Public Key Cryptography

Sushmita Ruj

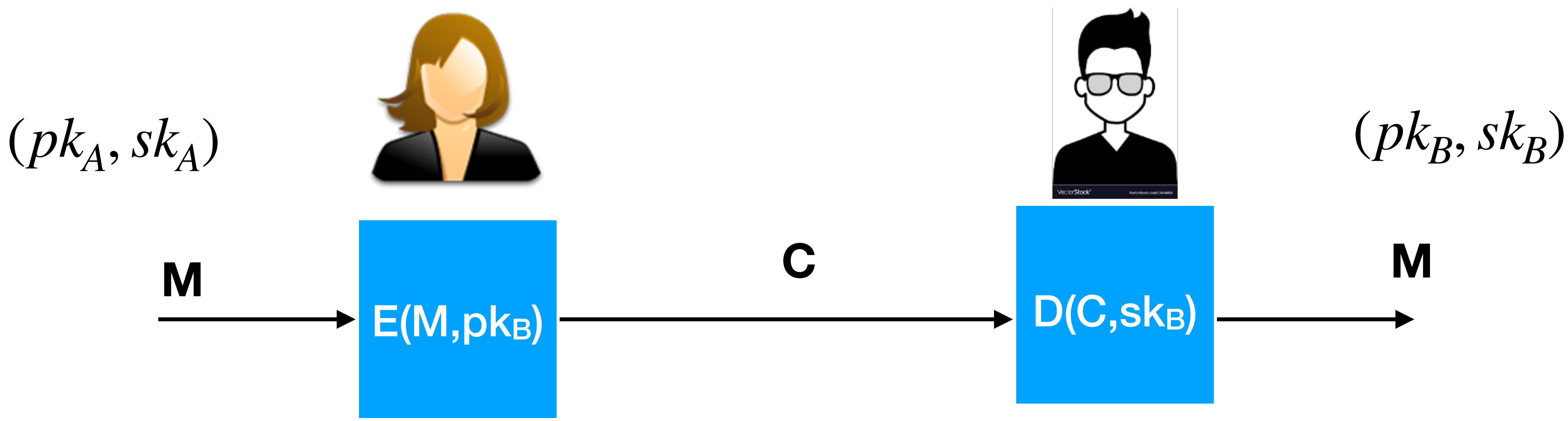
Recap

- Public Key Cryptography
- Public Key Infrastructure
- Number Theory

This Lecture

- Factoring Problem
- RSA (Textbox RSA)
- Discrete Logarithm Problem
- Diffie-Hellman Key exchange
- (Wo)Man in the middle atttack

Public Key Cryptography



Alice wants to send a message to Bob secretly

- $\varepsilon = (\mathscr{M}, \mathscr{C}, \mathscr{K})$
- $KG(1^k) \rightarrow (pk_A, sk_A), (pk_B, sk_B), \ldots$
- $For\ \ m \in \mathscr{M}, E(m, pk_B) \rightarrow c$

  Public key of Bob known to everyone
  E is a randomised algorithm
- $D(c, sk_B) \rightarrow m'$

  Secret key known only to Bob, else only Bob can decrypt
  D is a deterministic algorithm
- Correctness: $\forall k \in \mathscr{K}$ and messages $m \in \mathscr{M}$, if we execute $c \overset{R}{\leftarrow} E(m, pk_B)$, $m' \leftarrow D(c, sk_B)$, then with probability 1, $m = m'$

Tractable Problems

- Finding gcd
- Polynomial Evaluation
- Matrix multiplication
- Primality testing: Determine if a number is prime or composite (AKS Algorithm, Neeraj Kayal and Nitin Saxena were 4th Year UG students when they solved it !!!)

The factoring problem

Best known alg. (Number Field Sieve):

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

⇒ likely possible this decade

The RSA Algorithm

- Proposed by Ron Rivest, Adi Shamir and Leonard Adleman
- Published in Scientific American in 1977
- Used in SSL/TLS, Email, etc

Textbook RSA Algorithm

KeyGen: Choose random primes p,q ≈1024 bits. Set **N=pq**.

