Block ciphers

CS 5830

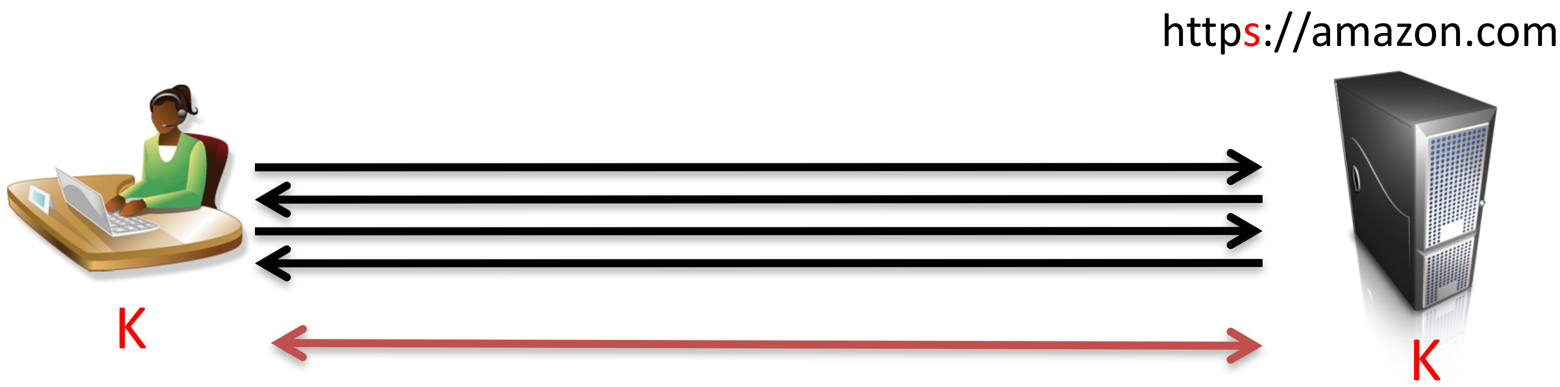
The game plan

- **Part 1: Theory underlying symmetric crypto**
  - Understand definitions, computational security and reductions
  - See constructions showing how to build crypto from weakest-possible assumptions
- **Part 2: Symmetric crypto in practice**
  - How we build and deploy symmetric crypto as used in TLS, elsewhere
- **Part 3: Asymmetric crypto**
  - Public-key encryption, digital signatures, key exchange
- **Part 4 (time allowing): Special topics**
  - Possibilities: anti-censorship, backdoor-resistant cryptography, zero-knowledge, blockchain, etc.

Review so far

- Foundations of symmetric cryptography
  - Shannon security
  - Computational security (reductions)
  - One-way functions
  - Pseudorandom generators (PRGs)
  - Pseudorandom functions (PRFs)
  - Symmetric encryption

How TLS works (high level view)



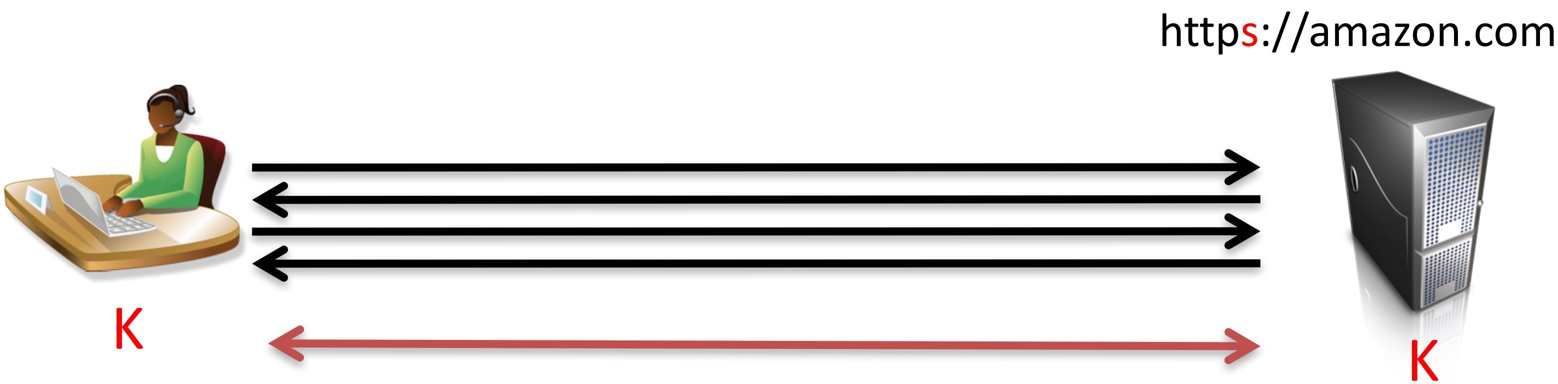
Step 1:
Key exchange protocol to share secret K

