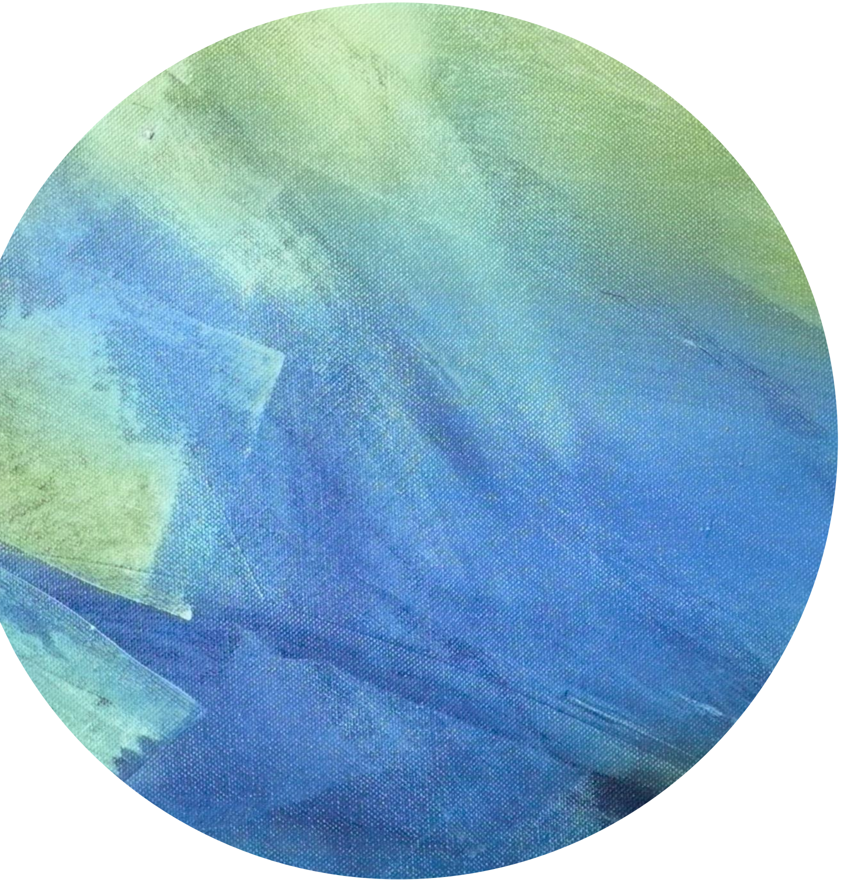
Lecture 12

Scribe : Mircea

Outline



~~Schnorr Signatures~~

Diffie-Hellman

~~Commitments~~

RSA

PKE can be used to establish a shared secret

Alice **Bob**

(pk, sk)

m

("Alice", pk) $\longrightarrow$

choose random
$x \in \{0,1\}^{128}$

$\longleftarrow$ $ct = Enc_{pk}(m; x)$

$Dec_{sk}(ct) \to m$

How to Build
Public Key Encryption

Recall: The Diffie-Hellman protocol

Fix a finite cyclic group G (e.g $G = (Z_p)^*$) of order n

Fix a generator g in G (i.e. $G = \{1, g, g^2, g^3, \dots, g^{n-1}\}$)

Alice's public key

choose **a** ← {1,…,p-1}

"Alice", $A \leftarrow g^a \pmod p$

"Bob", $B \leftarrow g^b \pmod p$

choose **b** ← {1,…,p-1}

$g^{ab} = B^a$

$g^{ab} = A^b$

Computational DH : Given (g^a, g^b), hard to find g^{ab}

Decisional DH :

Convert to PKE?

$$(g^a, g^b, g^{ab}) \approx (g^a, g^b, g^c)$$

$$a, b, c \leftarrow \{1, \dots p-1\}$$

choose **a** ← {1,…,p-1}

sk

g^a : Alice's public key

m

b

$\cancel{b}$

$$g^b, H(g^{ab}) \oplus m$$

ASSUMPTION: $(g^a, g^b, H(g^{ab})) \approx (g^a, g^b, \text{uniform})$

El-Gamal Encryption

El-Gamal is a public-key encryption system (Gen, Enc, Dec):

- Key generation Gen: $pk = g^a, \quad sk = a \qquad a \leftarrow \{1, \ldots, p-1\}.$

- Enc$(\underbrace{pk}_{= g^a}, m)$ = Sample b

  Output $ct = (g^b, H(g^{ab}) \oplus m)$.