Choose integers **e , d s.t. $e \cdot d = 1 \pmod{\phi(N)}$**

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

Encrypt E(x,pk): $c=x^e \mod N$

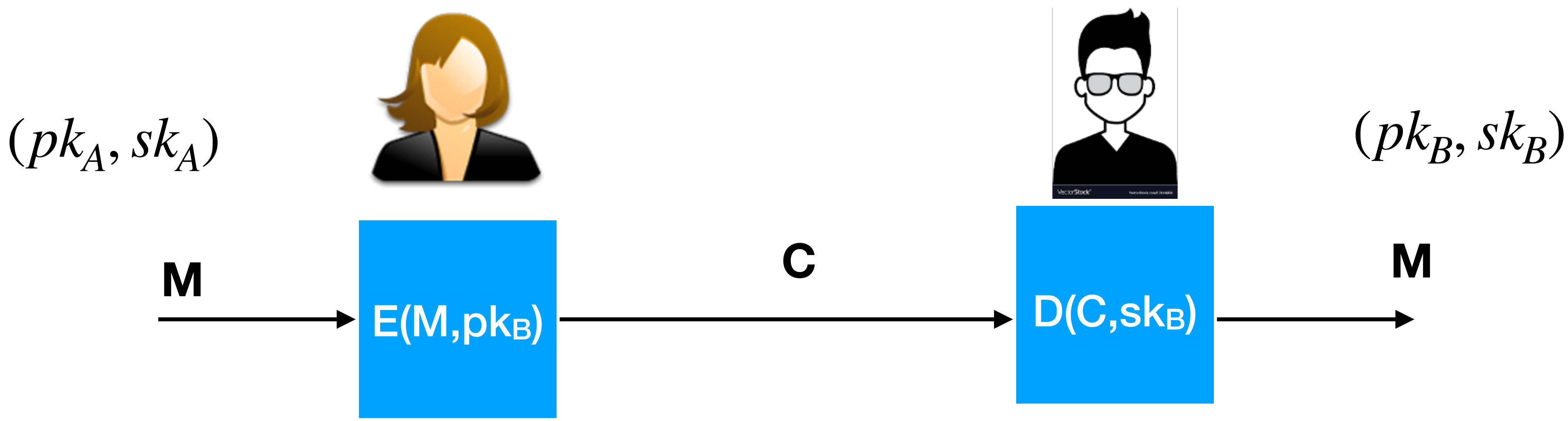
Decrypt D(x,sk,N): $c^d \mod N$

Correctness: output of decryption is $x^{ed} = x^{k\varphi(N)+1} = (x^{\varphi(N)})^k \cdot x = x$
(by Euler's thm, gcd(x, Z^*_N), $x^{\varphi(N)} = 1$ in Z_N)

If you can factor N easily, then you can compute $\phi(N)=(p-1)(q-1)$ quickly.
With the pk e, calculate sk $d=e^{-1} \bmod(\phi(N))$.

This is not how it is used!!!

Public Key Cryptography



Alice wants to send a message to Bob secretly

- $\varepsilon = (\mathscr{M}, \mathscr{C}, \mathscr{K})$
- $KG(1^k) \rightarrow (pk_A, sk_A), (pk_B, sk_B), \ldots$
- $For\ \ m \in \mathscr{M}, E(m, pk_B) \rightarrow c$

