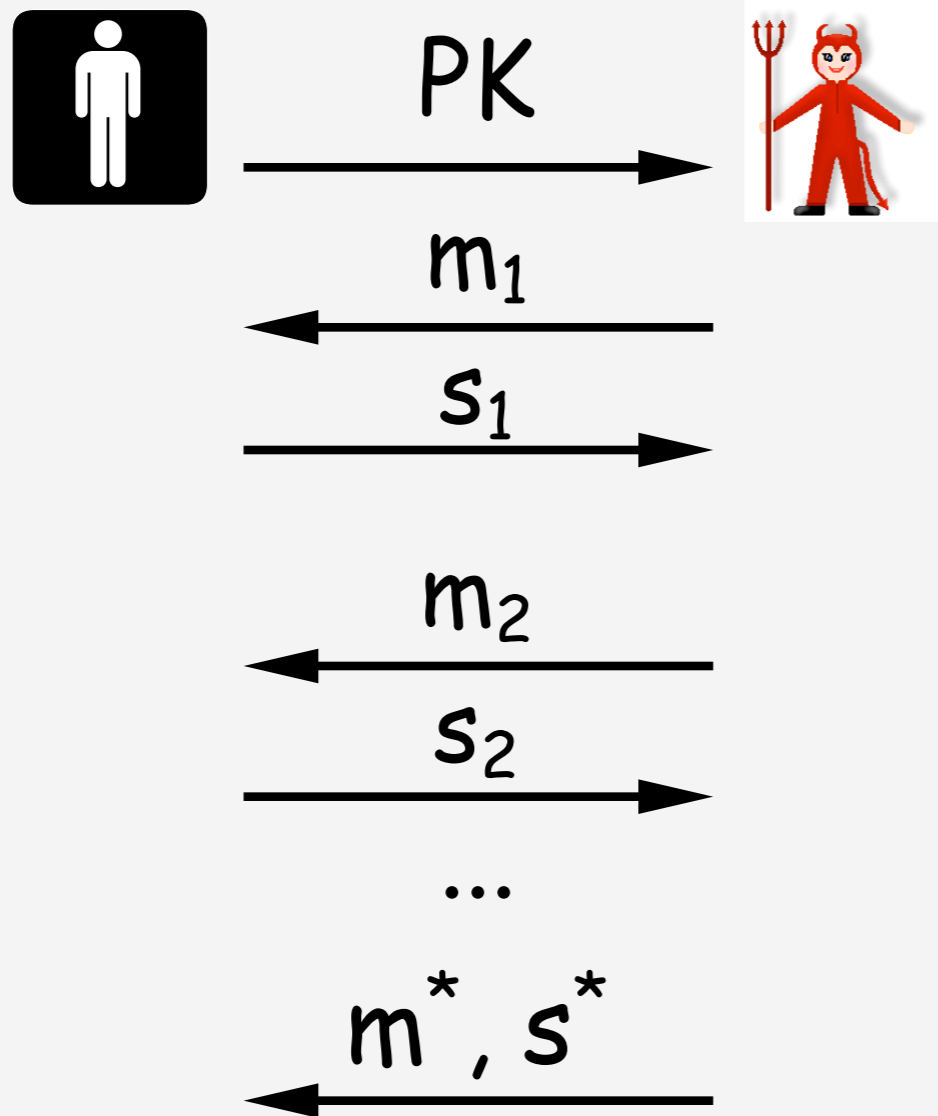
# Goldwasser-Micali-Rivest Definition



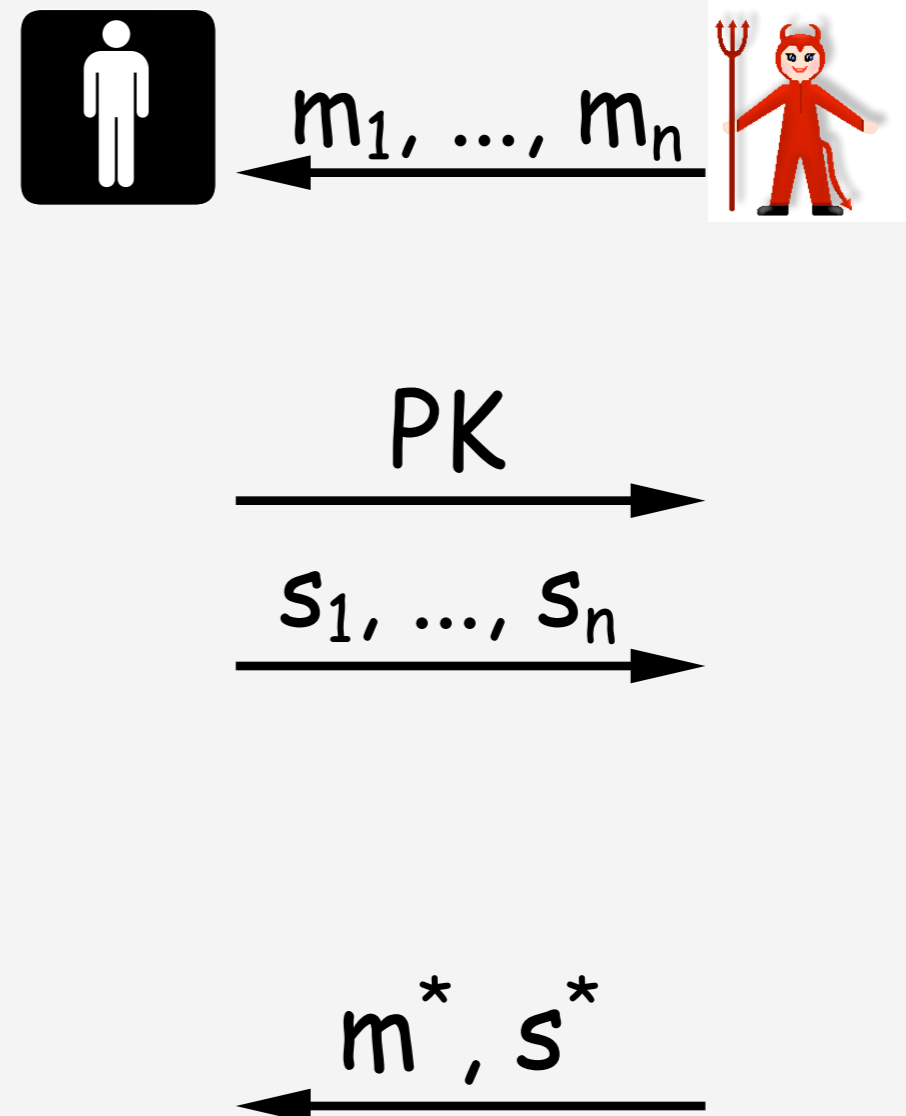
Proofs are tricky. How to answer all queries, except  $m^*$ ?

