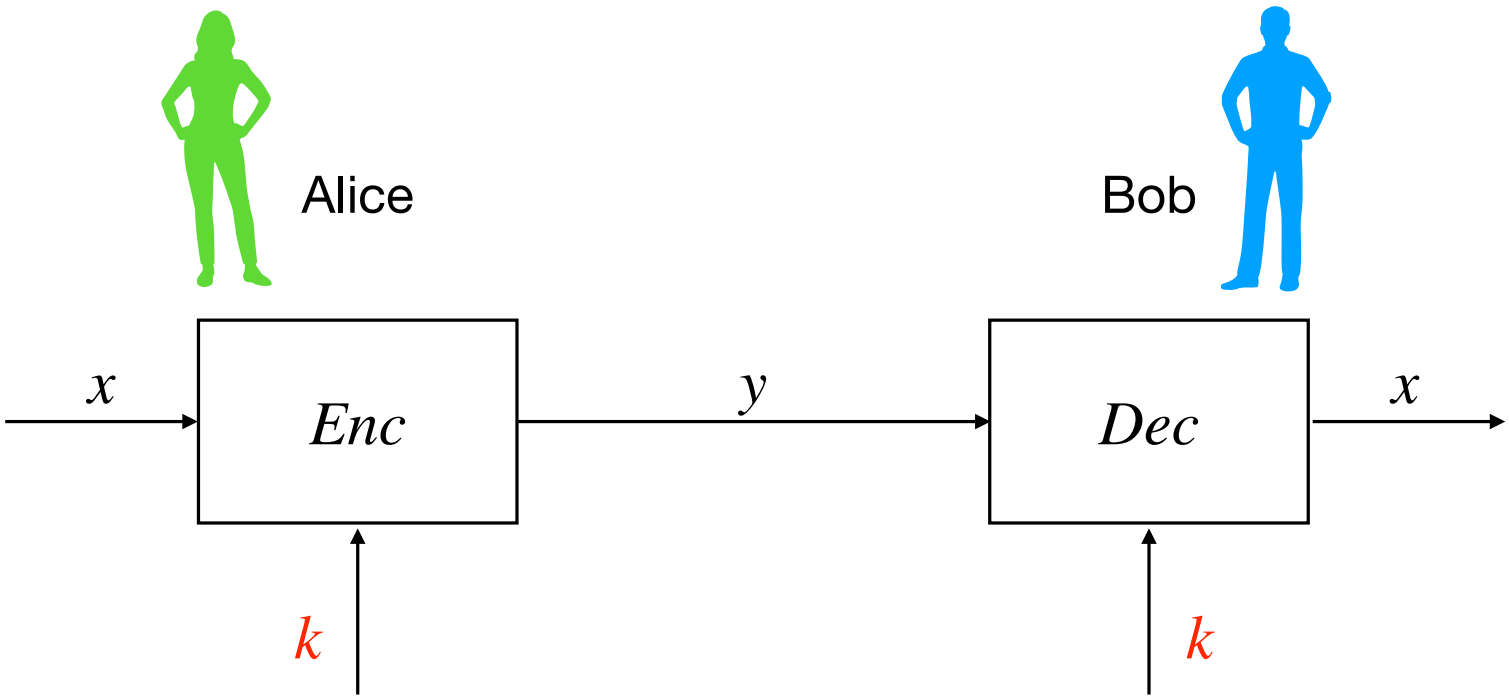
HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Introduction to Cryptography and Security

Introduction to Public-key Cryptography

Outline

Symmetric Cryptography



- The same secret key k is used for encryption and decryption.
- The encryption and decryption function are very similar (in the case of DES they are essentially identical).

Symmetric Cryptography Revisited

- Modern symmetric algorithms such as AES or 3DES are very secure, fast and are in widespread use.
- But the key must be established between Alice and Bob using a secure channel.

Key management

- n users.
- Storing mutual secret key is difficult.