Public key of Bob known to everyone
E is a randomised algorithm

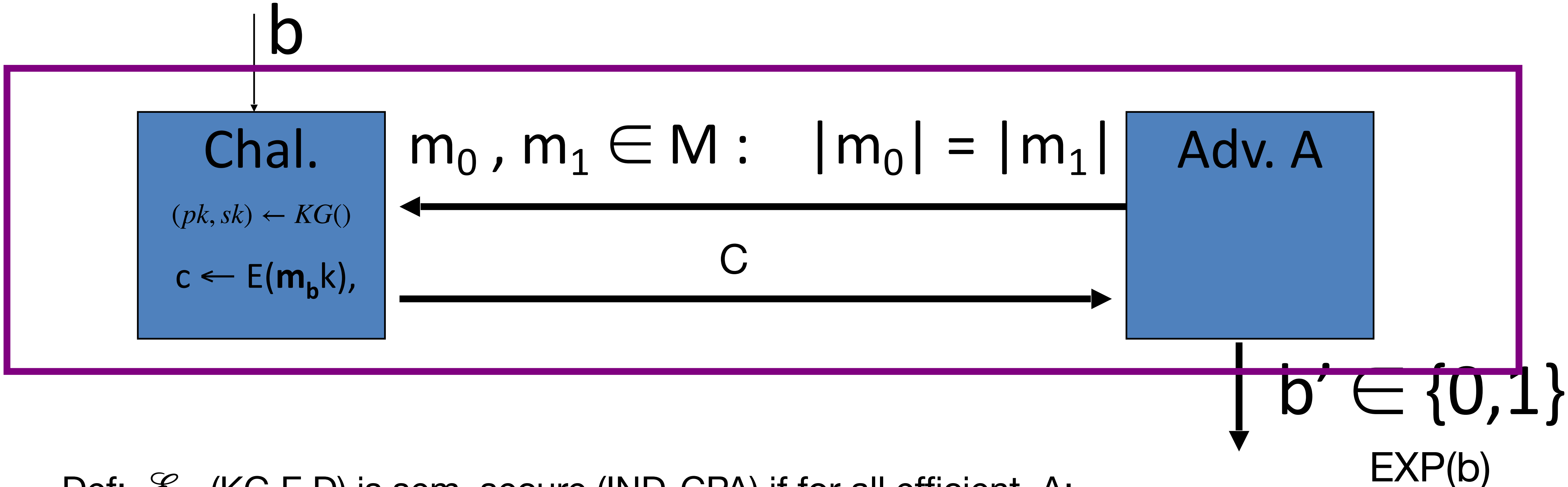
- $D(c, sk_B) \rightarrow m'$

Secret key known only to Bob, else only Bob can decrypt
D is a deterministic algorithm

- Correctness: $\forall k \in \mathscr{K}$ and messages $m \in \mathscr{M}$, if we execute $c \overset{R}{\leftarrow} E(m, pk_B)$, $m' \leftarrow D(c, sk_B)$, then with probability 1, $m = m'$

Security

For b=0,1 define experiments EXP(0) and EXP(1) as



Def: $\mathscr{E}$ =(KG,E,D) is sem. secure (IND-CPA) if for all efficient A:

$$Adv_{SS}[A,E] = |Pr[EXP(0)=1] - Pr[EXP(1)=1]| < \text{negligible}$$

The cipher is Semantically secure if for all efficient adversaries, A, $Adv_{SS}[A,\mathscr{E}]$ is neg

SKE Vs PKE

Recall: for symmetric ciphers we had two security notions:

- One-time security and many-time security (CPA)
- We showed that one-time security $\not\Rightarrow$ many-time security

For public key encryption:

- One-time security $\Rightarrow$ many-time security (CPA)

  (attacker knows PK, so can encrypt by itself)