# Definitions of Security

## Full Security



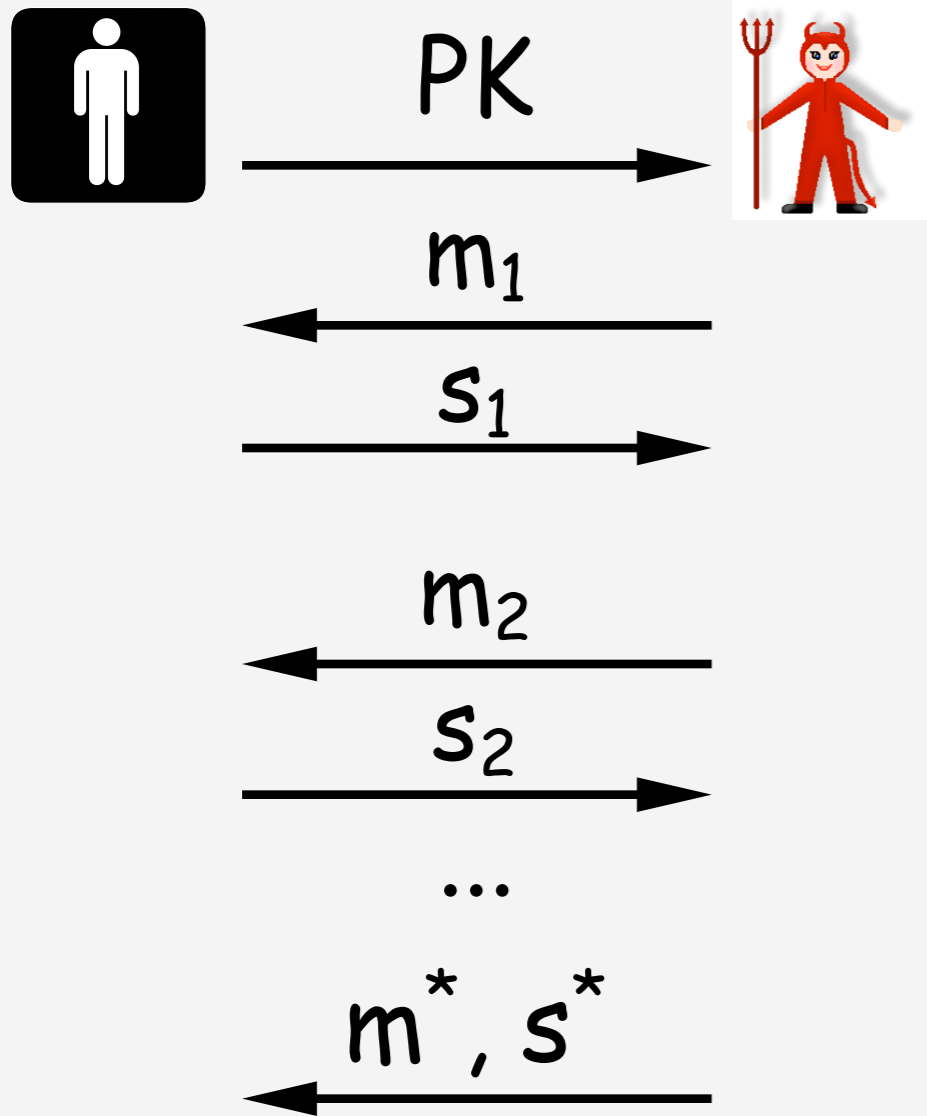
## Weak Security



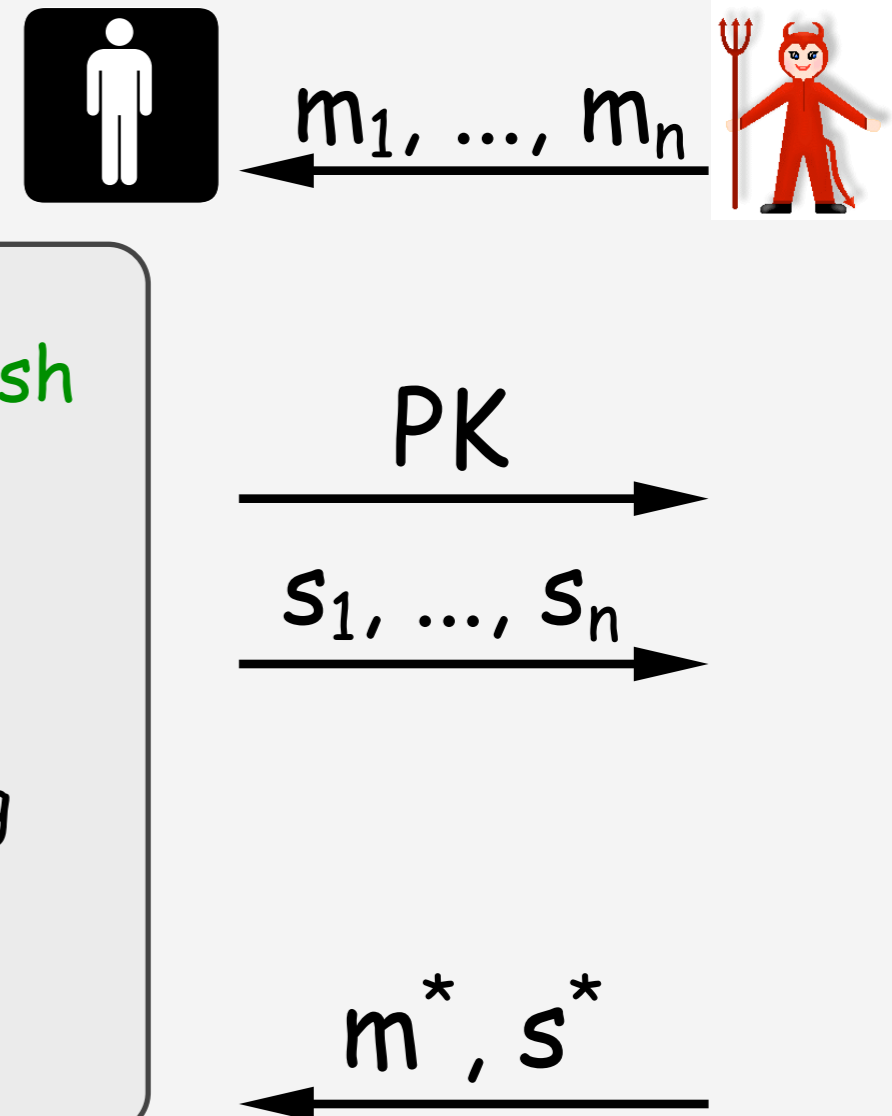
Negligible probability that  $\text{Verify}(PK, m^*, s^*)=1$  and  $m^*$  is new.

# Definitions of Security

## Full Security



## Weak Security



## Chameleon Hash

- exist under
- factoring
  - RSA
  - discrete log

$\Leftarrow =$

Theorem [GHR99,ST01]: Full Signatures  $\Leftarrow =$  Chameleon Hash + Weak Signatures.



# Gennaro-Halevi-Rabin Weak Sigs

Public Key:  $N$ ,  $h$ ,  $H: \{0,1\}^* \rightarrow \text{primes}$ .

Sign:  $s := h^{1/H(m)} \bmod N$ .

Verify: Accept iff  $h = s^{H(m)} \bmod N$ .

---

---

# Gennaro-Halevi-Rabin Weak Sigs

Public Key:  $N, h, H: \{0,1\}^* \rightarrow \text{primes}$ .

Sign:  $s := h^{1/H(m)} \bmod N$ .

Verify: Accept iff  $h = s^{H(m)} \bmod N$ .

---

**Strong RSA:** Given  $(N,y)$ , find **any**  $(x,e)$  s.t.  $e > 1$  and  $x^e = y \bmod N$ .

---

# Gennaro-Halevi-Rabin Weak Sigs

Public Key:  $N, h, H: \{0,1\}^* \rightarrow \text{primes}$ .

Sign:  $s := h^{1/H(m)} \bmod N$ .

Verify: Accept iff  $h = s^{H(m)} \bmod N$ .

---

Strong RSA: Given  $(N,y)$ , find any  $(x,e)$  s.t.  $e > 1$  and  $x^e = y \bmod N$ .

---

Proof sketch. Adversary gives  $m_1, \dots, m_q$ .

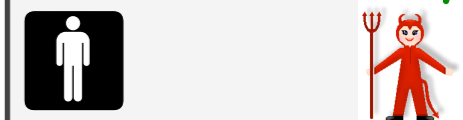
Set  $h := y^{e_1 e_2 \dots e_q} \bmod N$ , where  $H(m_i) = e_i$ .

To sign  $m_i$ , leave  $e_i$  out of product.

On forgery,  $s^{*e^*} = h = y^{e_1 e_2 \dots e_q}$ , where  $H(m^*) = e^*$ .

Use Shamir's trick to get  $x$  s.t.  $x^{e^*} = y \bmod N$ .

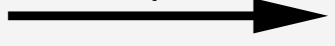
Weak Security



$m_1, \dots, m_q$



PK



$s_1, \dots, s_q$



$m^*, s^*$



# Gennaro-Halevi-Rabin Weak Sigs

Public Key:  $N, h, H: \{0,1\}^* \rightarrow \text{primes}$ .

Sign:  $s := h^{1/H(m)} \bmod N$ .

Verify: Accept iff  $h = s^{H(m)} \bmod N$ .

**Strong** RSA: Given  $(N,y)$ , find **any**  $(x,e)$  s.t.  $e > 1$  and  $y = x^e \bmod N$ .

No idea where to embed single  $e$ , so push issue to the assumption.

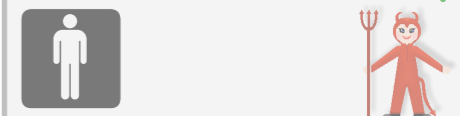
gives  $m_1, \dots, m_q$ .

where  $H(m_i) = e_i$ .

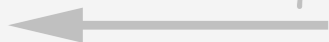
On  $m^*$ , we have  $e_1, \dots, e_q$ , where  $H(m^*) = e^*$ .

Use Shamir's trick to get  $x$  s.t.  $x^{e^*} = y \bmod N$ .

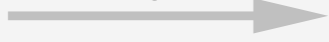
Weak Security



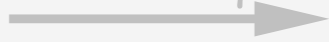
$m_1, \dots, m_q$



PK



$s_1, \dots, s_q$



$m^*, s^*$



# Gennaro-Halevi-Rabin Weak Sigs

Public Key:  $N, h, H: \{0,1\}^* \rightarrow \text{primes}$ .

Sign:  $s := h^{1/H(m)} \bmod N$ .

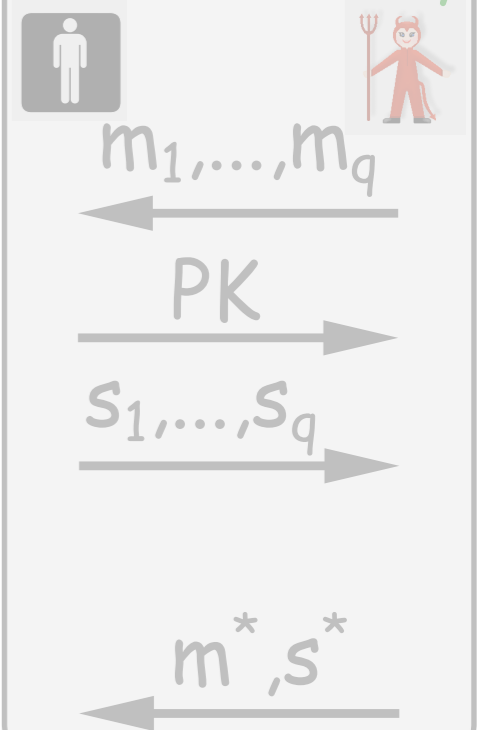
Verify: Accept iff  $h = s^{H(m)} \bmod N$ .

**RSA:** Given  $(N, y, e^*)$ , find **the**  $x$  s.t.  $e > 1$  and  $y = x^{e^*} \bmod N$ .

Pr... gives  $m_1, \dots, m_q$ .  
... where  $H(m_i) = e_i$ .  
...  
On...  $e_2 \dots e_q$ , where  $H(m^*) = e^*$ .  
Use Shamir's trick to get  $x$  s.t.  $x^{e^*} = y \bmod N$ .

If we knew  $m^*$ ,  
we could program  $H$   
with single RSA  $e^*$ .  
... what do we know  
about  $m^*$ ??