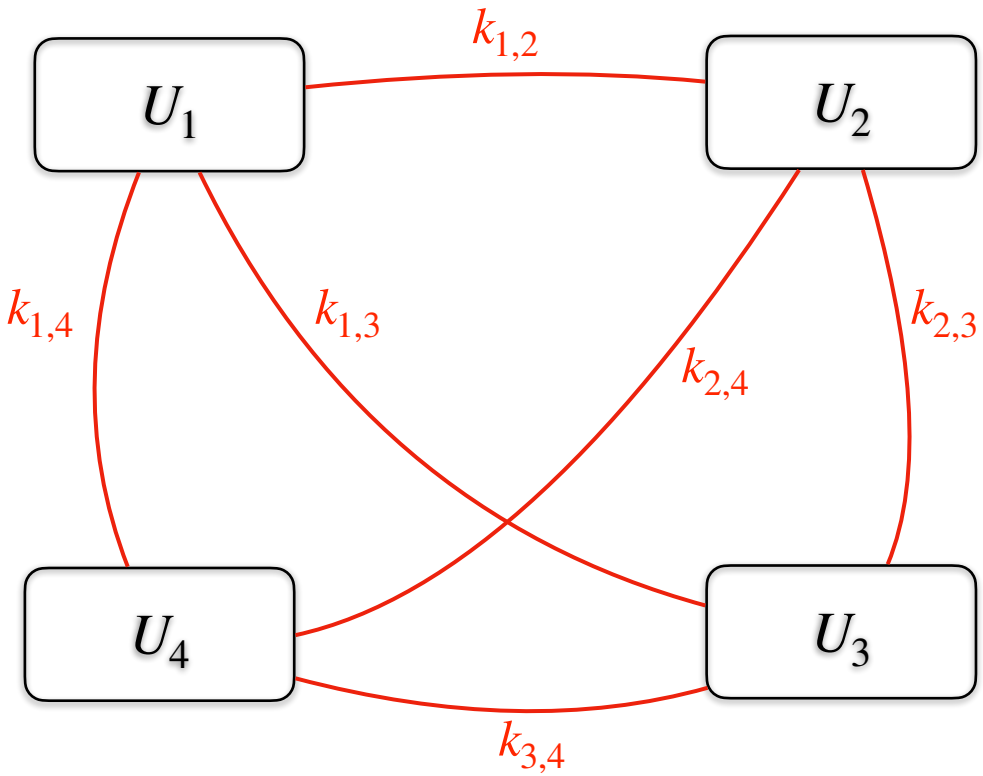


Figure: $O(n)$ keys per user

No Protection Against Cheating

by Alice or Bob

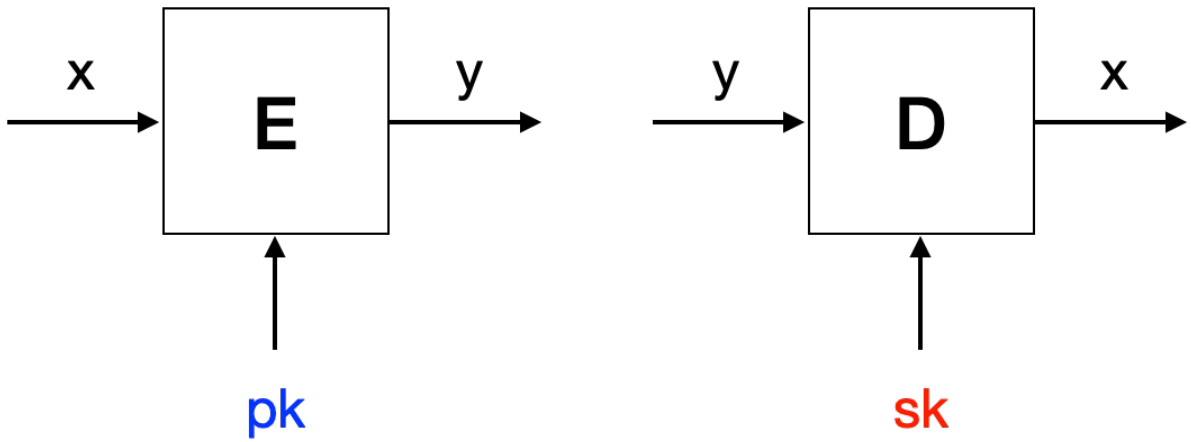
- Alice and Bob have the same capabilities, since they possess the same key.
- For example, Alice can claim that she never ordered online TV from Bob (he could make the order himself).
- Preventing this is called nonrepudiation and can be achieved with asymmetric cryptography.