- Public key encryption **must** be randomized

SKE Vs PKE

Secure symmetric cipher provides **authenticated encryption**

[chosen plaintext security + ciphertext integrity **]**

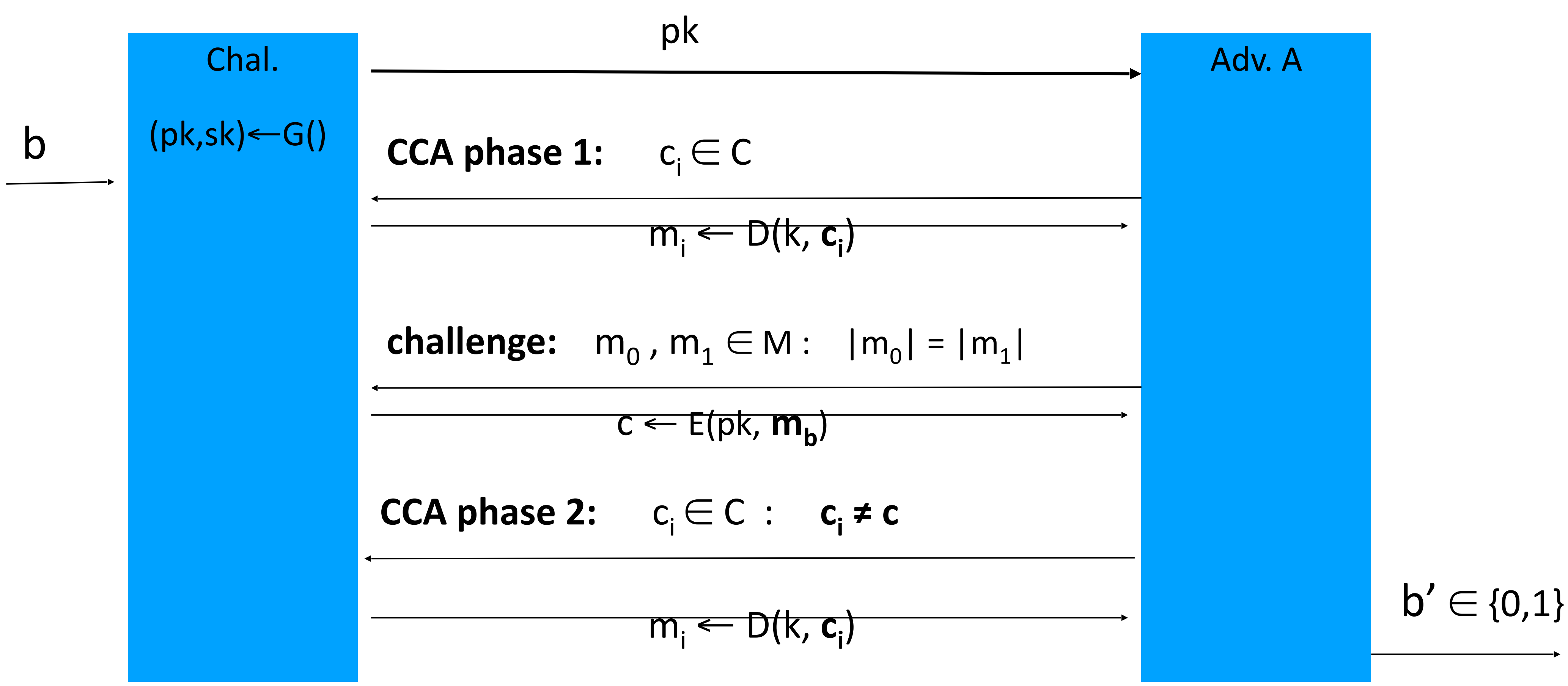
- Roughly speaking: **attacker cannot create new ciphertexts**
- Implies security against chosen ciphertext attacks

In public-key settings:

- Attacker **can** create new ciphertexts using pk !!
- So instead: we directly require chosen ciphertext security

(pub-key) Chosen Ciphertext Security: definition

$\mathscr{E}$ = (G,E,D) public-key enc. over (M,C). For b=0,1 define EXP(b):



Def: $\mathscr{E}$ is CCA secure (a.k.a IND-CCA) if for all efficient A:

$\text{Adv}_{\text{CCA}}[A,\mathscr{E}] = \left|\Pr[\text{EXP}(0)=1] - \Pr[\text{EXP}(1)=1]\right|$ is negligible.