- Dec$(\underbrace{sk}_{a}, ct)$ $\hookrightarrow (\overbrace{g^b}^{c_1}, \overbrace{H(g^{ab}) \oplus m}^{c_2})$ $\rightarrow$ Compute $g^{ab} = (g^b)^a$.

  $H(g^{ab}) \oplus c_2 = m$

El-Gamal Encryption

Why is this secure?

Recall: Semantic security (CPA - secure)

Computational Diffie-Hellman Assumption

G: finite cyclic group of order n

Comp. DH (CDH) assumption holds in G if: $g,\ g^a,\ g^b \quad \not\Rightarrow \quad g^{ab}$

for all efficient algs. A:

$$\Pr\Big[\ A(g, g^a, g^b) = g^{ab}\ \Big] < \text{negligible}$$

where $g \longleftarrow \{\text{generators of } G\}$, $\quad a, b \longleftarrow Z_n$

Hash Diffie-Hellman Assumption

G: finite cyclic group of order n , $\quad$ H: $G^2 \longrightarrow K$ a hash function

Def: Hash-DH (HDH) assumption holds for (G, H) if:

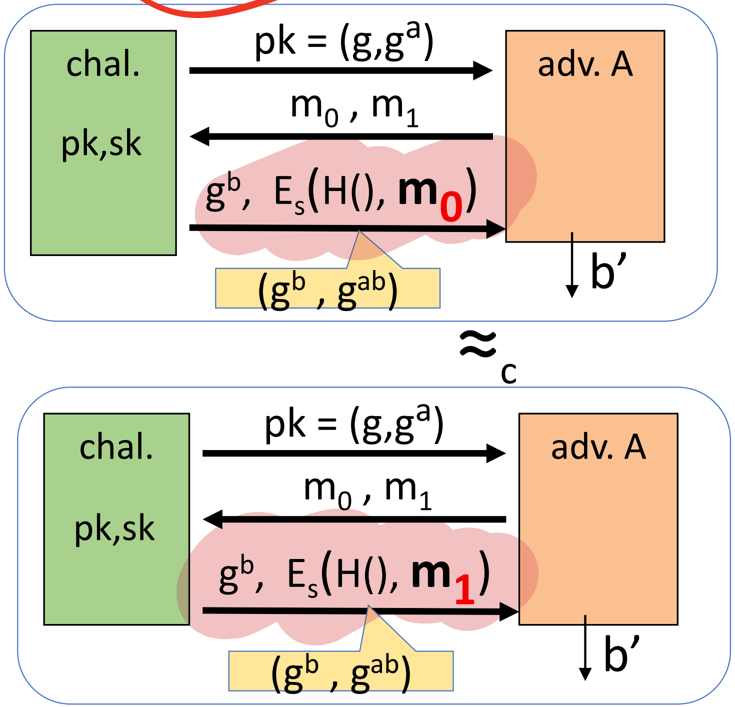
$$\left(g,\ g^a,\ g^b,\ H(g^b, g^{ab})\right) \approx_p \left(g,\ g^a,\ g^b,\ R\right)$$

where $g \leftarrow \{\text{generators of } G\}$, $\quad a, b \leftarrow Z_n$, $\quad R \leftarrow K$