Outline

Principles of Asymmetric Cryptography

Whitfield Diffie, Martin Hellman, and Ralph Merkle – 1976

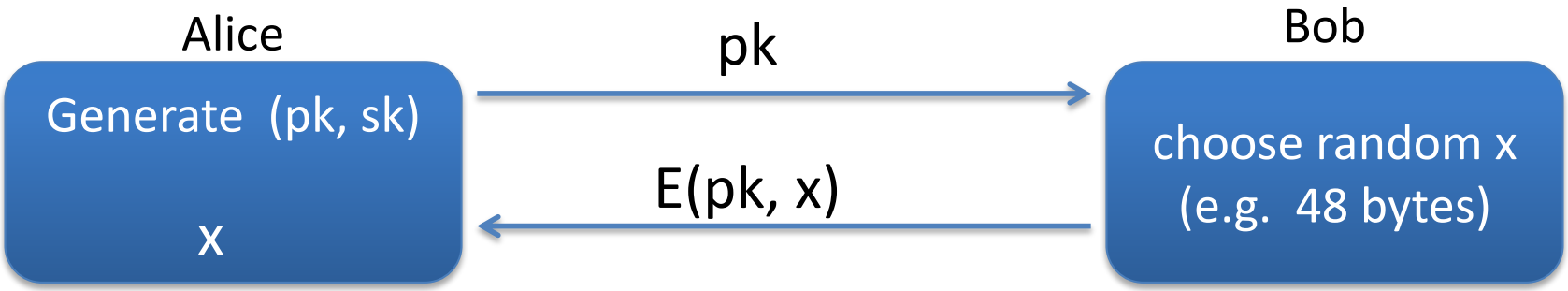
Public key encryption



Bob generates $k = (\text{pk}, \text{sk})$ and gives pk to Alice.

Application: Session setup

for now, only eavesdropping security



Non-interactive applications: : (e.g. Email)

- Bob sends email to Alice encrypted using pk_{alice}
- Note: Bob needs pk_{alice} (public key management)

Public key encryption

Definition

A ***Public key encryption system*** is a triple of algorithms

$$(\mathsf{G}, \mathsf{E}, \mathsf{D})$$

where:

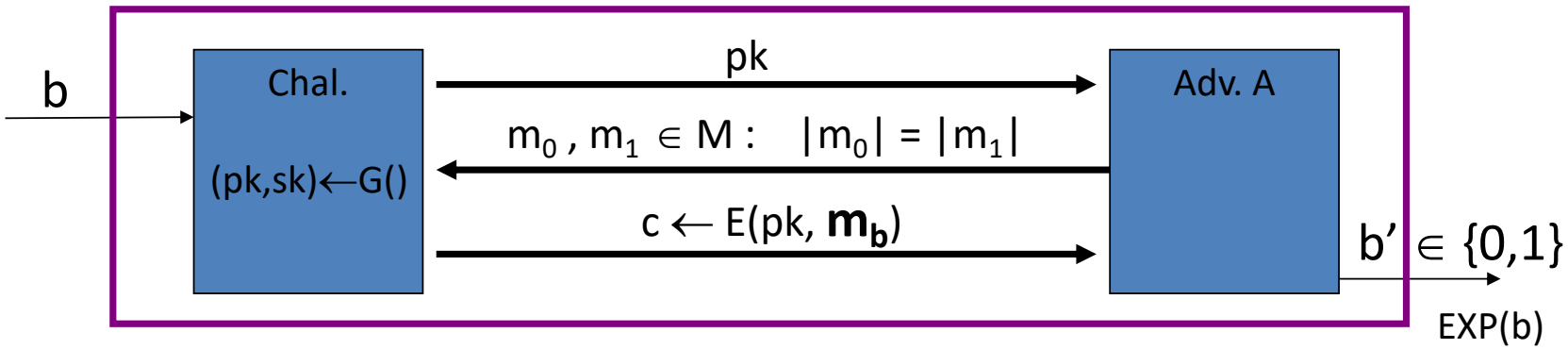
- $\mathsf{G}()$: randomized algorithm output a key pair $(\mathsf{pk}, \mathsf{sk})$
- $\mathsf{E}(pk, m)$: randomized algorithm that takes $m \in M$ and output $c \in C$
- $\mathsf{D}(sk, c)$: deterministic algorithm that takes $c \in C$ and output $m \in M$ or $\perp$