Trapdoor Functions

Definition: A trapdoor func. $X \to Y$ is a triple of efficient algs. (KG, F, F^{-1})

- KG(): randomized algorithm outputs a key pair (pk, sk)
- F(pk, ⋅): deterministic algorithm that defines a function $X \to Y$
- F^{-1}(sk, ⋅): defines a function $Y \to X$ that inverts F(pk, ⋅)

More precisely: ∀(pk, sk) output by KG

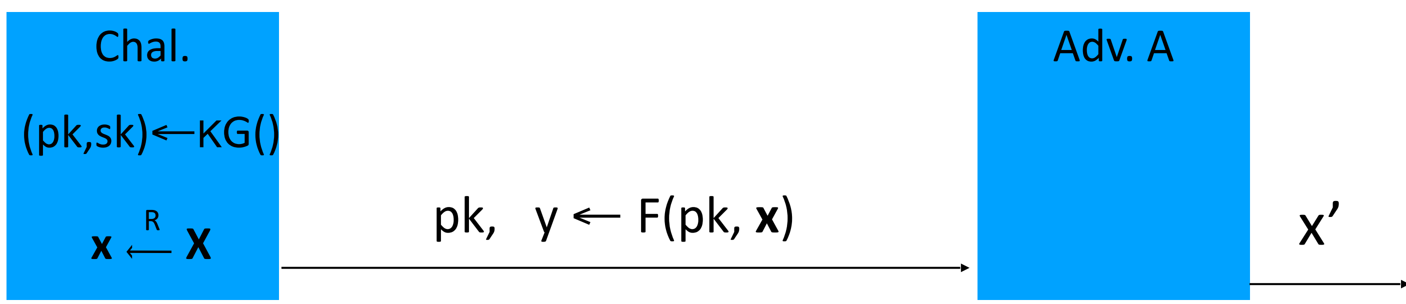
$$\forall x \in X: \quad F^{-1}(sk, F(pk, x)) = x$$

Definition: A trapdoor permutation $X \to X$ is a triple of efficient algorithms (KG, F, F^{-1})

Security of Trapdoor Function

(KG, F, F^{-1}) is secure if F(pk, ⋅) is a "one-way" function:

Easy to compute, but hard to invert without sk



Definition: (KG, F, F^{-1}) is a secure TDF if for all efficient A:

$$\mathrm{Adv}_{\mathrm{OW}}[A,F] = \Pr[\, x = x' \,] < \text{negligible}$$

PKE from TDF

- (KG, F, F^{-1}): secure TDF $X \longrightarrow Y$
- (E_s, D_s) : symmetric auth. encryption defined over (K,M,C)
- $H: X \longrightarrow K$ a hash function

E(pk, m) :

$x \longleftarrow X$, $y \longleftarrow F(pk, x)$

$k \longleftarrow H(x)$, $c \longleftarrow E_s(k, m)$

output (y, c)

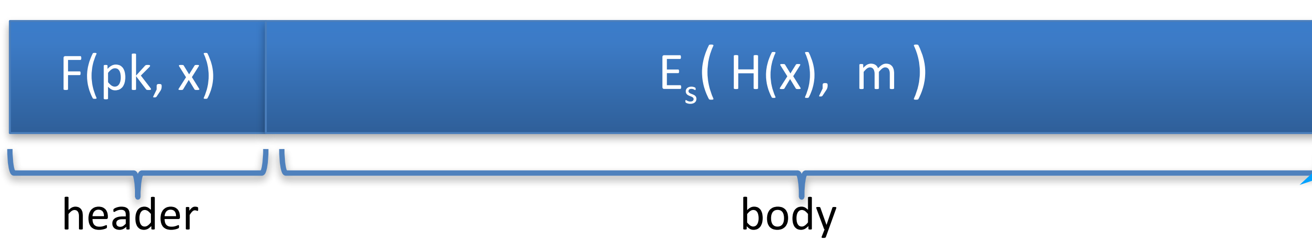
D(sk, (y,c)) :

$x \longleftarrow F^{-1}(sk, y)$,

$k \longleftarrow H(x)$, $m \longleftarrow D_s(k, c)$

output m

Use PKE to generate common key k, them encrypt message using SKE and common key k



Security and Use

Security Theorem:

If **(KG, F, F^{-1})** is a secure TDF, **(E_s, D_s)** provides auth. enc.

and **H:** $X \longrightarrow K$ is a "random oracle"

then **(KG,E,D)** is CCA^{ro} secure.