## Weak Security



# A New Technique for Designing

S

Si

Sig

Sign

Signa

Signat

Signatu

Signatur

Signature

Signatures

# What about $m^*$ ?

## Weak Security



$m_1, \dots, m_q$



PK



$s_1, \dots, s_q$



$m^*, s^*$



# What about $m^*$ ?

## Weak Security



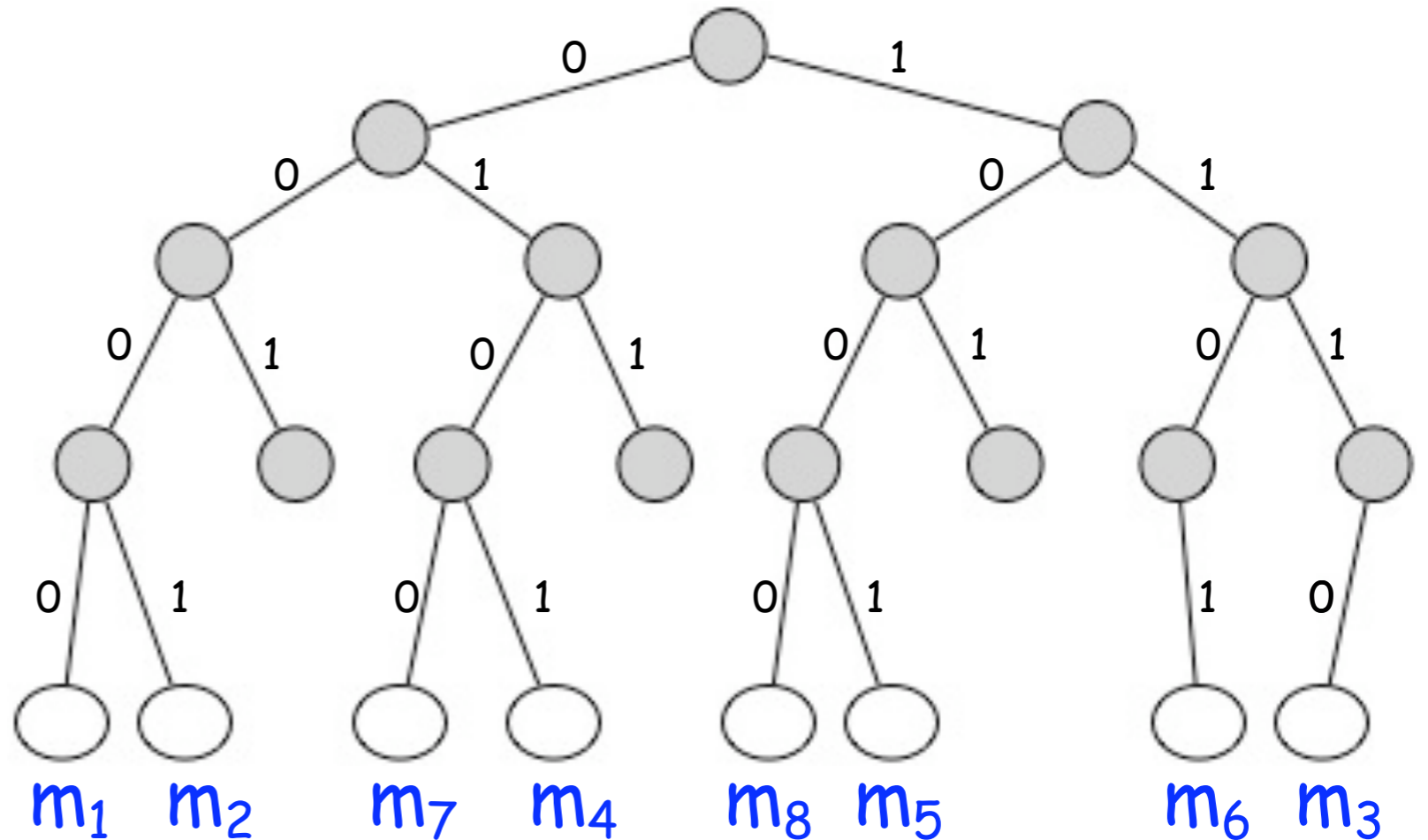
$m_1, \dots, m_q$



PK

$s_1, \dots, s_q$

$m^*, s^*$





# What about $m^*$ ?

## Weak Security



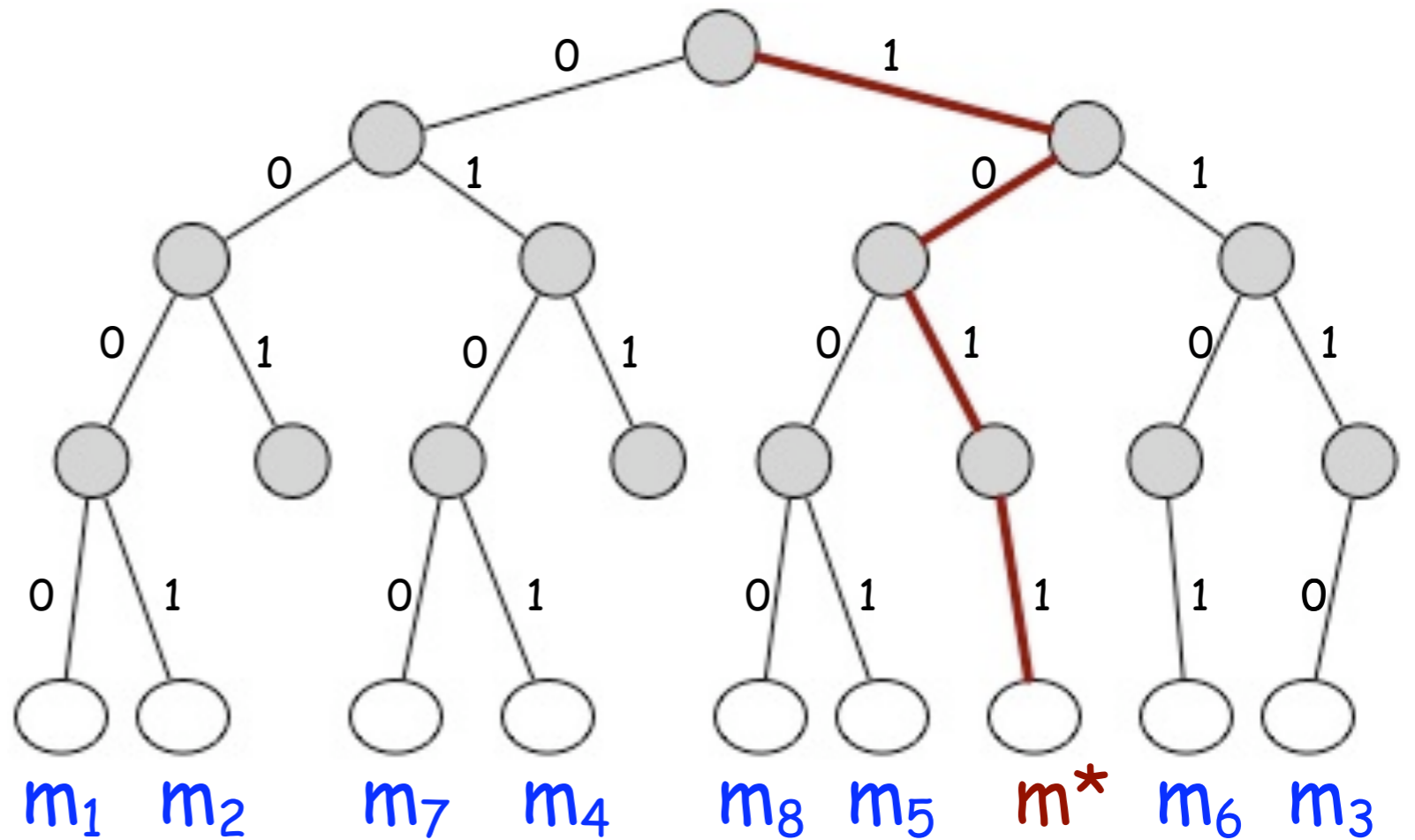
$m_1, \dots, m_q$



PK

$s_1, \dots, s_q$

