CS 4770: Cryptography

CS 6750: Cryptography and Communication Security

Alina Oprea
Associate Professor, CCIS
Northeastern University

April 12 2018

Historical cryptography

Cryptography ≈ Encryption

Main applications: **military and diplomacy**

ancient times **world war II**

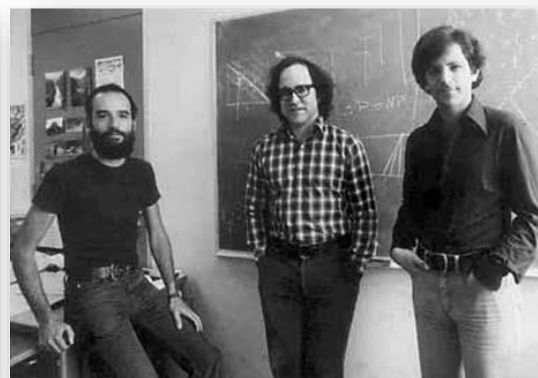
Modern cryptography

Cryptography based on rigorous science/math

multiparty-computations

zero-knowledge

threshold crypto



electronic auctions

electronic voting

crypto currencies

private info retrieval

public-key cryptography

information theory

signature schemes

rigorous definitions

computation in cloud

...

post-war sevenites now

Course objectives

- Introduction to basic cryptographic primitives
  - Secret-key cryptography
  - Public-key cryptography
  - Threat models
- Modern cryptographic protocol design
  - Sound, rigorous proofs of security
  - Understand fundamental assumptions
- Applications
  - Secure network communication, TLS, crypto currencies

http://www.ccs.neu.edu/home/alina/classes/Spring2018

What we covered

Key distribution / PKI	TLS / HTTPS	Crypto currencies

Public-key cryptography
- Key exchange
- Trapdoor functions and permutations
- Secure encryption (CPA, CCA)
- Digital signatures

Collision-Resistant Hash Functions

Symmetric-key cryptography
- Pseudorandom generators
- Pseudorandom functions and permutations
- Secure encryption (EAV, CPA, CCA)
- Message Authentication Codes (MACs)

- Definitions of security
- Relationships between primitives
- Secure and insecure constructions
- Security proofs by reduction
- Standards for cryptographic primitives

Probability and statistics	Number theory

Takeaway 1: Kerckhoffs' principle

Auguste Kerckhoffs (1883):

The enemy knows the system

The cipher should remain secure even if **the adversary knows the specification of the cipher.**

The only thing that is **secret** is a

key **k**

that is **usually chosen uniformly at random**

Takeaway 2: Computational Security

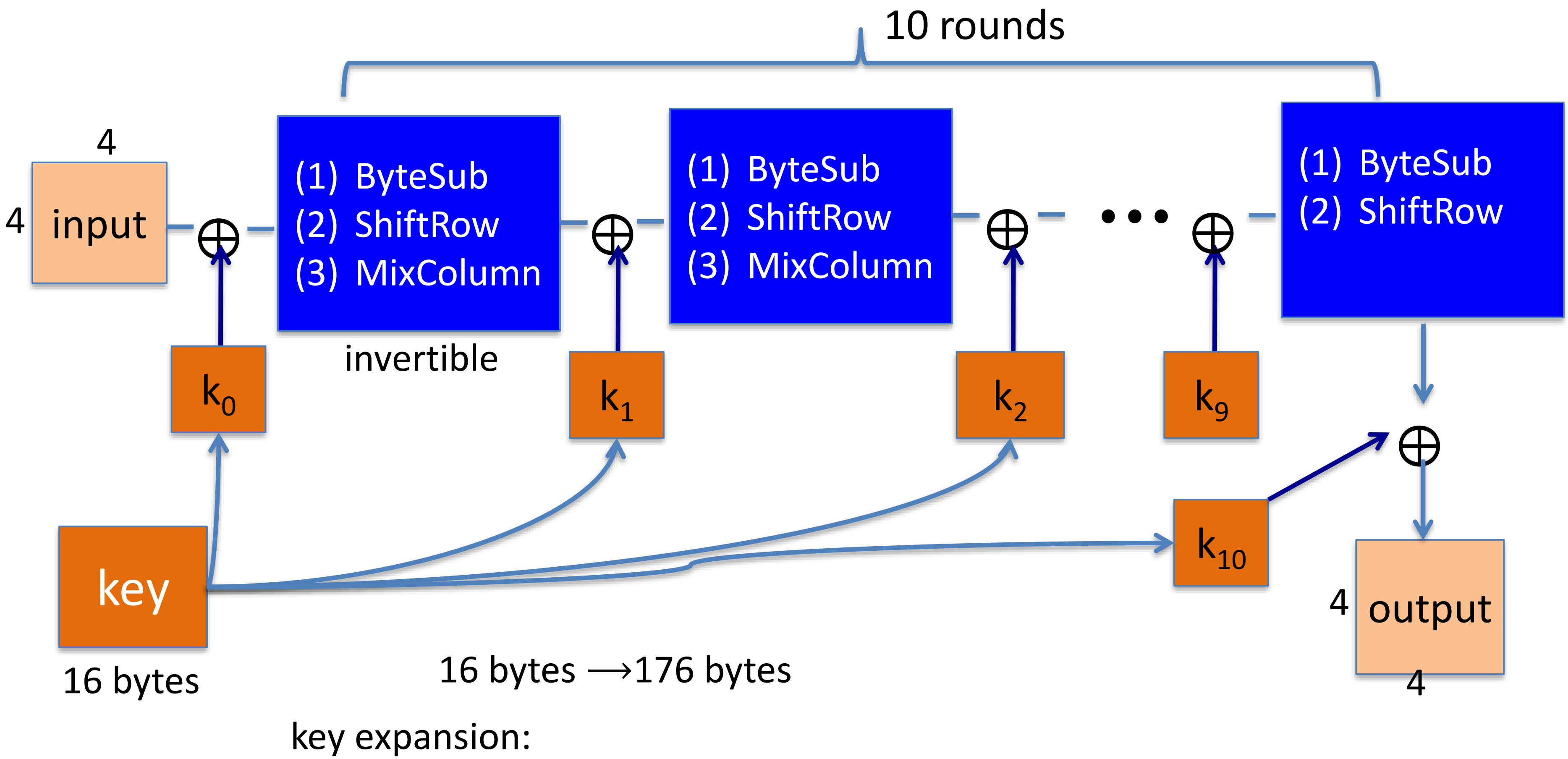
Typically, we will say that a **scheme C is secure** if