Never encrypt by applying F directly to plaintext:

E(pk, m) :	**D(sk, c) :**
output $c \longleftarrow F(pk, m)$	output $F^{-1}(sk, c)$

Problems:

- Deterministic: cannot be semantically secure !!

The RSA trapdoor permutation

KG: Choose random primes p,q ≈1024 bits. Set **N=pq**.

Choose integers **e , d s.t. $e \cdot d = 1 \pmod{\phi(N)}$**

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

F(x,pk): $c = x^e \mod N$

F^{-1}(x,sk,N):$c^d \mod N$

Correctness: output of decryption is $x^{ed} = x^{k\varphi(N)+1} = (x^{\varphi(N)})^k \cdot x = x$
(by Euler's thm, gcd(x, Z^*_N), $x^{\varphi(N)} = 1$ in Z_N)

HW: Try out public key encryption and key management using OpenSSL

RSA Assumption

RSA assumption: RSA is one-way permutation

For all efficient algs. A:

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

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

RSA

(E_s, D_s): SKE scheme providing authenticated encryption.

H: $Z_N \rightarrow K$ where K is key space of (E_s, D_s)

- **KG**(): generate RSA params: pk = (N,e), sk = (N,d).
- **E**(pk,m): (1) choose random x in Z_N

  (2) $y \leftarrow RSA(x) = x^e$, $k \leftarrow H(x)$

  (3) output $(y , E_s(m,k))$

- **D**(sk,(c,y)): output $D_s(H(RSA^{-1}(y)),c)$

Testbook RSA is Insecure

Textbook RSA encryption:

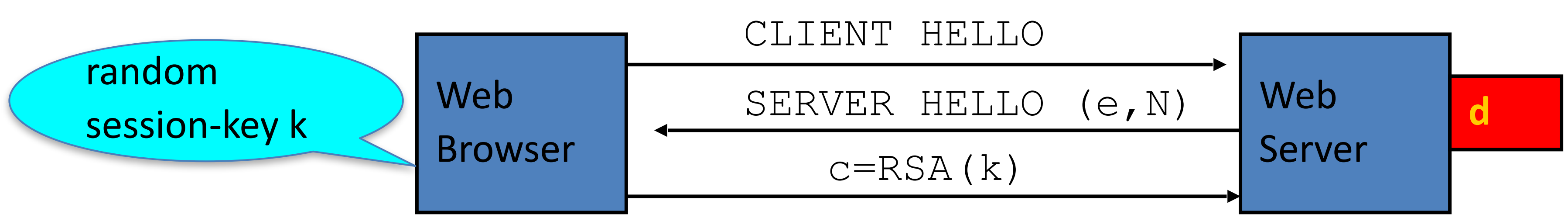
- public key: **(N,e)** Encrypt: $\mathbf{c} \longleftarrow \mathbf{m^e}$ (in Z_N)
- secret key: **(N,d)** Decrypt: $\mathbf{c^d} \longrightarrow \mathbf{m}$

Insecure cryptosystem !!

- Is not semantically secure and many attacks exist

$\Rightarrow$ The RSA trapdoor permutation is not an encryption scheme !