$m^*, s^*$



# What about $m^*$ ?

## Weak Security



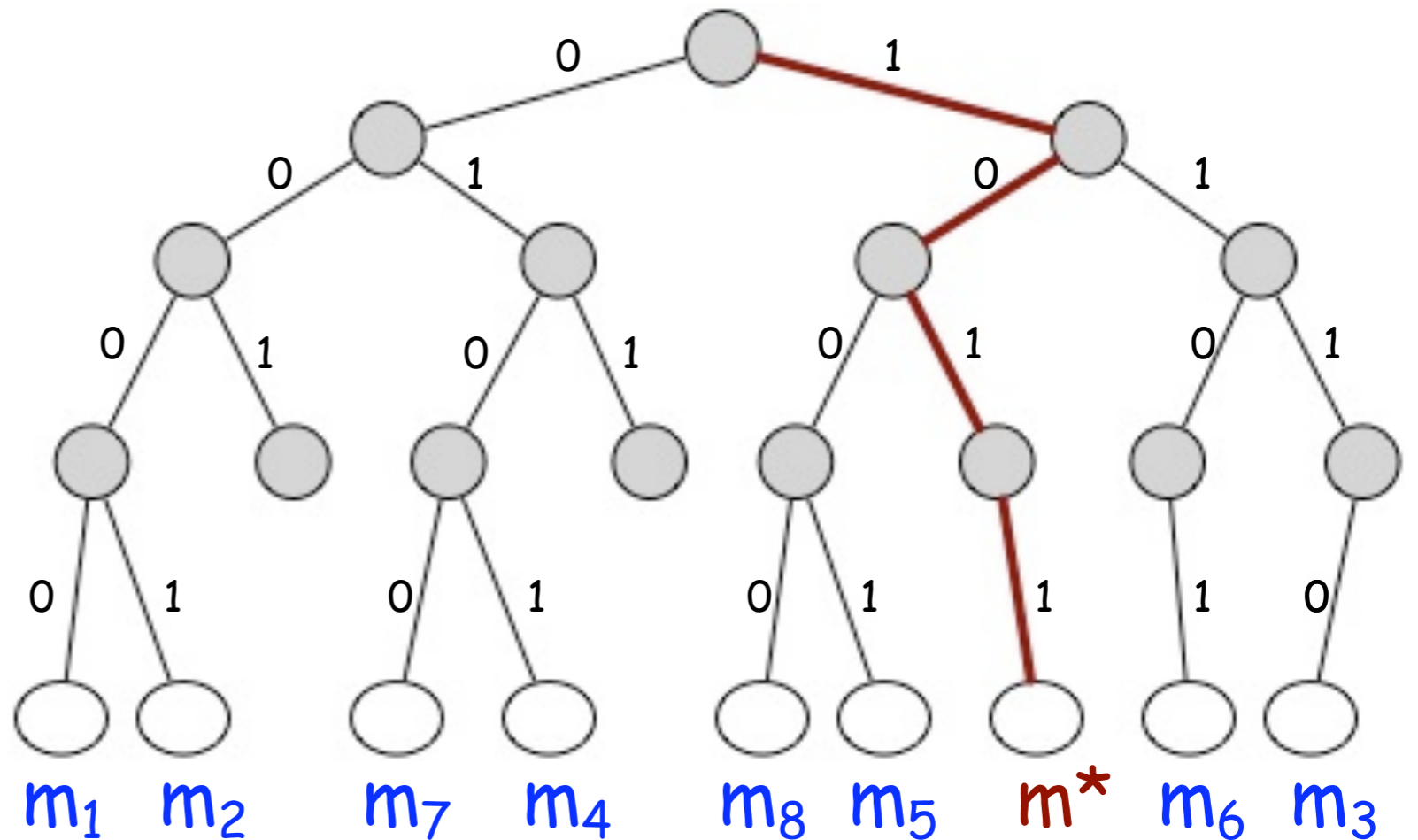
$m_1, \dots, m_q$



PK

$s_1, \dots, s_q$

$m^*, s^*$



Shortest unique prefix of  $m^* = 101$ .

# What about $m^*$ ?

## Weak Security



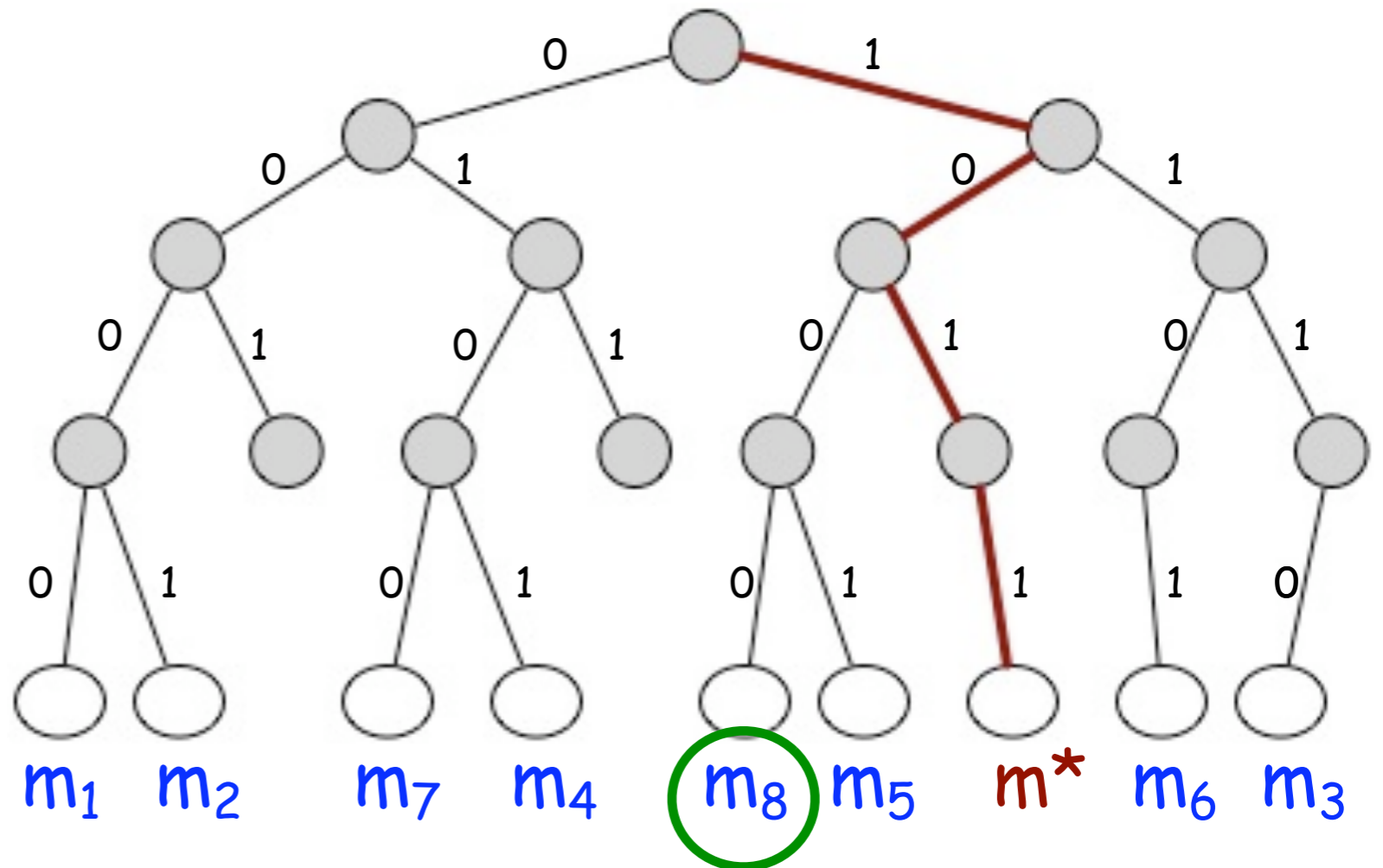
$m_1, \dots, m_q$



PK

$s_1, \dots, s_q$

$m^*, s^*$



Shortest unique prefix of  $m^* = 101$ .

IDEA: Guess this prefix (before seeing  $m^*$ ).

-- guess  $m_i$  which  $m^*$  follows longest:  $\geq 1/q$  chance.