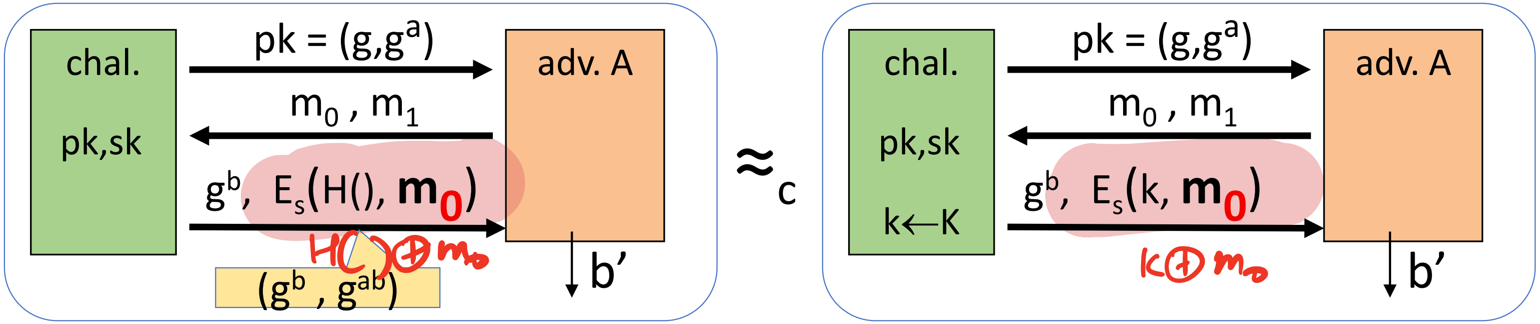
H acts as an extractor: strange distribution on $G^2 \Rightarrow$ uniform on K