$\forall$ Pr[A(n) "breaks the scheme" C(n)] is **negligible** in n.

Probabilistic polynomial-time algorithm A

- Scheme **C** and the **adversary A** take input **security parameter**.
- 2 relaxations of perfect security
  - PPT adversary
  - Adversary can succeed, but with very small probability (negligible)

Takeaway 3: Standards for encryption (AES-128)



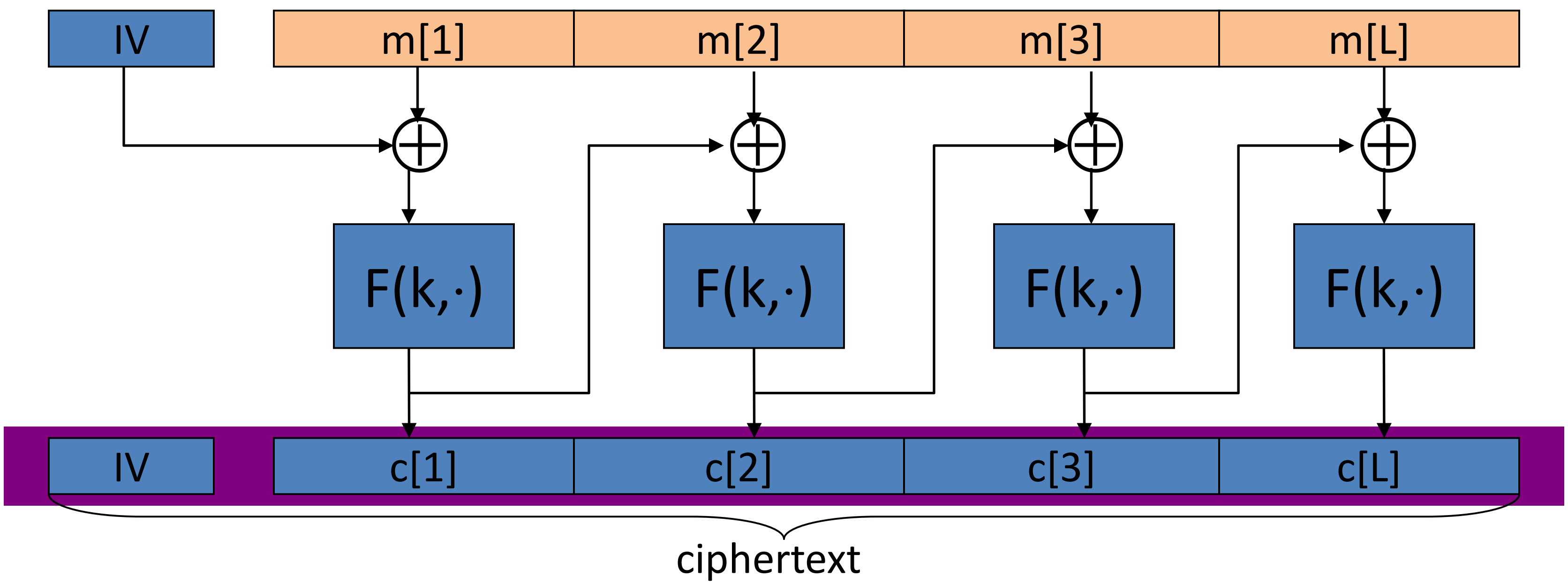
Can be used as a PRF or PRP

Building block in many constructions

Takeaway 4: Encryption modes
CBC encryption

Let F be a PRP; F: K × $\{0,1\}^n \longrightarrow \{0,1\}^n$ - use AES

$\text{Enc}_{\text{CBC}}(k,m)$: choose **random** IV∈ $\{0,1\}^n$ and do:



$$c_i = F_k(c_{i-1} \oplus m_i)$$

Takeaway 5: Relation between security notions

- CPA security implies EAV security
- CCA security implies CPA security

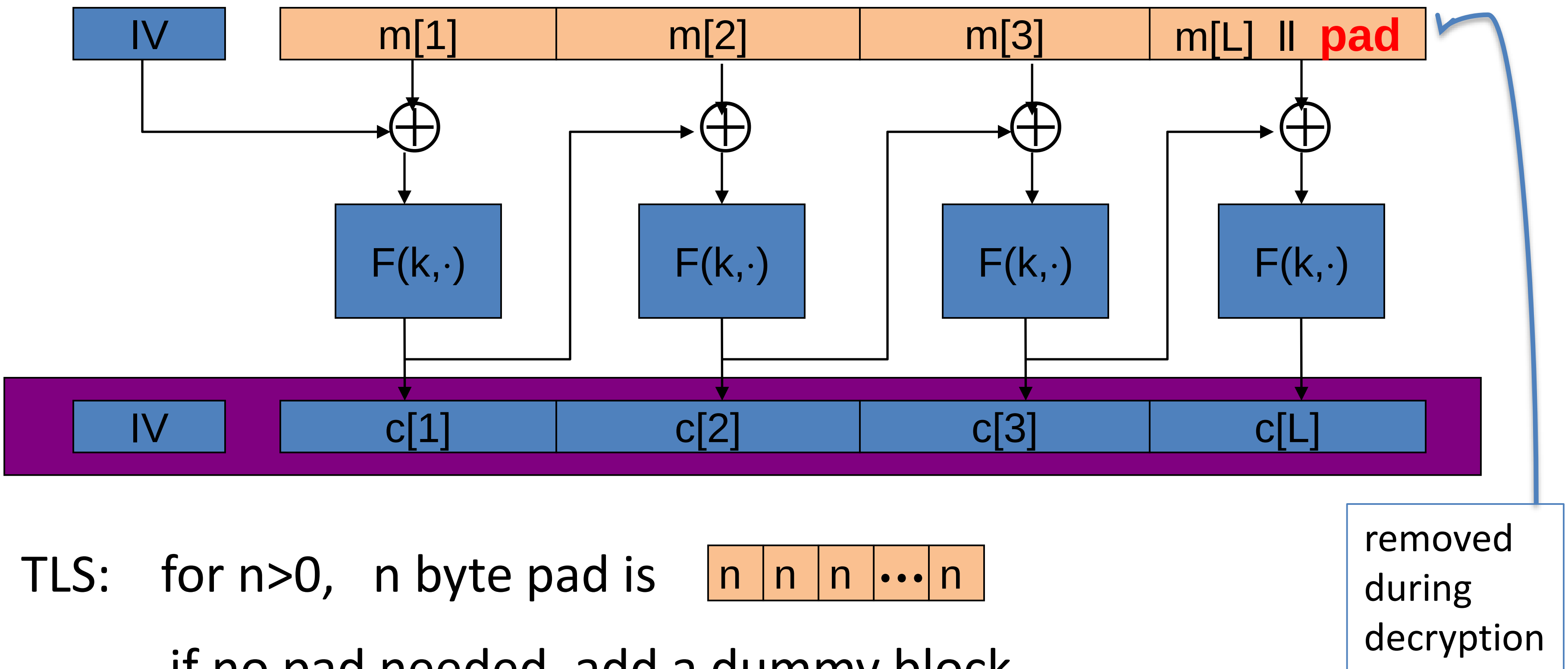
Symmetric-key world

- CPA security strictly stronger than EAV security
- CCA security strictly stronger than CPA security

Public-key world

- CPA security is equivalent to EAV security

Takeaway 6: Padding might be vulnerable



TLS: for n>0, n byte pad is | n | n | n | ••• | n |

if no pad needed, add a dummy block

Takeaway 7: Encryption does not provide integrity!