# What about $m^*$ ?

## Weak Security



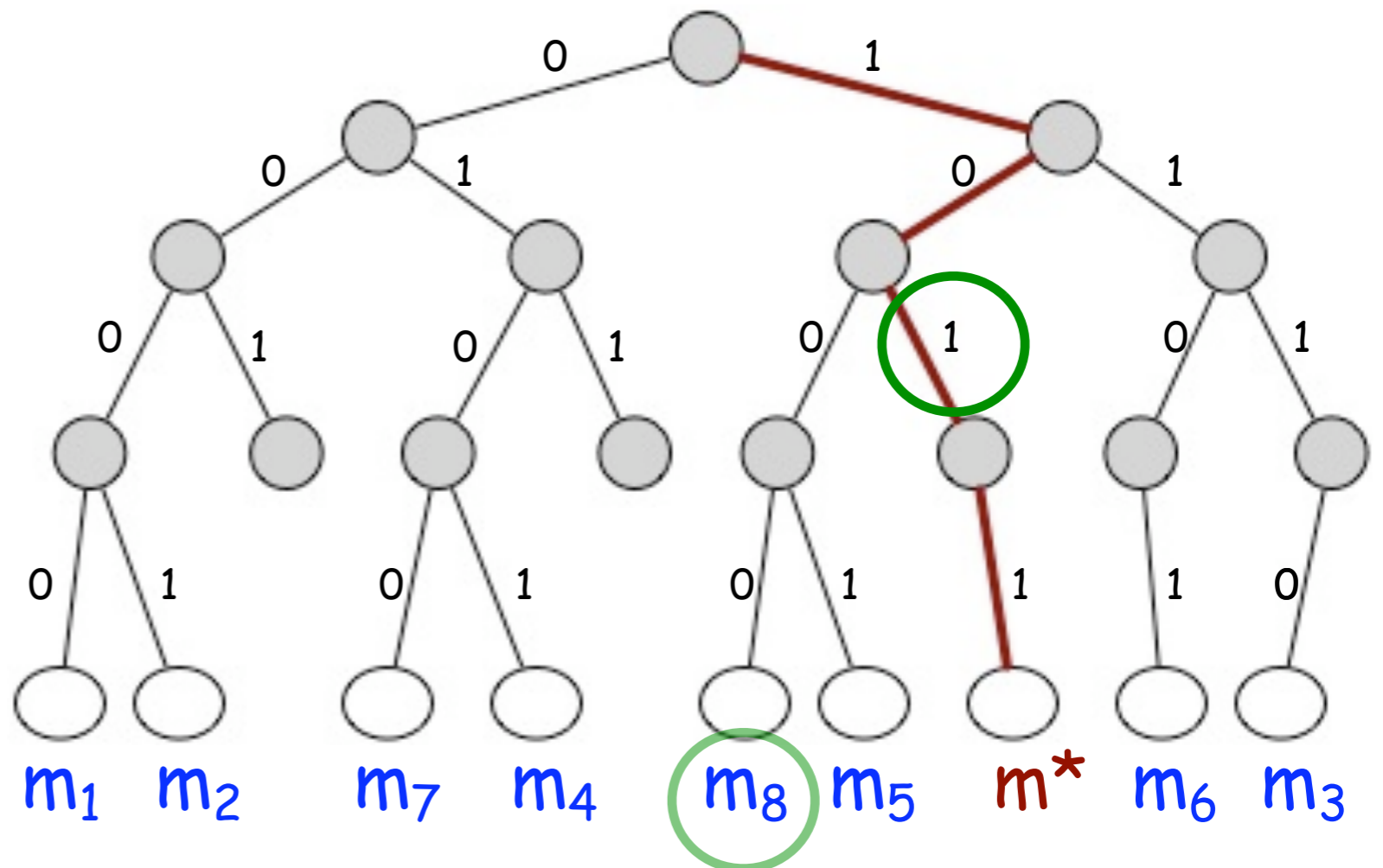
$m_1, \dots, m_q$



PK

$s_1, \dots, s_q$

$m^*, s^*$

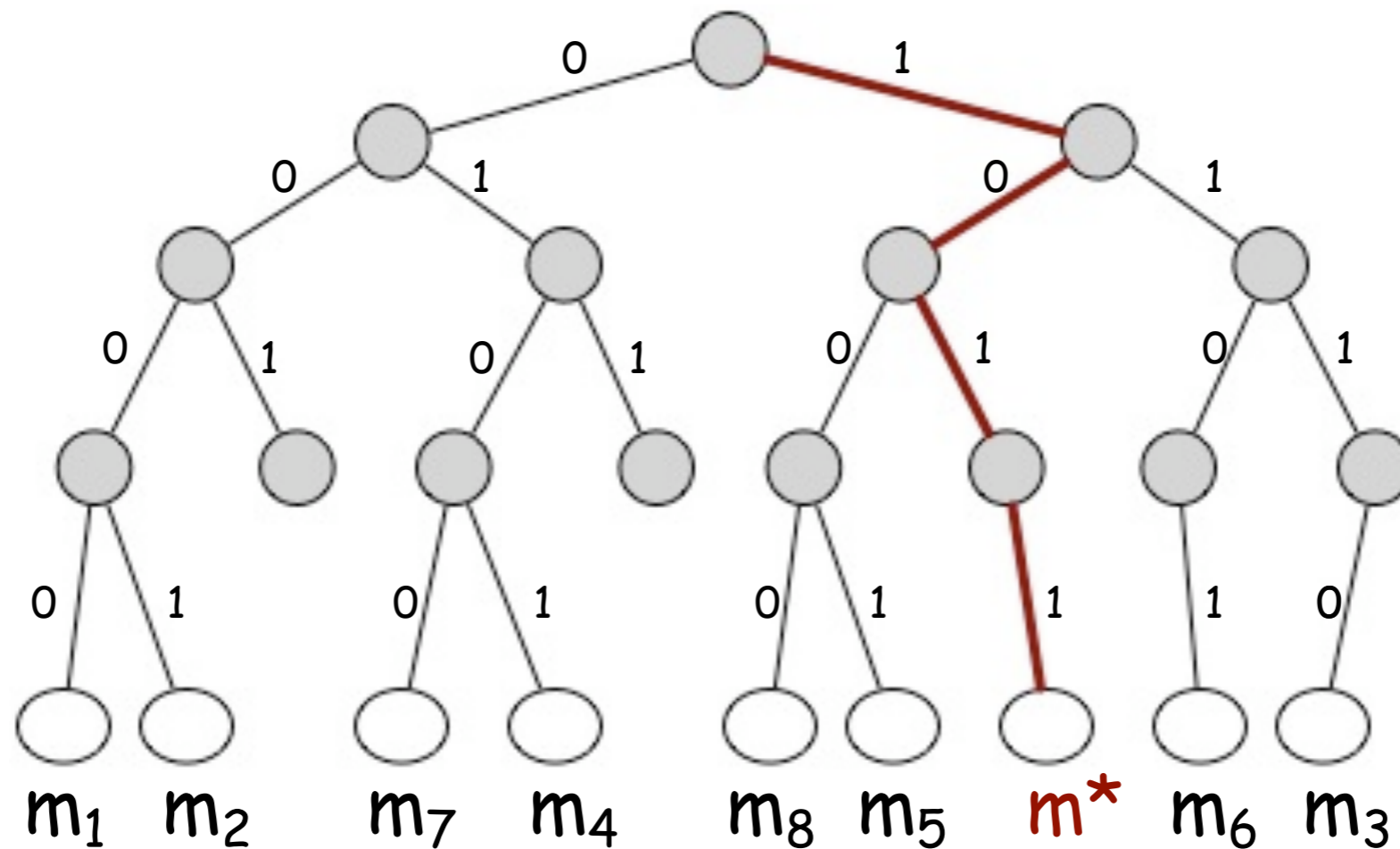


Shortest unique prefix of  $m^* = 101$ .

IDEA: Guess this prefix (before seeing  $m^*$ ).

- guess  $m_i$  which  $m^*$  follows longest:  $\geq 1/q$  chance.
- guess first bit where  $m^*$  differs:  $\geq 1/n$  chance.

# What to do with observation?



Shortest unique prefix of  $m^* = 101$

1. Design signature using all prefixes of message.
2. Guess unique prefix of  $m^*$  to embed challenge.

# RSA Construction

**Public Key:**  $N, h$ , and  $H: \{0,1\}^* \rightarrow \text{primes}$ .

**Sign:** Let  $M^i :=$  first  $i$  bits of  $M$ .

$$s := h^{1/e_1 e_2 \dots e_n} \bmod N, \text{ where } e_i := H(M^i).$$

**Verify:** Accept iff  $h = s^{e_1 e_2 \dots e_n} \bmod N$ , where  $e_i := H(M^i)$ .

$$\text{GHR: } s := h^{1/H(M)} \bmod N$$

# RSA Construction

Public Key:  $N$ ,  $h$ , and  $H: \{0,1\}^* \rightarrow \text{primes}$ .

Sign: Let  $M^i :=$  first  $i$  bits of  $M$ .

$s := h^{1/e_1 e_2 \dots e_n} \bmod N$ , where  $e_i := H(M^i)$ .

Verify: Accept iff  $h = s^{e_1 e_2 \dots e_n} \bmod N$ , where  $e_i := H(M^i)$ .

---

**RSA:** Given  $(N, y, e)$ , find the  $x$  s.t.  $e > 1$  and  $y = x^e \bmod N$ .

---

# RSA Construction

Public Key:  $N, h$ , and  $H: \{0,1\}^* \rightarrow \text{primes}$ .

Sign: Let  $M^i :=$  first  $i$  bits of  $M$ .

$s := h^{1/e_1 e_2 \dots e_n} \bmod N$ , where  $e_i := H(M^i)$ .

Verify: Accept iff  $h = s^{e_1 e_2 \dots e_n} \bmod N$ , where  $e_i := H(M^i)$ .

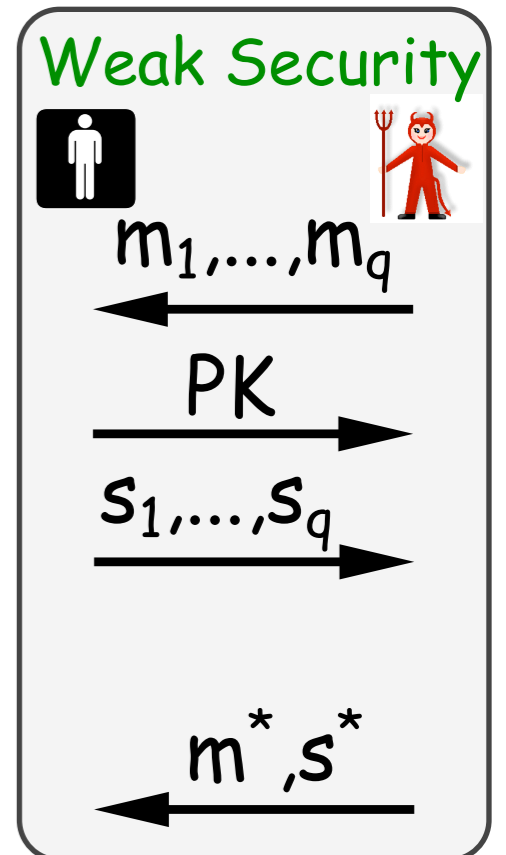
---

**RSA:** Given  $(N, y, e)$ , find the  $x$  s.t.  $e > 1$  and  $y = x^e \bmod N$ .

---

Proof sketch. Adversary gives  $M_1, \dots, M_q$ .

1. Guess  $w^*$  as shortest unique prefix of  $M^*$ .
2. Choose  $H$  so that  $H(w^*) = e$ .
3.  $h := y$  (product of hash of all prefixes of  $M_1, \dots, M_q$ ).
4. Sign for  $M_1, \dots, M_q$  by omit from product.
5. Extract  $x$  from  $M^*$  forgery by Shamir's Trick.





# Performance



**Public Key:**  $O(1)$  elements ( $N, h, \text{hash descriptions}$ )

**Signature:** 1 element in  $Z_N^*$ , 1 integer

---

**Signing:** 1 exp.  $E(\text{primality tests}) = nk.$

**Verification:**  $n$  exp.  $E(\text{primality tests}) = nk.$

$n = \text{length of message}, k = \text{security parameter}$

# Performance



**Public Key:**  $O(1)$  elements ( $N, h, \text{hash descriptions}$ )

**Signature:** 1 element in  $Z_N^*$ , 1 integer

---

**Signing:** 1 exp.  $E(\text{primality tests}) = nk.$

**Verification:**  $n$  exp.  $E(\text{primality tests}) = nk.$

$n = \text{length of message}, k = \text{security parameter}$



## Optimizations?

1. use larger alphabet for prefixes

-  $v$  bit chunk  $\Rightarrow n/v$  primes, but security loss of  $1/(2^v - 1)$ .