HDH => El-Gamal is Semantically Secure

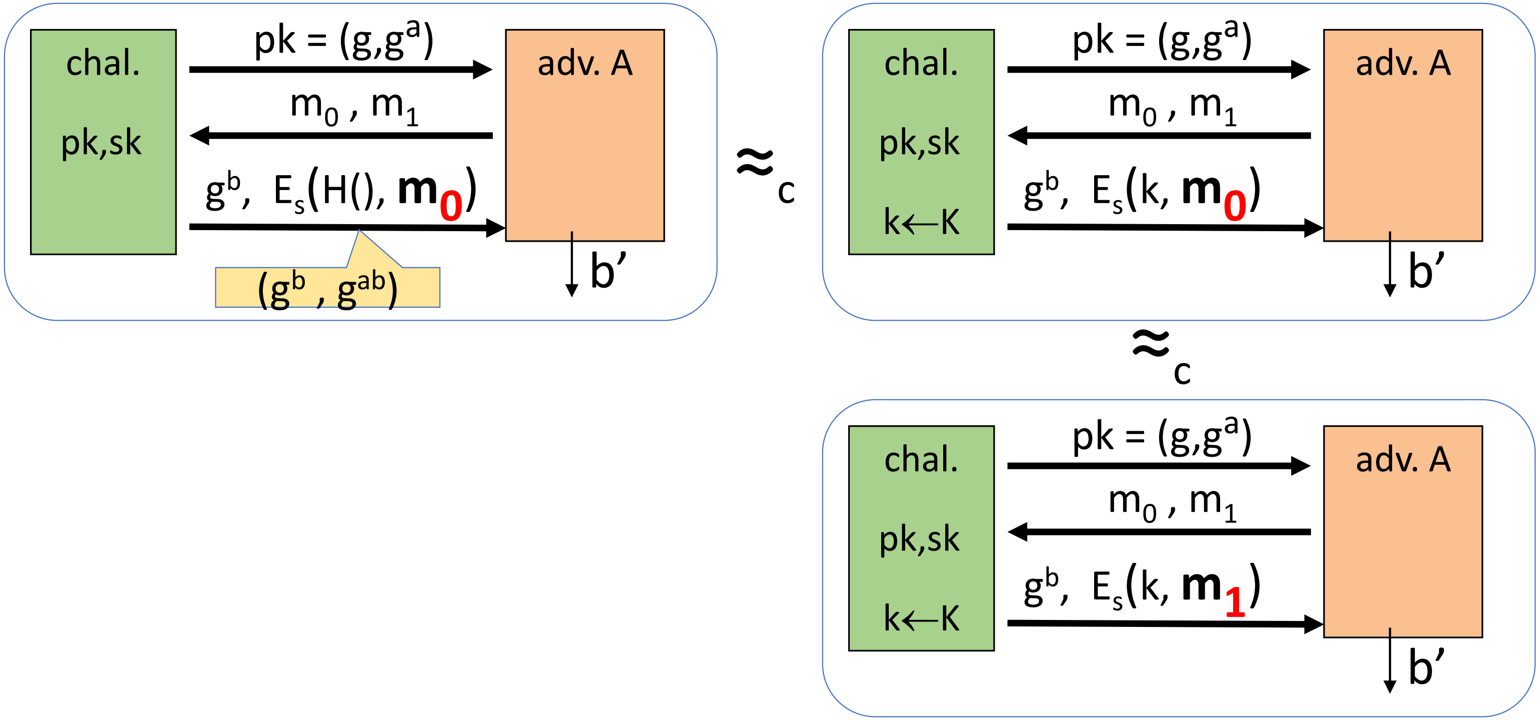
Goal



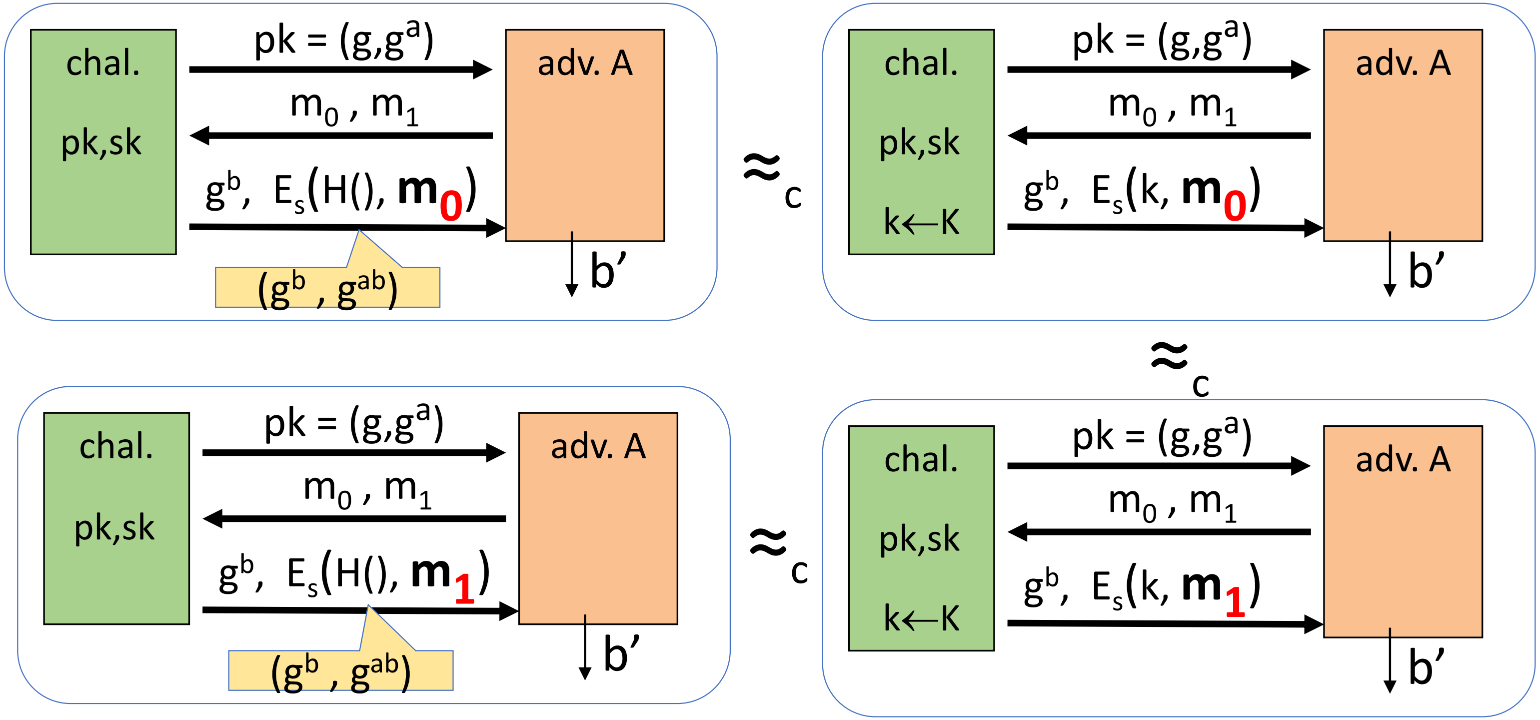
HDH => El-Gamal is Semantically Secure



HDH => El-Gamal is Semantically Secure



HDH => El-Gamal is Semantically Secure



The RSA Cryptosystem

Review: arithmetic mod composites

$|p| = |q| = 2048$ bits

Let $N = p \cdot q$ where p,q are prime

$Z_N = \{0,1,2,\ldots,N-1\}$; $(Z_N)^* = \{\text{invertible elements in } Z_N\}$

$\{1, 2, 3 \ldots p-1, p+1, \ldots q-1, q+1 \ldots 2p-1, 2p+1, \ldots\}$

<u>Facts</u>: $x \in Z_N$ is invertible $\Leftrightarrow$ $\gcd(x,N) = 1$

- Number of elements in $(Z_N)^*$ is $\varphi(N) = (p-1)(q-1) = N-p-q+1$

<u>Euler's thm</u>: $\boxed{\forall\, x \in (Z_N)^* \ : \ x^{\varphi(N)} = 1 \text{ in } Z_N^*}$

Textbook RSA System

Gen(.): choose random primes p,q ≈1024 bits. Set **N=pq**.

choose integers **e , d** s.t. **e·d = 1 (mod φ(N))** (random)

output pk = (N, e) , sk = (N, d)

RSA-Enc (pk, x) = x^e = y (in Z_N) $x \in Z_N$

RSA-Dec (~~pk~~ sk, y) = y^d ~~(in Z_N)~~ → should give x.

$$y^d = x^{ed} = x^{k\varphi(N)+1} = \left(x^{\varphi(N)}\right)^k \cdot \mathbf{x} = (1)^k \cdot \mathbf{x} = x$$

Textbook RSA System

Weak game

Ch $\qquad\qquad$ A

$\xrightarrow{e}$