Step 2:
Send data via secure channel

Goals of handshake (key exchange protocol):

- Negotiate version
- Negotiate parameters (crypto to use)
- Authenticate server (Is server actually Amazon.com?)
  - Digital signatures and certificates
- Establish shared secret
  - Asymmetric encryption primitives

How TLS works (high level view)



Step 1:
Key exchange protocol to share secret K

Step 2:
Send data via secure channel

Goals of secure channel (record layer protocol):

- Confidentiality
  - Only sender/recipient can learn information about plaintext
- Integrity
  - Only sender/recipient can generate valid ciphertext

Towards a practical record layer

- We saw how to build multi-message secure encryption from PRFs

$$\mathrm{Enc}_K(m):\ \ r \leftarrow U_n\ ;\ \text{Return}\ (\ r,\ m \oplus f_K(r)\ )$$

- How do we build fast PRFs?
  - Blockciphers!
- Symmetric encryption from fast PRFs
  - Extending many-message construction to many message blocks
  - Modes of operations of blockciphers
- The perils of chosen-ciphertext attacks

Blockciphers

Family of permutations, one permutation for each key

$$E : \{0,1\}^k \times \{0,1\}^n \dashrightarrow \{0,1\}^n$$

Use notation $E(K,X) = E_K(X) = Y$

Define inverse $D(K,Y) = D_K(Y) = X$ such that $D_K(E_K(X)) = X$

E,D must be efficiently computable

Key generation: pick K uniformly at random from $\{0,1\}^k$

Nowadays $k \geq 128$

Blockciphers vs. Encryption

Blockcipher	Symmetric encryption
Deterministic	Randomized
Length-preserving (ciphertexts same size as plaintexts)	Length-increasing
Will target being secure as PRFs	Multi-message security (we will expand on this soon)
Leaks message equality	Does not leak message equality

Length-increasing symmetric encryption preferred choice in applications.
Some applications where length-preserving encryption (blockcipher) is required

One-time pad as a blockcipher

Family of permutations, one permutation for each key

$E : \{0,1\}^k \times \{0,1\}^n \dashrightarrow \{0,1\}^n$

Let $E_K(X) = X \oplus K$ Then $D_K(Y) = Y \oplus K$

This defines a family of permutations, one for each key.

Efficient to compute

Is this secure as a PRF?

Data encryption standard (DES)

Originally called Lucifer
- team at IBM
- input from NSA
- standardized by NIST in 1976