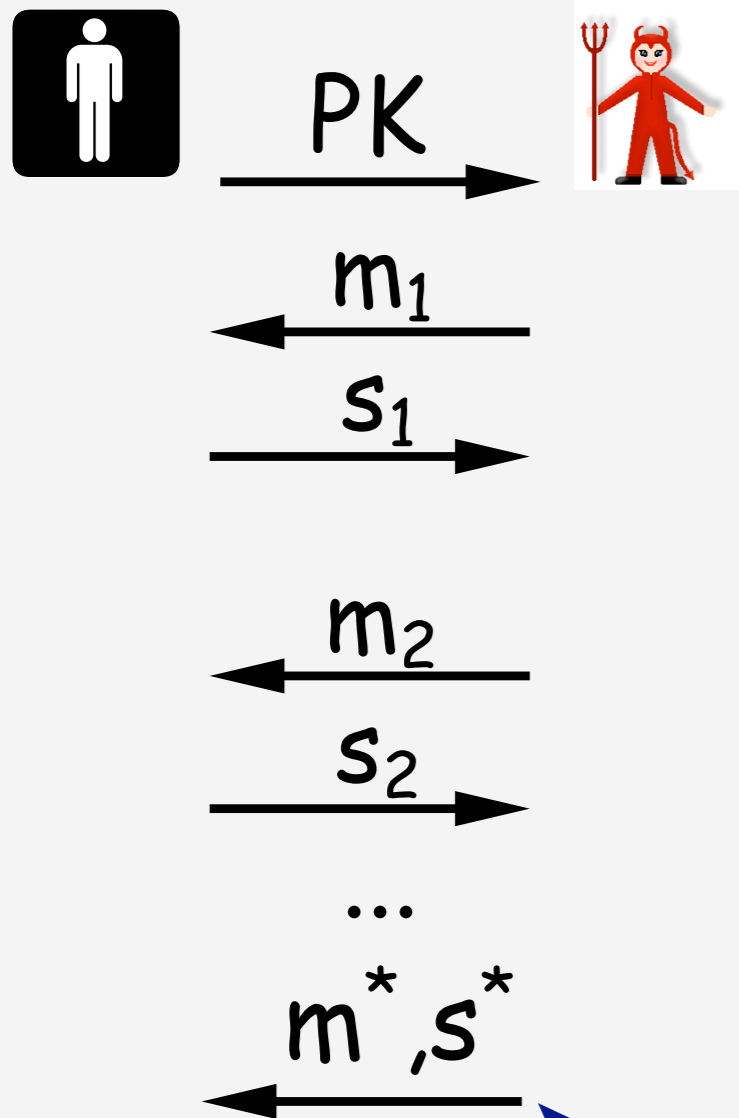
2. hash to smaller primes

- good idea, slightly changes the RSA assumption.

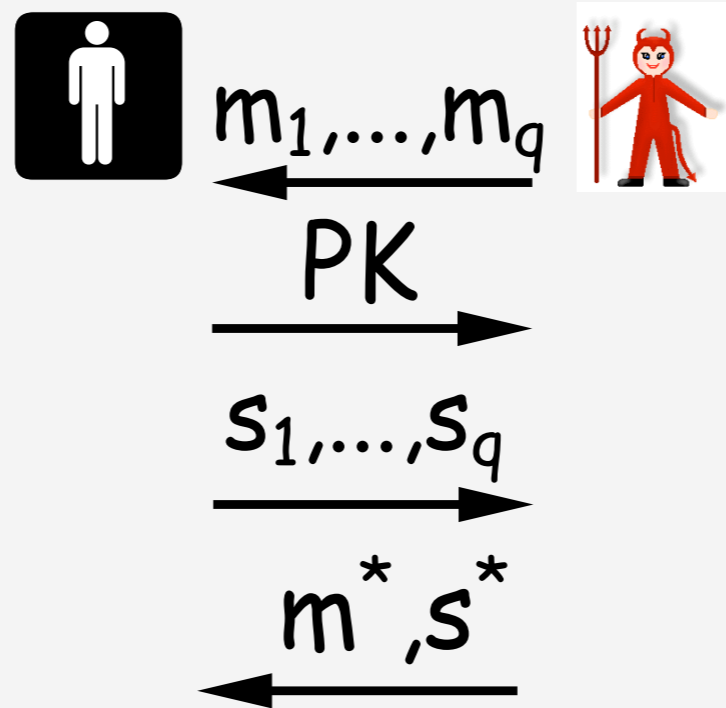
3. more in paper....

# Definitions of Security

## Full Security



## Weak Security



+

## Chameleon Hash

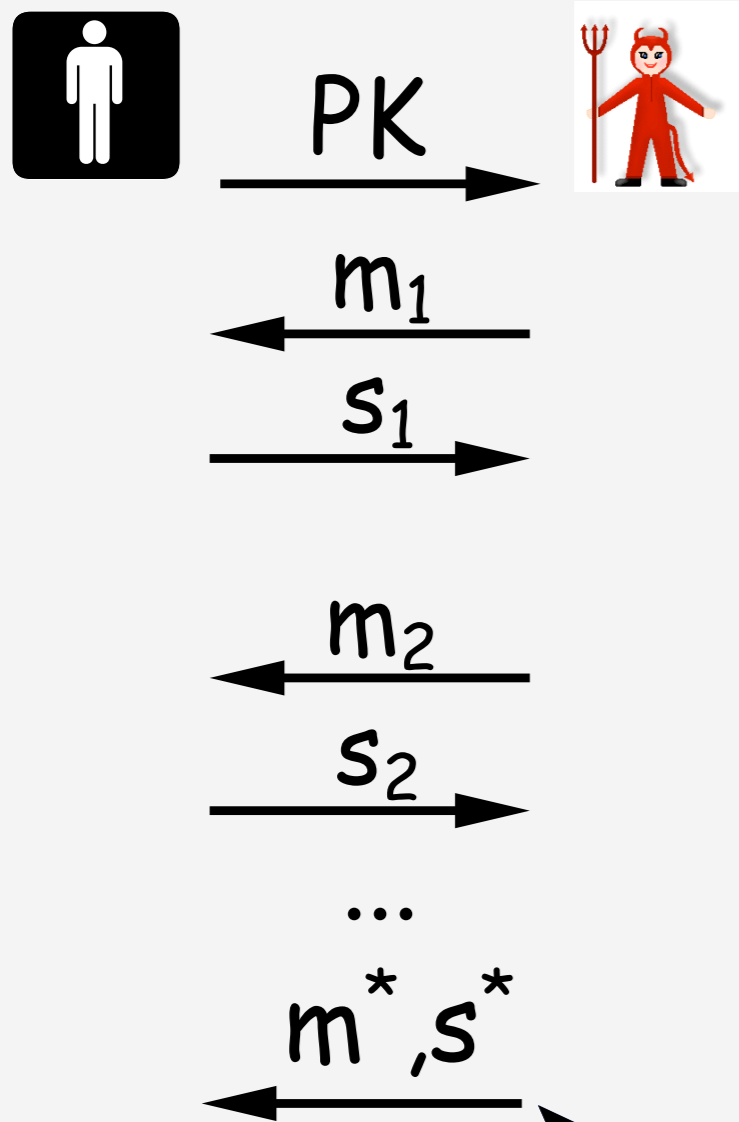
exist under

- factoring
- RSA
- discrete log

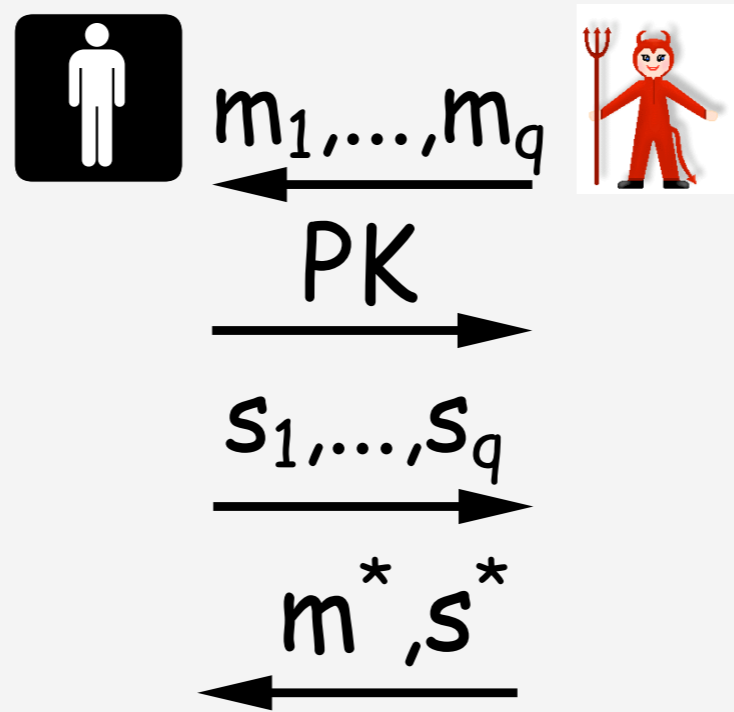
Theorem [GHR99,ST01]: Full Signatures  $\Leftarrow$  Chameleon Hash + Weak Signatures.