for $x \leftarrow \{2, \dots N-1\}$ $\qquad \xrightarrow{x^e \text{ in } Z_N}$

Guess x?

Let's analyze this:

RSA-Enc (pk, x) = x^e $\qquad$ (in Z_N)

RSA-Dec (pk, y) = y^d $\qquad$ (in Z_N)

$$y^d = x^{ed} = x^{k\varphi(N)+1} = (x^{\varphi(N)})^k \cdot \mathbf{x} = (1)^k \cdot \mathbf{x} = x$$

~~Insecure cryptosystem !!~~

Is not semantically secure and many attacks exist

Deterministic!

So what is this?

Trapdoor Permutation

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

$= \#$ elements in Z_N^*

Trapdoor Permutation

$\forall e$, unique d s.t.

$ed = 1 \mod \varphi(N)$

Three algorithms: (G, F, F^{-1})

- G: outputs pk, sk. pk defines a function $F(pk, \cdot): X \to X$

  e , d

  $e \quad x \longrightarrow x^e$

- F(pk, x): evaluates the function at x

  $d, x^e \longrightarrow x$

- F^{-1}(sk, y): inverts the function at y using sk

Secure trapdoor permutation:

The function F (pk, ·) is one-way without the trapdoor sk

Ch A

$\xrightarrow{pk = e}$

random x

$\xrightarrow{x^e}$

Hard to find x.

The RSA assumption

RSA assumption: RSA is one-way permutation

For all efficient algs. A:

$$\Pr\left[\ A(N,e,y) = y^{1/e}\ \right] < \text{negligible}$$

where $p,q \xleftarrow{R}$ n-bit primes, $N \leftarrow pq$, $y \xleftarrow{R} Z_N^*$

But hardness of factoring does not imply hardness of RSA.