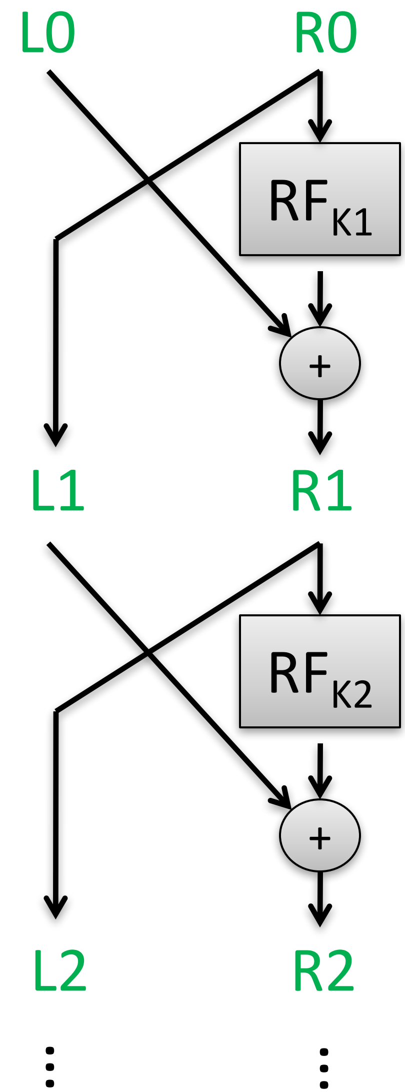
$n = 64$
$k = 56$

Number of keys:
72,057,594,037,927,936

Split 64-bit input into L0,R0 of 32 bits each

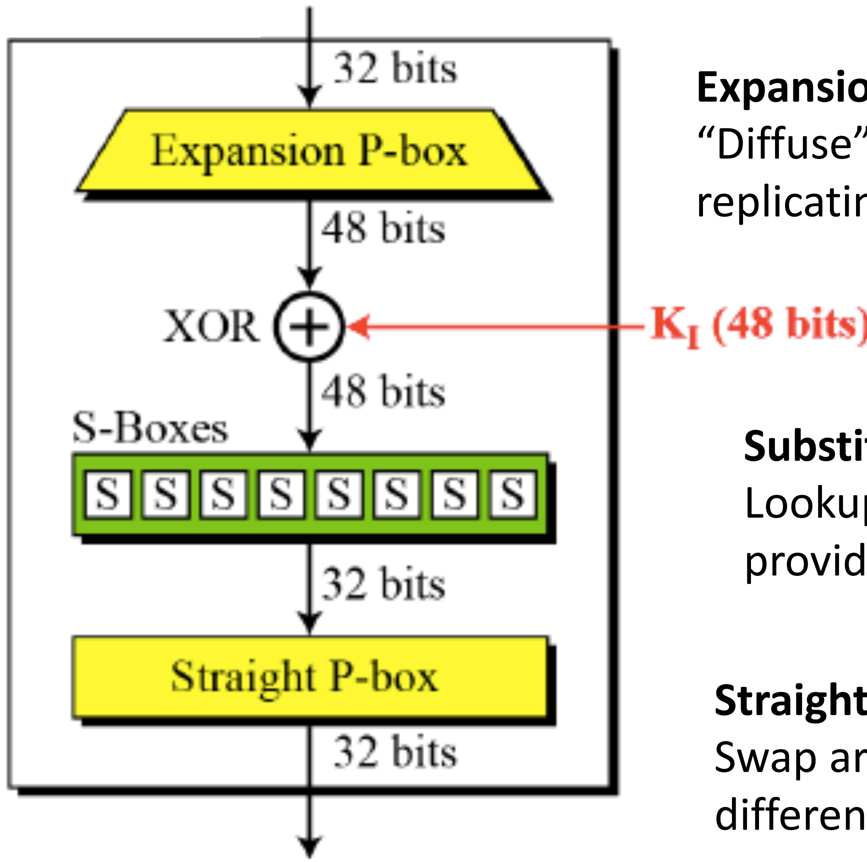
Repeat Feistel round 16 times

Each round applies function F using separate round key K1, K2, ..., K16 that are derived from K



Round functions in DES

X



$RF_{Ki}(X)$

Expansion permutation box:
"Diffuse" X across 48 bits by replicating bits

Substitution box:
Lookup tables that provide non-linearity

Straight permutation box:
Swap around values so they go to different S-boxes in next round