# Definitions of Security

## Full Security



## Weak Security

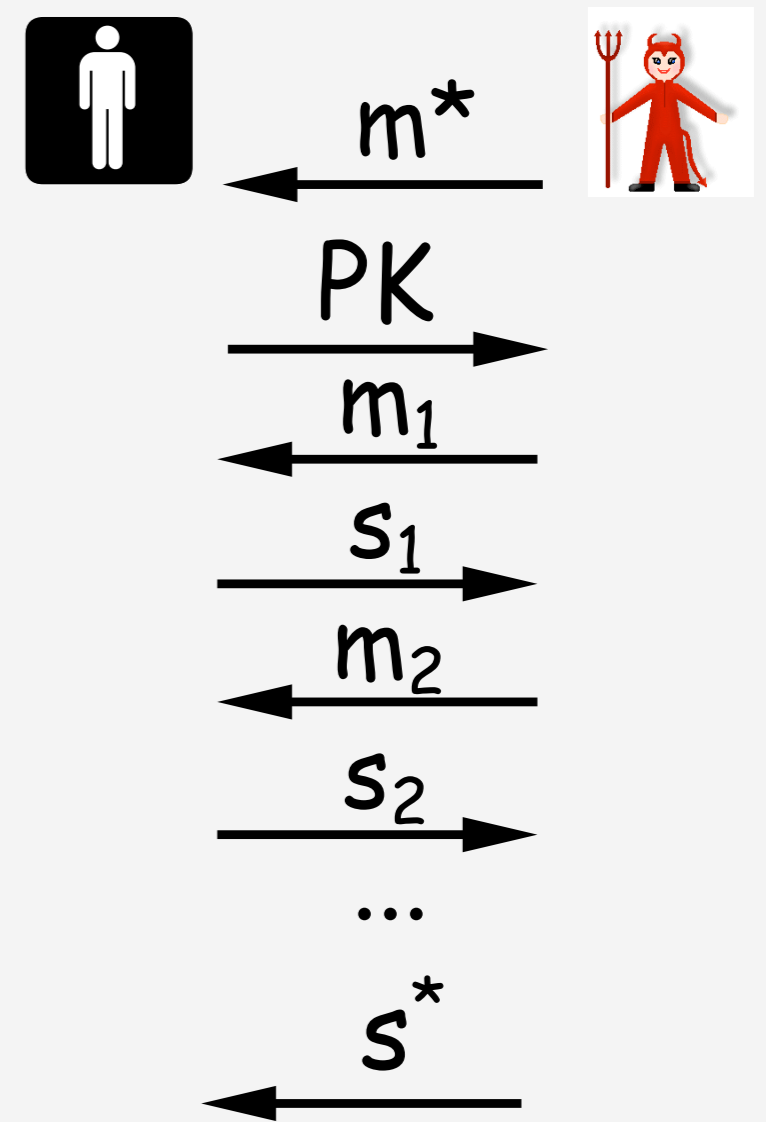


+

## Chameleon Hash

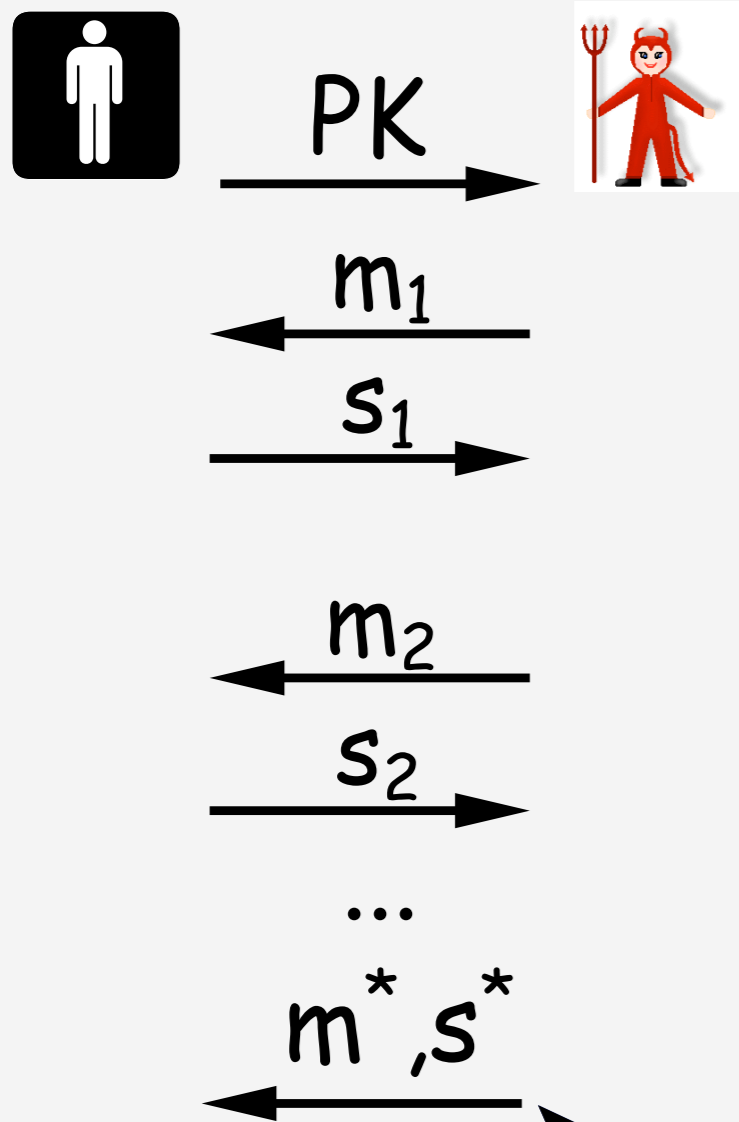
- exist under
- factoring
  - RSA
  - discrete log

## Selective Security

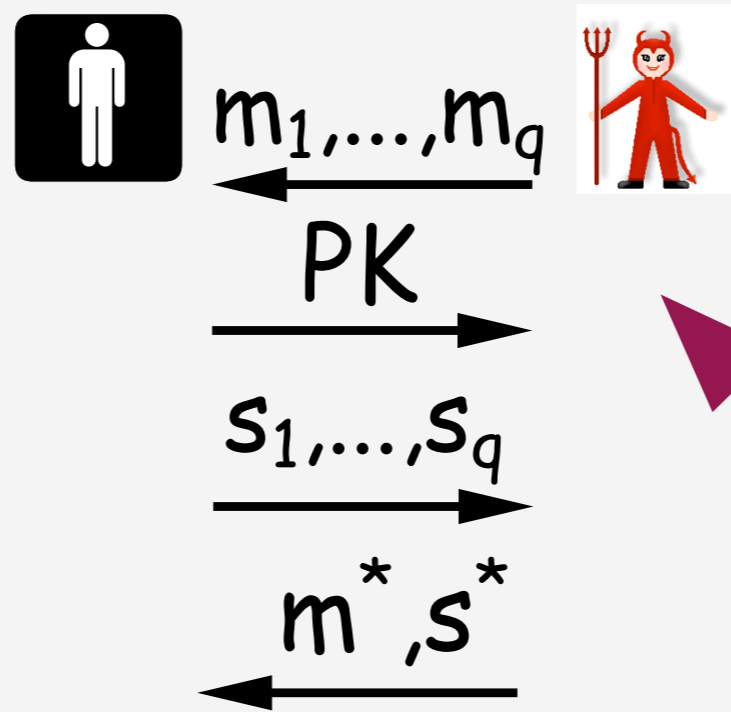


# Definitions of Security

## Full Security



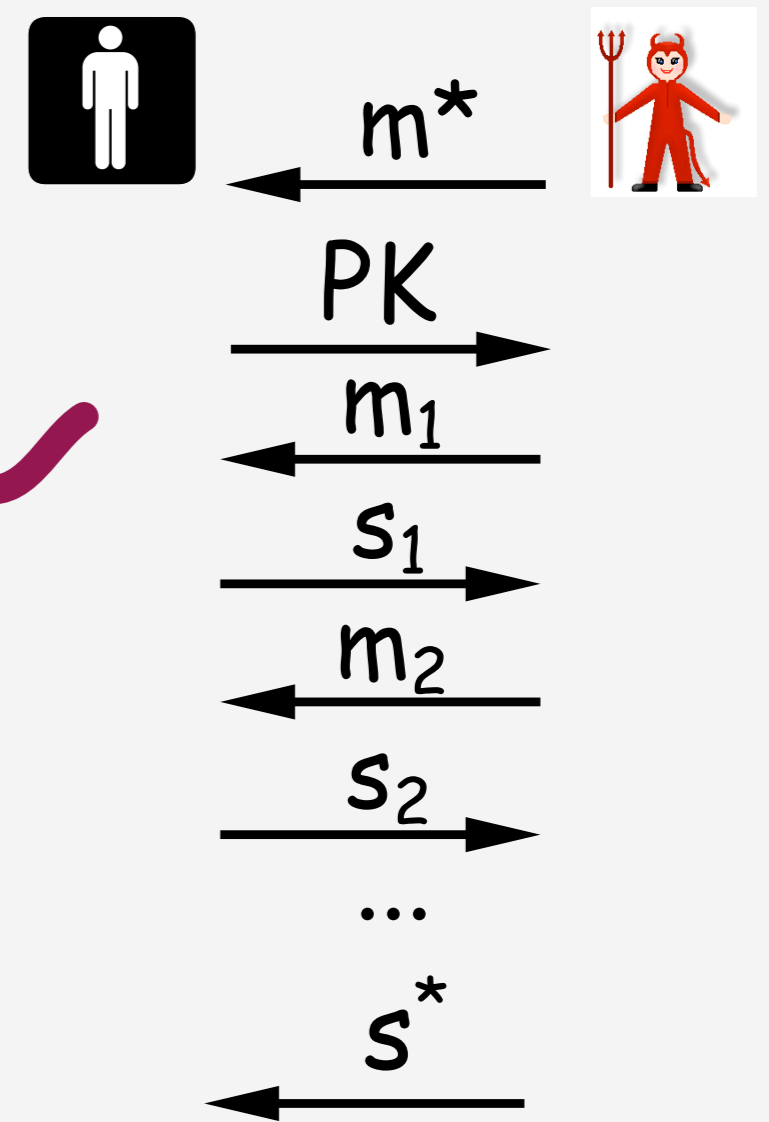
## Weak Security



## Chameleon Hash

- exist under
- factoring
  - RSA
  - discrete log

## Selective Security

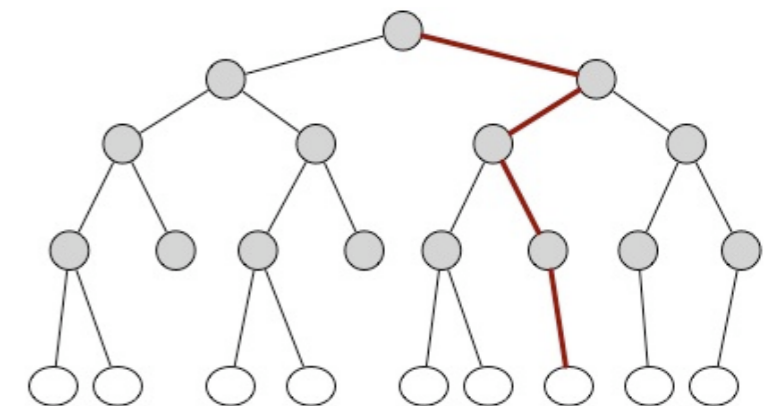


[HW09b]: a non-generic technique for selective to weak security.