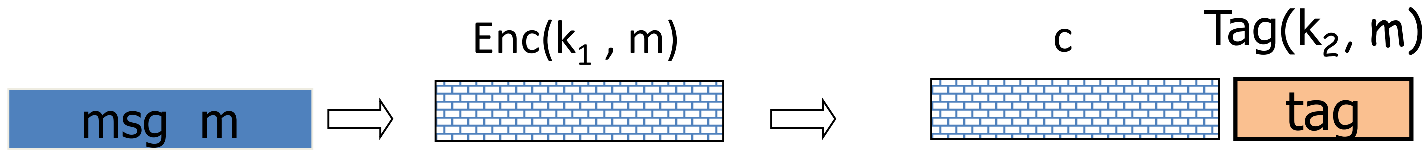
- Stream ciphers
  - Enc(k, m) = m $\oplus$ G(k), G a secure PRG
  - Modify 1 bit in c implies one bit modification in the decrypted message
- Block ciphers
  - CTR: Enc is one-time pad with output of PRF function
  - Can modify the ciphertext and decrypt to a different message

Need another primitive: MACs

Takeaway 8: Order of encryption and integrity matters!

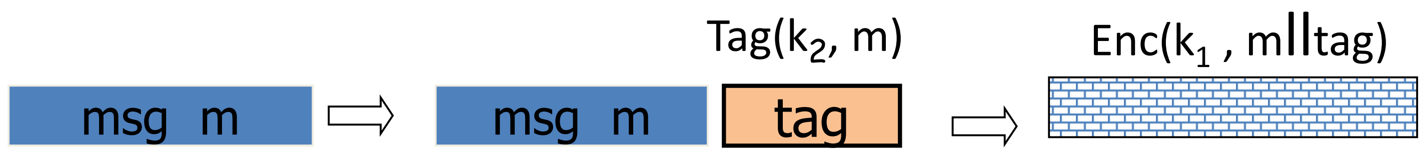
Enc-and-MAC

<u>Option 1</u>: (SSH)



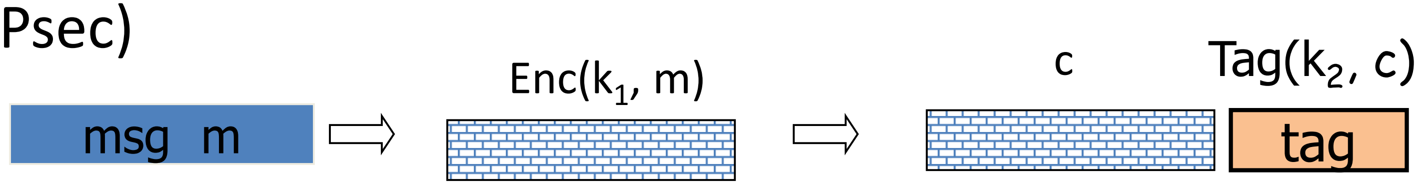
<u>Option 2</u>: (SSL)