Consistency: For all $(\mathsf{pk}, \mathsf{sk})$ output by G:

$$\forall m \in M: \quad \mathsf{D}(\mathsf{sk}, \ \mathsf{E}(\mathsf{pk}, \ m)) = m.$$

Semantic Security

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



Definition

The public key system $(\mathrm{G}, \mathrm{E}, \mathrm{D})$ is ***semantic secure*** if for all efficient attacker A:

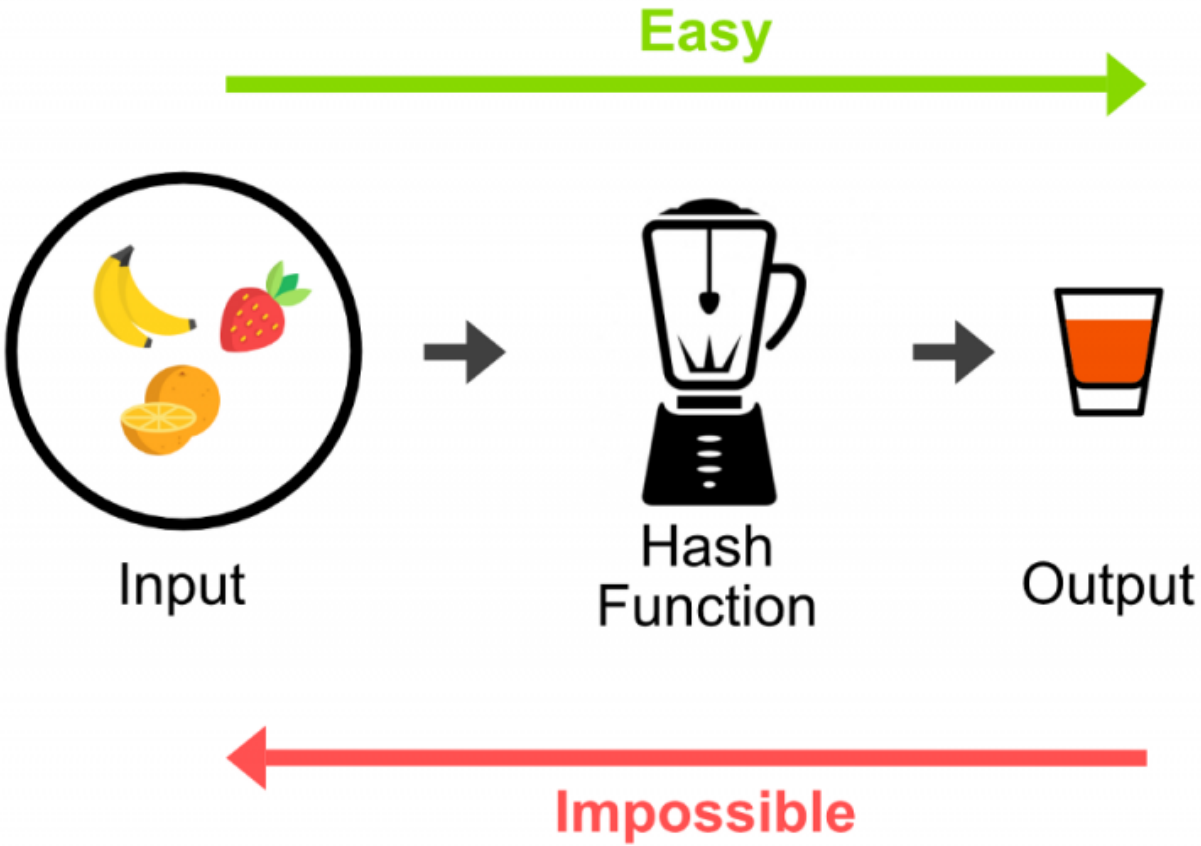
$$|\Pr[EXP(0) = 1] - \Pr[EXP(1) = 1]| < \text{ negligible}$$

One-way function

The public-key algorithms are all built from one common principle, the ***one-way function***.