# Wider Application of Technique

We give a new proof for Waters signatures under **CDH**.

---



# Wider Application of Technique

We give a new proof for Waters signatures under **CDH**.

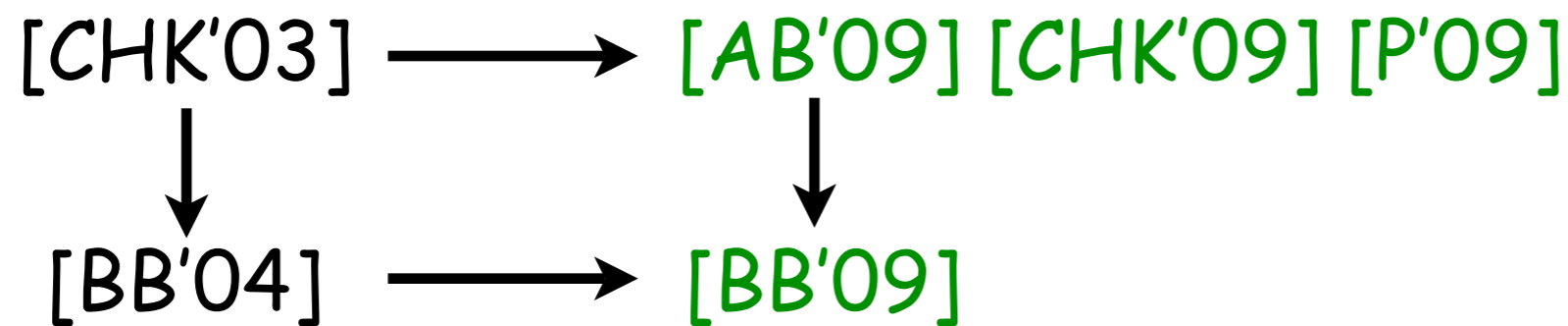
---

Selectively-secure IBE from Lattices.

Pairing world

Lattice world

ID processing

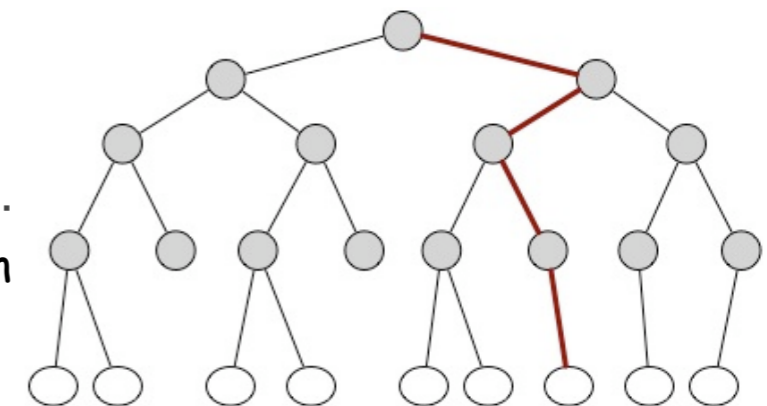


bit by bit

all at once

---

[CHK'03] - Canetti, Halevi, Katz, [BB'04] - Boneh, Boyen, [AB'09] - Agrawal, Boyen  
[CHK'09] - Cash, Hofheinz, Kiltz, [P'09] - Peikert, [BB'09] - Boneh, Boyen



# Wider Application of Technique

We give a new proof for Waters signatures under **CDH**.

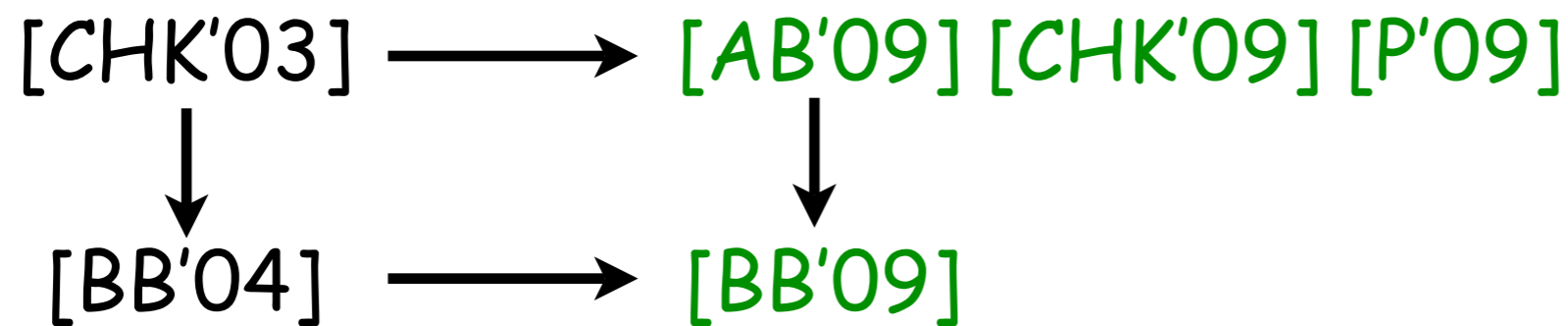
---

Selectively-secure IBE from **Lattices**.

Pairing world

Lattice world

ID processing



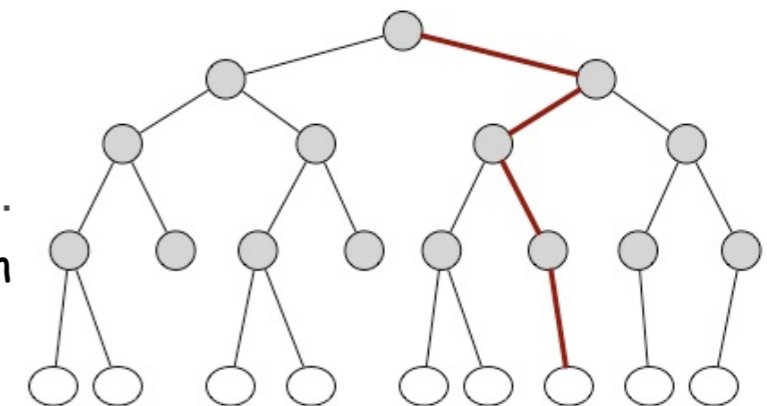
bit by bit

all at once

Admits **selectively-secure signatures** [Naor].

---

[CHK'03] - Canetti, Halevi, Katz, [BB'04] - Boneh, Boyen, [AB'09] - Agrawal, Boyen  
[CHK'09] - Cash, Hofheinz, Kiltz, [P'09] - Peikert, [BB'09] - Boneh, Boyen





# Wider Application of Technique

We give a new proof for Waters signatures under **CDH**.

---

Selectively-secure IBE from **Lattices**.

Pairing world

Lattice world

ID processing

[CHK'03]



[AB'09] [CHK'09] [P'09]



[BB'04]



[BB'09]

????

bit by bit

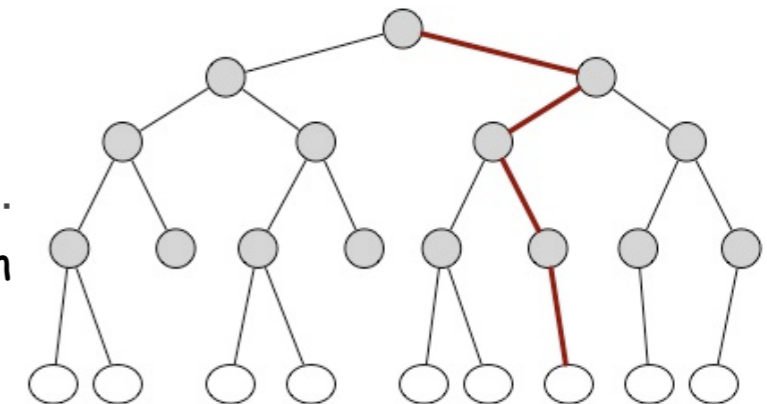
all at once

Admits selectively-secure signatures [Naor].

Apply prefix technique to get **full signatures!**

---

[CHK'03] - Canetti, Halevi, Katz, [BB'04] - Boneh, Boyen, [AB'09] - Agrawal, Boyen  
[CHK'09] - Cash, Hofheinz, Kiltz, [P'09] - Peikert, [BB'09] - Boneh, Boyen



# Open Directions

1. Better performance under RSA.
2. General selective to full security technique.
3. Short, standard model signatures from
  - discrete logarithm
  - CDH without bilinear groups
4. Standard model/assumptions for:
  - anonymous credentials
  - electronic cash
  - aggregate signatures
  - etc.