MAC-then-enc



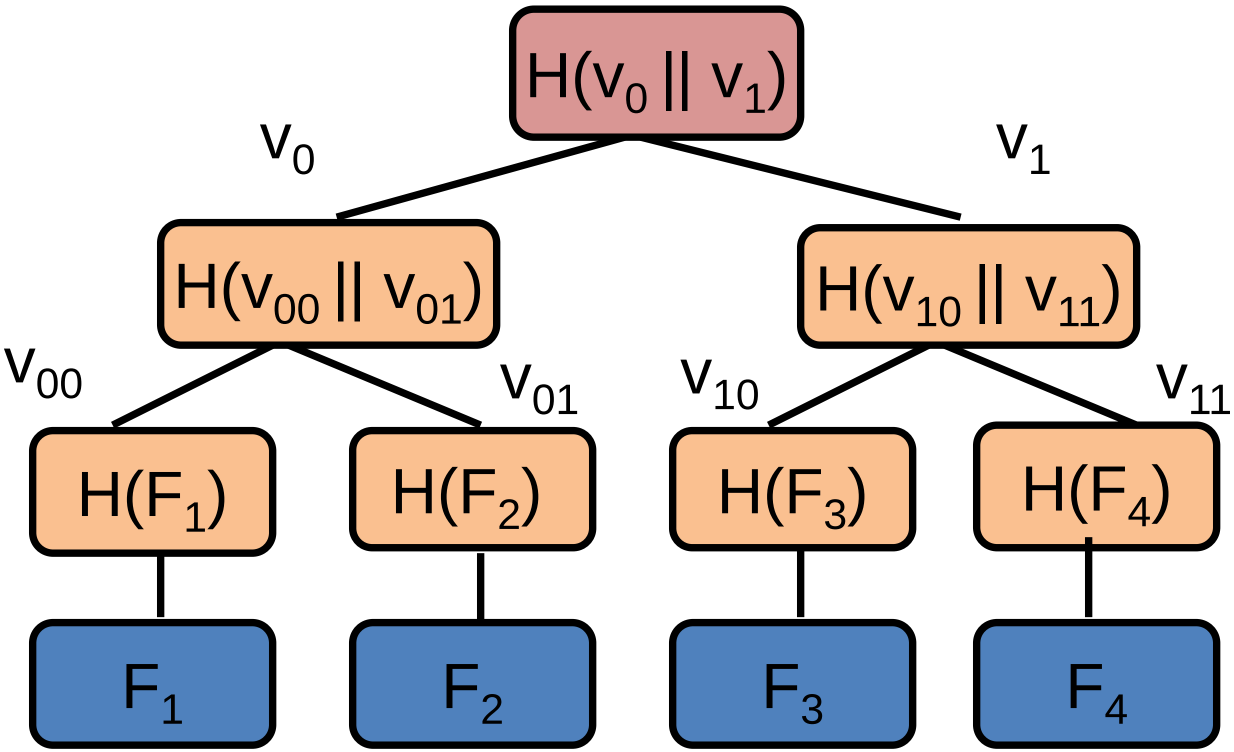
Enc-then-MAC

<u>Option 3</u>: (IPsec)



Takeaway 9: Hash functions have many applications

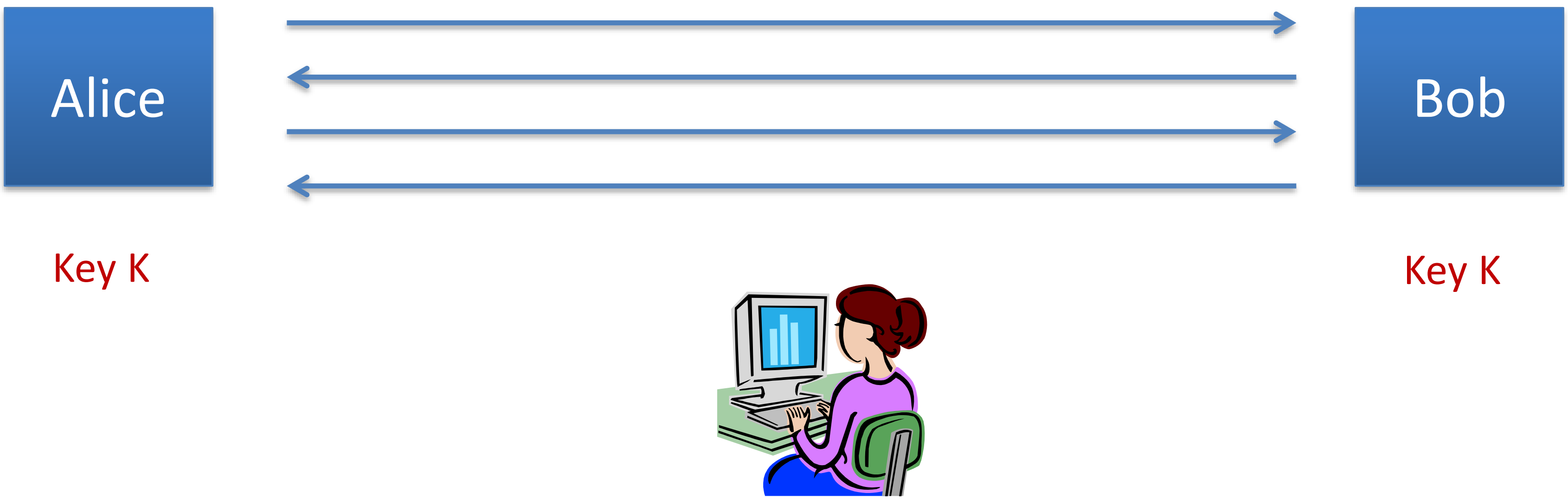
- Design MACs and digital signatures
- Merkle trees (Blockchain, Git)
- Password management



Files
Transactions
Blocks

Takeaway 10: Key exchange without trusted party is possible!

Goal: Alice and Bob want shared secret, unknown to eavesdropper



Diffie-Hellman key exchange

Public key encryption

Takeaway 11: Public-key cryptography relies on number theory

Consider the set of integers: $C(n) := \{ N = p \cdot q ,\ p,q$ are n-bit primes $\}$

RSA assumption: Taking modular roots $\mathbf{c^{1/e}}$ **in** $\mathbf{Z^*_N}$ is hard

Let **G** be a finite cyclic group and **g** generator of G

$$G = \{ 1 , g , g^2 , g^3 , \ldots , g^{q-1} \} , \text{order}(G) = q$$

DDH assumption: For all PPT adversaries A:

$| \Pr[A(g^x ,g^y ,g^{xy}) = 1] - \Pr[A(g^x ,g^y ,g^z) = 1] |$ is negligible.

x, y, z are chosen at random in $\{1,\ldots q-1\}$

Takeaway 12: Textbook RSA is insecure

Textbook RSA encryption:

- public key: **(N,e)**
- secret key: **(N,d)**

Encrypt: $\mathbf{c \longleftarrow m^{e} \bmod N}$

Decrypt: $\mathbf{c^{d} \longrightarrow m \bmod N}$

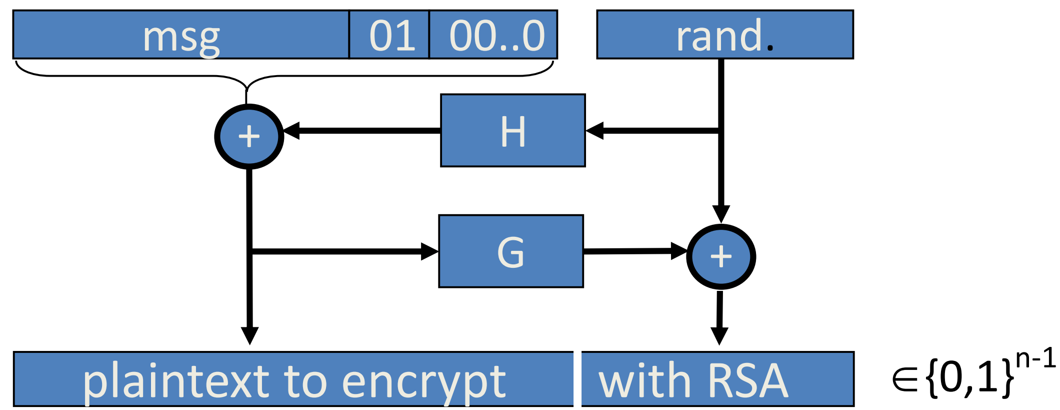
Insecure cryptosystem !!

- Is not CPA secure and many attacks exist
- Deterministic encryption

⇒ The RSA trapdoor permutation is not an encryption scheme !

Takeaway 13: RSA trapdoor is a building block for secure encryption

check pad
on decryption.
reject CT if invalid.



Theorem [FOPS'01] : RSA is a trapdoor permutation $\Rightarrow$
RSA-OAEP is CCA secure when H,G are *random functions*

in practice: use SHA-256 for H and G

Takeaway 14: Converting Diffie-Hellman to public-key encryption

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

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

Alice

choose random $\mathbf{x}$ in $\{1,\ldots,q\}$

$h = g^x$ →

Bob

choose random $\mathbf{y}$ in $\{1,\ldots,q\}$

compute $k=g^{xy} = h^y$

$\mathrm{Enc}(m) = [\ u=g^y, c= k \cdot m\ ]$

←

To decrypt (u,c):

compute $k = u^x$

and decrypt $m = k^{-1} \cdot c$

El-Gamal encryption scheme

CPA secure based on DDH assumption

Takeaway 15: RSA trapdoor can be used to design digital signatures

$N = pq$, such that **p** and **q** are large random primes
e is such that $\gcd(e, \phi(N)) = 1$
d is such that $ed = 1 \pmod{\phi(N)}$

$\mathrm{Sign}_d: Z_N^* \rightarrow Z_N^*$ is defined as:

$$\mathrm{Sign}(m) = H(m)^d \bmod N.$$

Ver_e is defined as:

$$\mathrm{Ver}_e(m,\sigma) = \text{yes} \ \text{ iff } \sigma^e = H(m) \pmod{N}$$