Hardness of the RSA assumption

relies on the hardness of factoring

Consider the set of integers: (e.g. for n=1024)

$$\{\ N = p \cdot q \quad \text{where} \quad (p,q) \ \text{ are n-bit primes}\ \}$$

Problem: Factor a random element in the set (e.g. for n=1024) HARD!

Recall RSA assumpn: Given pk = (e, N) and y, *find y^d where $\boxed{d = e^{-1} \pmod{\varphi(N)}}$*

If you could factor N → find (p, q) → compute $\varphi(N) = (p-1)(q-1) = N-p-q+1$

→ compute $d = e^{-1} \pmod{\varphi(N)}$ → break RSA

Textbook RSA System

Gen(.): **pk** = (N, e), **sk** = (N, d) such that **e·d = 1 (mod φ(N))**

RSA-Enc (pk, x) = x^e (in Z_N)

RSA-Dec (pk, y) = y^d (in Z_N)

Is not semantically secure

How would you make it semantically secure?

Enc(pk, x) pick r

$(x \| r)^e$

Speeding up RSA

To speed up RSA use a small e: $c = m^e \pmod N$

- Minimum value: **$e=3$** $(\gcd(e, \varphi(N)) = 1)$
- Recommended value: **$e=65537=2^{16}+1$**

Encryption: 17 multiplications

<u>Asymmetry of RSA:</u> fast enc. / slow dec.

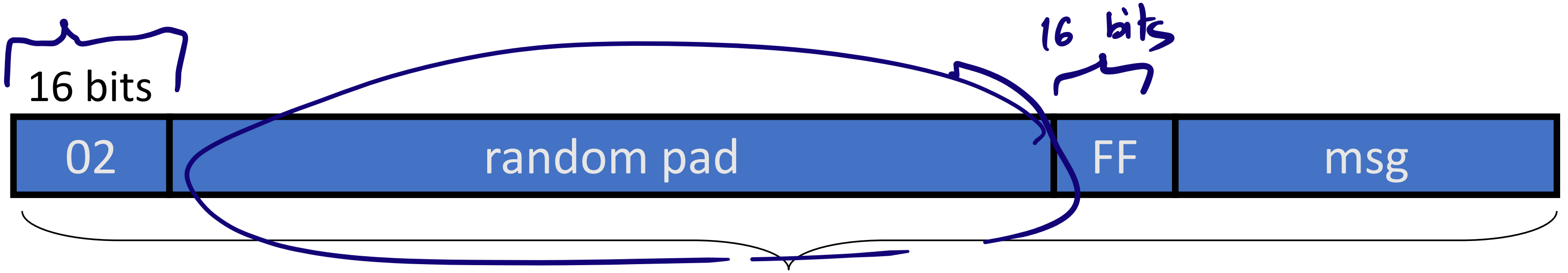
- ElGamal (next module): approx. same time for both.

RSA in practice: PKCS1 v1.5

$\left(\overbrace{\text{pad}}^{} \,\|\, m\right)^e$

insecure under CCA.

PKCS1 mode 2: (encryption)