1. $y = f(x)$ is computationally easy,
2. $x = f^{-1}(y)$ is computationally infeasible.

One-way function



Source: https://computersciencewiki.org

Public-Key Algorithm Families

- ***Integer Factorization Scheme***: The most prominent representative of this algorithm family is RSA.
- ***Discrete Logarithm Scheme***: For example, the Diffie–Hellman key exchange, Elgamal encryption or the Digital Signature Algorithm (DSA).
- ***Elliptic Curve Schemes***: For example, Elliptic Curve Diffie–Hellman key exchange (ECDH) and the Elliptic Curve Digital Signature Algorithm (ECDSA).

Comparison

Symmetric key size (bits)	Elliptic Curve	RSA or Diffie-Hellman
80	160	1024
112	224	2048
128	256	3072
192	384	7680
256	521	15360

Outline

Trapdoor functions (TDF)

Definition

A trapdoor function $X \to Y$ is a triple of efficient algorithms (G, F, F^{-1})

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

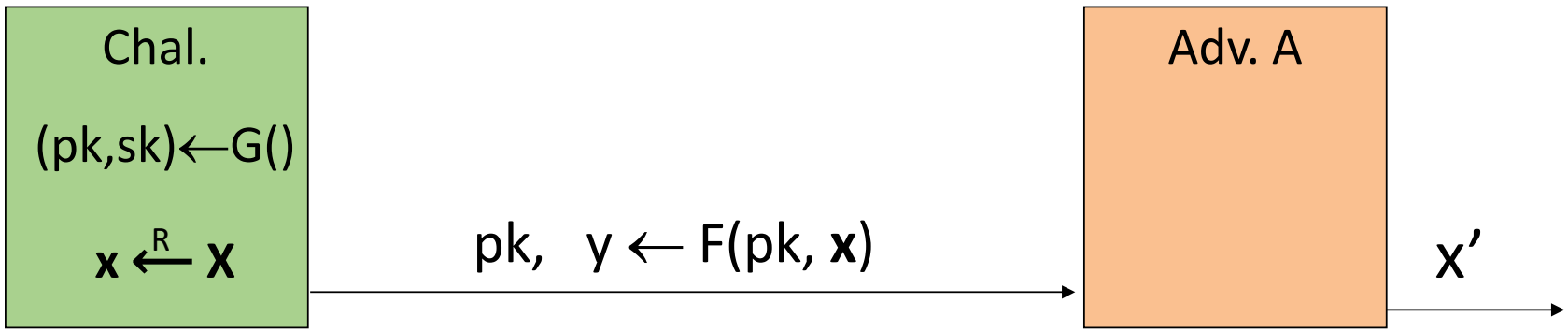
More precisely: $\forall(\mathsf{pk}, \mathsf{sk})$ output by $\mathsf{G}()$, we have

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

Secure Trapdoor Functions (TDFs)

(G, F, F^{-1}) is ***secure*** if $F(pk, \cdot)$ is a "one-way" function:

can be evaluated, but cannot be inverted without sk.



Definition

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

$$\Pr[x = x'] < \text{ negligible}$$

Public-key encryption from TDFs

- (G, F, F^{-1}): secure TDF $X \to Y$;
- $(\mathsf{E}_s, \mathsf{D}_s)$: symmetric encryption defined over (K, M, C);
- $\mathsf{H} : X \to Y$: a hash function.

We construct a pub-key encryption system

$$(\mathsf{G}, \mathsf{E}, \mathsf{D})$$

where

- key generation G: same as G for TDF.