A simple attack on Textbook RSA



Suppose k is 64 bits: $k \in \{0,\ldots,2^{64}\}$. Eve sees: $c = k^e$ in Z_N

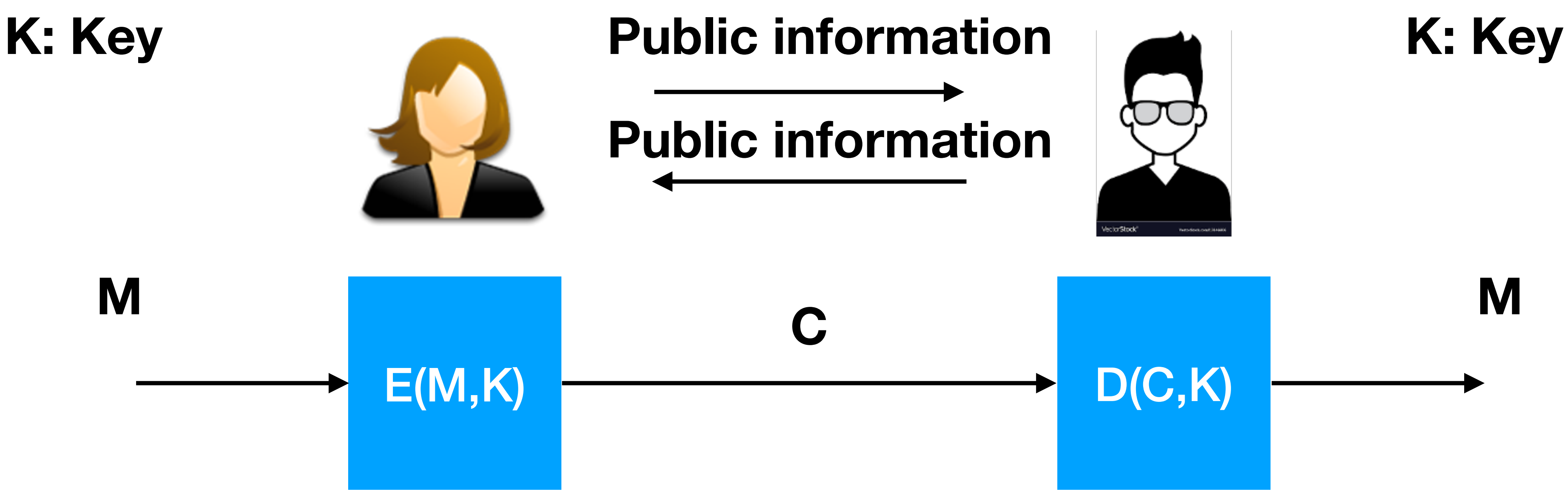
If $\mathbf{k = k_1 \cdot k_2}$ where $k_1, k_2 < 2^{34}$ (prob. ≈20%) then $\mathbf{c/k_1^e = k_2^e}$ in Z_N

Step 1: build table: $c/1^e, c/2^e, c/3^e, \ldots, c/2^{34e}$. time: 2^{34}

Step 2: for $k_2 = 0,\ldots, 2^{34}$ test if k_2^e is in table. time: 2^{34}

Output matching (k_1, k_2). Total attack time: $\approx 2^{40} << 2^{64}$

Key Agreement



- Establish shared key between Alice and Bob
- **Without** assuming an existing shared ('master') key or trusted authority
- Use public information from A, B to setup shared secret key k.
- Eavesdropper cannot learn the key k.

Whitfield Diffie & Martin Hellman 1976 "New Directions of Cryptography"
Awarded Turing Award in 2015

Diffie-Hellman Key Agreement

Choose group G of order p and generator of $g \in G$

$$a \leftarrow Z_p^* \qquad \xrightarrow{\quad A \quad} \qquad b \leftarrow Z_p^*$$