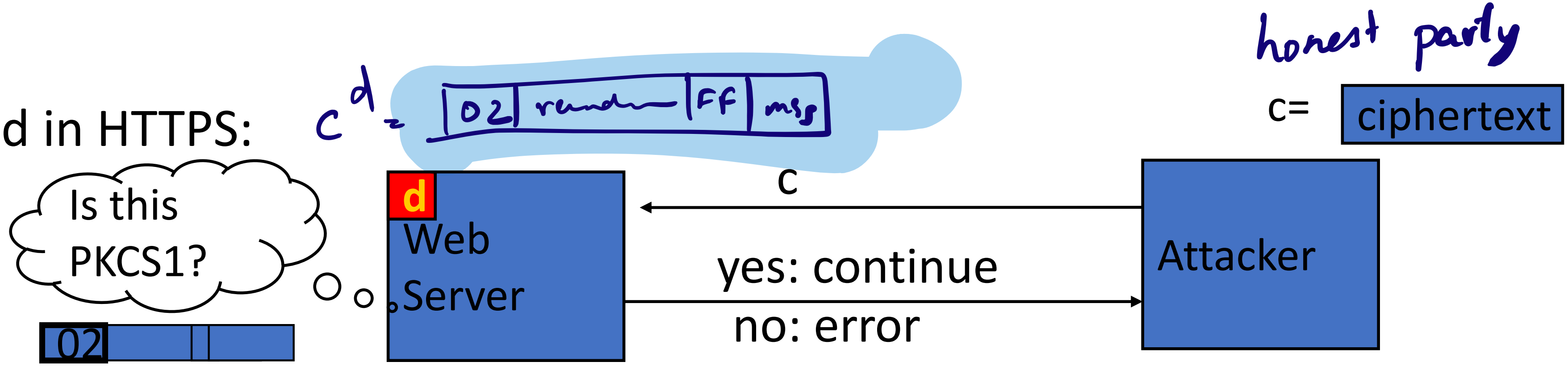


RSA modulus size (e.g. 2048 bits)

- Resulting value is RSA encrypted
- Widely deployed, e.g. in HTTPS
- Suffered from a CCA attack

CCA Attack on PKCS1 v1.5 (Bleichenbacher 1998)

PKCS1 used in HTTPS:



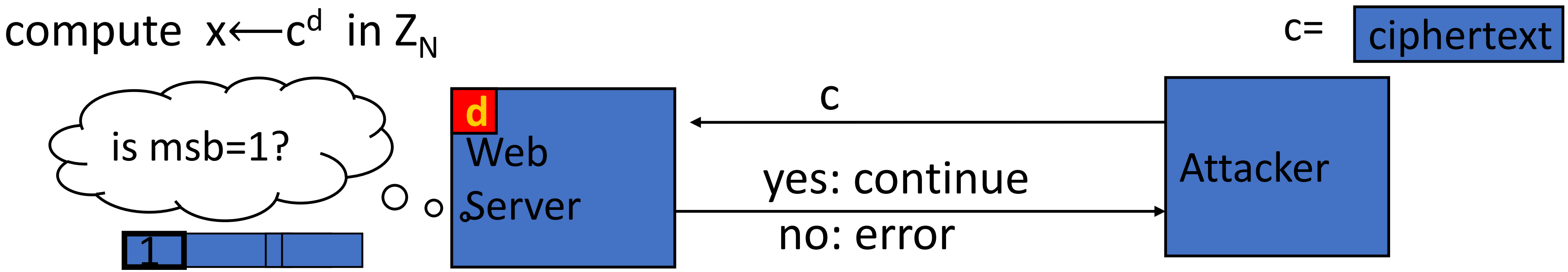
$\Rightarrow$ attacker can test if 16 MSBs of plaintext = '02'

Chosen-ciphertext attack: to decrypt a given ciphertext **c** do:

- Choose $r \in Z_N$. Compute $c' \leftarrow r^e \cdot c = (r \cdot \text{PKCS1}(m))^e$
- Send c' to web server and use response

choose r carefully.

Baby Bleichenbacher



Suppose N is $N = 2^n$ (an invalid RSA modulus). Then:

- Sending c reveals msb(x)
- Sending $\mathbf{2^e \cdot c = (2x)^e}$ in Z_N reveals $msb(2x \bmod N) = \mathbf{msb_2(x)}$
- Sending $\mathbf{4^e \cdot c = (4x)^e}$ in Z_N reveals $msb(4x \bmod N) = \mathbf{msb_3(x)}$
- … and so on to reveal all of x

The factoring problem

Gauss (1805): *"The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic."*

Best known alg. (NFS): run time exp($\tilde{O}(\sqrt[3]{n})$) for n-bit integer

Current world record: **RSA-768** (232 digits)

- Work: two years on hundreds of machines
- Factoring a 1024-bit integer: about 1000 times harder

$\Rightarrow$ likely possible this decade

Summary

- Key concepts in number theory
- Hardness of discrete logarithm, factoring
- Diffie-Hellman key exchange from hardness of DDH
- Public key encryption => shared key derivation (called key exchange)