Hash-and-sign paradigm

Takeaway 16: Cryptographic design is modular

TLS Handshake

1. Get server public key — PKI

2. Set up pre-master secret — RSA public-key encryption | Diffie-Hellman

3. Derive 4 secret keys — PRG | Key derivation function

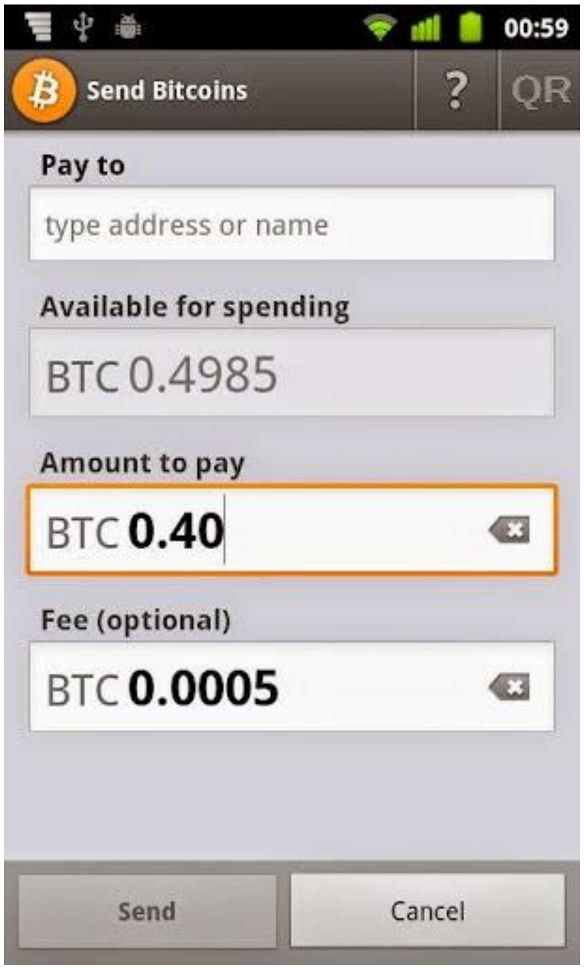
TLS Record

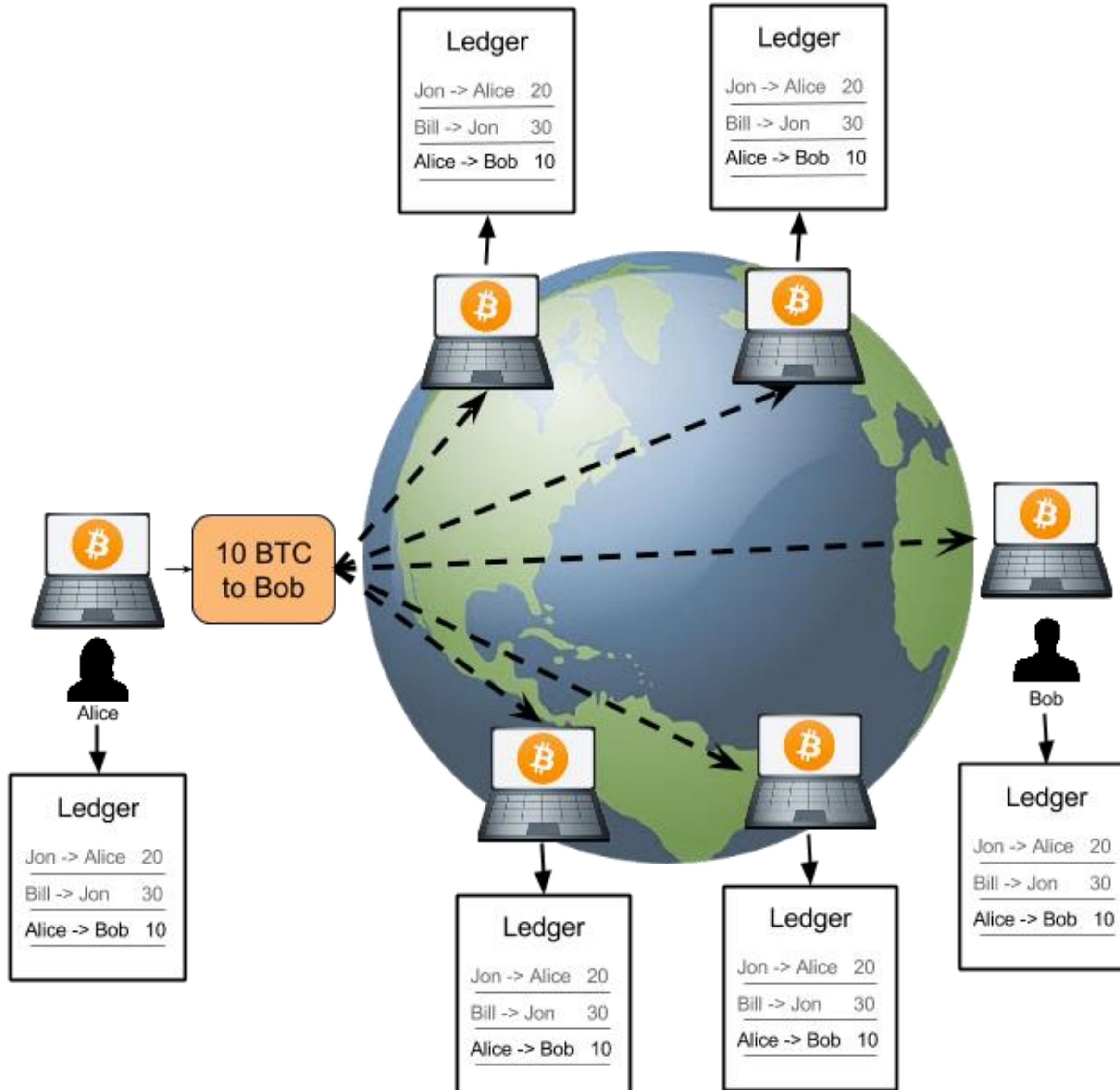
4. Secure communication — Authenticated encryption

Composition

CPA	CBC-AES	CTR-AES
Secure MAC	CBC-MAC	HMAC

Takeaway 17: Distributed ledger applications on the rise





Cryptographic Primitives

Cryptographic PRG

outputs:

a random string **r**

or

G(s) (where **s** random)

0 if he thinks it's **r**

1 if he thinks it's **G(s)**

Should not be able to distinguish...