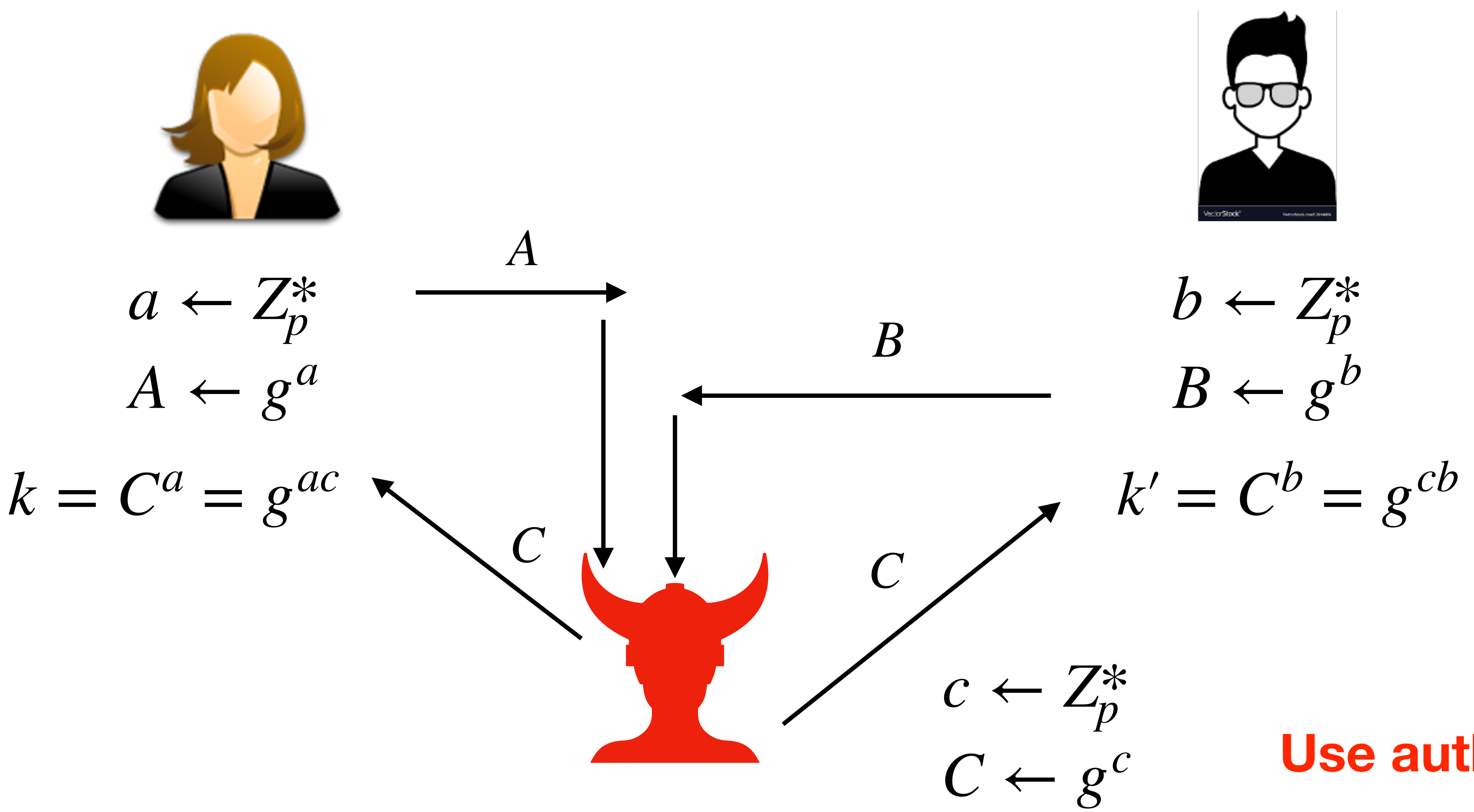
$$A \leftarrow g^a \qquad \xleftarrow{\quad B \quad} \qquad B \leftarrow g^b$$

$$k = B^a = g^{ab} \qquad\qquad k = A^a = g^{ba}$$

K is the common key

(Wo)man in the middle attack

Choose group G of order p and generator of $g \in G$



Use authentication

Alice encrypts message with k which can be decrypted by Charlie
Bob encrypts message with k’ which can be decrypted by Charlie

Thank you