Public-key encryption from TDFs

- (G, F, F^{-1}): secure TDF $X \to Y$;
- $(\mathsf{E}_s, \mathsf{D}_s)$: symmetric encryption defined over (K, M, C);
- $\mathsf{H} : X \to Y$: a hash function.

Encryption & Decryption

$\mathsf{E}(\mathsf{pk}, m)$:

$x \leftarrow\!\!\$\, X, \qquad y = F(\mathsf{pk}, x)$

$k = \mathsf{H}(x), \quad c \leftarrow \mathsf{E}_s(k, m)$

return (y, c)

$\mathsf{D}(\mathsf{sk}, (y, c))$:

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

$k = \mathsf{H}(x), \quad m = \mathsf{D}_s(k, c)$

return m

Incorrect use of a Trapdoor Function

Never encrypt by applying F directly to plaintext:

Encryption & Decryption

$\mathsf{E}(\mathsf{pk}, m)$:	$\mathsf{D}(\mathsf{sk}, c)$:
return $c = F(\mathsf{pk}, m)$	return $m = F^{-1}(\mathsf{sk}, c)$

Problems:

- Deterministic: cannot be semantically secure!
- Many attacks exist

Outline

Review: trapdoor permutations

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

- G: outputs pk, sk. pk defines a function $F(pk, \cdot) : X \to X$
- $F(\text{pk}, x)$: evaluates the function at x
- $F^{-1}(\text{sk}, y)$: inverts the function at y using sk

Secure trapdoor permutation:

- The function $F(pk, \cdot)$ is one-way without the trapdoor sk

Review: arithmetic mod composites

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

$$\mathbb{Z}_N = \{0, \ldots, N-1\} \quad ; \quad \mathbb{Z}_N^* = \{ \text{ invertible elements in } \mathbb{Z}_N \}$$

Fact

- $x \in \mathbb{Z}_N$ *is invertible* *iff* $\gcd(x, N) = 1$
- *Number of elements in* $\mathbb{Z}_N^*$ *is*

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

Theorem (Euler)

$$\forall x \in \mathbb{Z}_N^* : \quad x^{\varphi(N)} = 1$$

The RSA trapdoor permutation

First published: Scientific American, Aug. 1977.

Very widely used:

- SSL/TLS: certificates and key-exchange
- Secure e-mail and file systems

The RSA trapdoor permutation

Randomized algorithm G()

- choose random primes $p, q \approx 1024$ bits. Set $N = p \cdot q$.
- choose integers e, d such that $e \cdot d = 1 \mod \varphi(N)$
- output $\mathsf{pk} = (N, e)$, $\mathsf{sk} = (N, d)$

Algorithm $F(\mathsf{pk}, x) : \mathbb{Z}_N^* \to \mathbb{Z}_N^*$

$$\mathsf{RSA}(x) = x^e \qquad (\text{in } \mathbb{Z}_N)$$

Algorithm $F^{-1}(\mathsf{sk}, y) = y^d$

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

The RSA assumption

RSA is one-way permutation

RSA assumption

For all efficient algorithms A:

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

where $p, q \leftarrow\!\!\$$ n-bit primes, $N = pq$, and $y \leftarrow\!\!\$\ \mathbb{Z}_N^*$

RSA pub-key encryption (ISO std)

$(\mathsf{E}_s, \mathsf{D}_s)$: symmetric encryption scheme providing auth. encryption.

$H : \mathbb{Z}_N \to K$ where K is key space of $(\mathsf{E}_s, \mathsf{D}_s)$.

- $\mathsf{G}()$: generate RSA params: $\mathsf{pk} = (N, e)$, $\mathsf{sk} = (N, d)$
- $E(\mathsf{pk}, m)$:
  - choose random x in $\mathbb{Z}_N$
  - $y = \mathrm{RSA}(x) = x^e, \quad k = H(x)$
  - output $(\ y,\ \mathsf{E}_s(k, m)\ )$
- $\mathsf{D}(\mathsf{sk}, (y, c))$: output $\mathsf{D}_s(\ H(\mathrm{RSA}^{-1}(y)),\ c)$.

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

soict.hust.edu.vn/ fb.com/groups/soict