Definition

n – a parameter

s – a variable distributed uniformly over $\{0,1\}^n$

r – a variable distributed uniformly over $\{0,1\}^{\ell(n)}$

Definition: **G** is a **secure PRG** if for every PPT algorithm **D** we have:

$$| \Pr[D(G(s)) = 1] - \Pr[D(r) = 1] |$$

is negligible in **n**.

Pseudorandom Functions

- We say that F is a **pseudorandom function (PRF) family** if for all **PPT distinguisher D** the probability to correctly distinguish **scenario 0** from **scenario 1** is **negligible**.

Formally: For all PPT distinguisher **D**:

| Pr[D outputs "1" in scenario 0] – Pr[D outputs "1" in scenario 1]|

is negligible in **n**

$$|Pr\left[D^{F_k(\cdot)}(n) = 1\right] - Pr[D^{f(\cdot)}(n) = 1]| \leq negl(n)$$

Polynomial number of queries to oracle

CPA security definition

- Experiment $\mathrm{Exp}_{\Pi,A}^{\mathrm{CPA}}(n)$:
  1. Choose $k \leftarrow^{R} Gen(1^n)$
  2. $m_0, m_1 \leftarrow A_1^{Enc_k(\cdot)}(\cdot)$
  3. $b \leftarrow^{R} \{0,1\}; c \leftarrow Enc_k(m_b)$
  4. $b' \leftarrow A_2^{Enc_k(\cdot)}(m_0, m_1, c)$
  5. Output 1 if $b = b'$ and 0 otherwise

We say that **(Enc,Dec)** is **chosen-plaintext attack (CPA) secure** if

For every **PPT** adversary $A = (A_1, A_2)$:

|Pr[$\mathrm{Exp}_{\Pi,A}^{\mathrm{CPA}}(n)$ = 1]- ½ | negligible in n

CCA security definition

- Experiment $\mathrm{Exp}^{\mathrm{CCA}}_{\Pi,A}(n)$:
  1. Choose $k \leftarrow^R Gen(1^n)$
  2. $m_0, m_1 \leftarrow A_1^{Enc_k(\cdot),Dec_k(\cdot)}(\cdot)$
  3. $b \leftarrow^R \{0,1\}; c \leftarrow Enc_k(m_b)$
  4. $b' \leftarrow A_2^{Enc_k(\cdot),Dec_k(\cdot)}(m_0, m_1, c)$
  5. Output 1 if $b = b'$ and 0 otherwise

Adversary can not submit c to decryption oracle

We say that **(Enc,Dec)** is **chosen-ciphertext attack (CCA) secure** if

For every **PPT** adversary $A = (A_1, A_2)$:

|Pr[$\mathrm{Exp}^{\mathrm{CCA}}_{\Pi,A}(n)$ = 1]- ½ | negligible in n

Security experiment for MAC

- Experiment $\mathrm{Exp}^{\mathrm{MAC}}_{\Pi,A}(n)$:
  1. Choose $k \leftarrow Gen(n)$
  2. *m,t* $\leftarrow A^{Tag()}(n)$
  3. Output 1 if Ver(*m*,*t*) = 1 and *m* was not queried to the Tag() oracle
  4. Output 0 otherwise

We say that **(Gen, Tag,Ver)** is **a secure** MAC if:

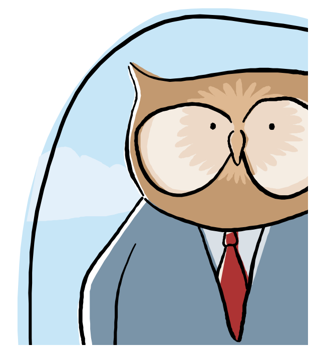
For every **PPT** adversary $A = (A_1, A_2)$:

Pr[$\mathrm{Exp}^{\mathrm{MAC}}_{\Pi,A}(n)$ = 1] is negligible in n

Hash functions – the security definition

security parameter
n

outputs **(m,m')**



H is a **collision-resistant hash function** if

∀

polynomial-time
adversary **A**

Pr[A outputs m, m' such that H(m)=H(m')]
is negligible

Security experiment for Signatures

- Experiment $\mathrm{Exp}_{\Pi,A}^{\mathrm{Sign}}(n)$:
  1. Choose $(pk,sk) \leftarrow Gen(n)$
  2. $m,\sigma \leftarrow A^{Sign_{sk}()}(pk)$
  3. Output 1 if $Ver_{pk}(m, \sigma) = 1$ and m was not queried to the Sign() oracle
  4. Output 0 otherwise

(Gen,Tag,Ver) is **a secure (existential unforgeable)** signature if:

For every **PPT** adversary A:

Pr[$\mathrm{Exp}_{\Pi,A}^{\mathrm{Sign}}(n)$ = 1] is negligible in n


THANK
YOU