Best attacks against DES

Attack	Attack type	Complexity	Year
Biham, Shamir	Chosen plaintexts, recovers key	2^{47} plaintext, ciphertext pairs	1992
DESCHALL	Brute-force attack	$2^{56/4}$ DES computations 41 days	1997
EFF Deepcrack	Brute-force attack	~4.5 days	1998
Deepcrack + DESCHALL	Brute-force attack	22 hours	1999

- DES is still used in some places
- 3DES (use DES 3 times in a row with more keys) expands keyspace and still used widely in practice

Secure | https://crack.sh

crack.sh

HOME GET CRACKING 100% GUARANTEE THE TECHNOLOGY FAQ CONTACT

THE WORLD'S FASTEST DES CRACKER

In 1998 the Electronic Frontier Foundation built the EFF DES Cracker. It cost around $250,000 and involved making 1,856 custom chips and 29 circuit boards, all housed in 6 chassis, and took around 9 days to exhaust the keyspace. Today, with the advent of Field Programmable Gate Arrays (FPGAs), we've built a system with 48 Virtex-6 LX240Ts which can exhaust the keyspace in around 26 hours, and have provided it for the research community to use. Our hope is that this will better demonstrate the insecurity of DES and move people to adopt more secure modern encryption standards.

The History

- DES (under name Lucifer) designed by IBM in 1970s
- NIST standardized it
  - NSA evaluated it and made suggested changes to shorten key length to 56 bits and changes to S-boxes
  - Many public criticisms of these changes, though S-boxes change actually strengthened DES
- AES competition run by NIST (1997-2000)
  - Many good submissions (15 total submissions)
  - AES chosen as winner

Advanced Encryption Standard (AES)

Rijndael (Rijmen and Daemen)

$n = 128$

$k = 128, 192, 256$

Number of keys for k=128:
340,282,366,920,938,463,463,374,607,431,768,211,456

Substitution-permutation design.
For k=128 uses 10 rounds of:

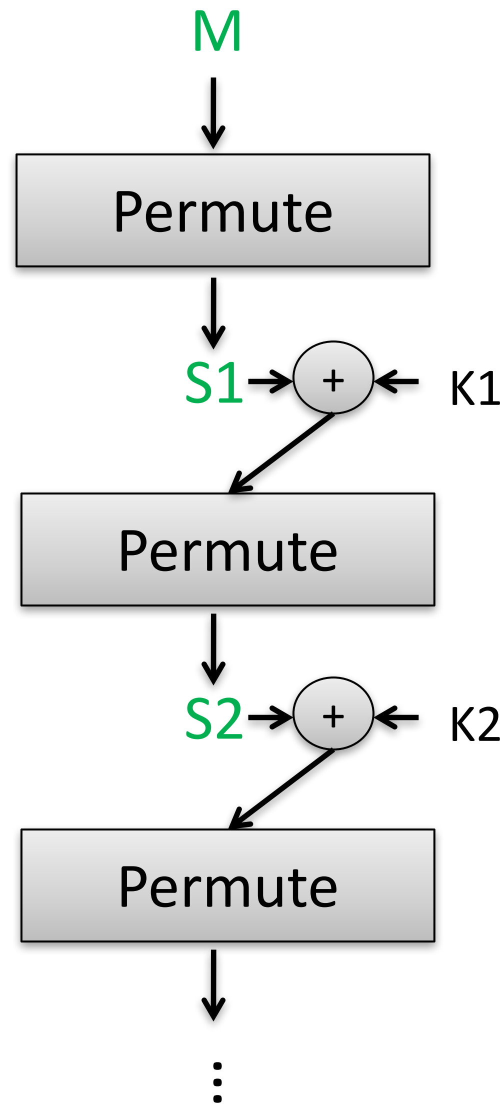
1) Permute:

SubBytes (non-linear S-boxes)

ShiftRows + MixCols (invertible linear transform)

2) XOR in a round key derived from K

(Actually last round skips MixCols)



Best attacks against AES

Brute-force attack (try all keys): worst case time about 2^{128}

Attack	Attack type	Complexity	Year
Bogdanov, Khovratovich, Rechberger	chosen ciphertext, recovers key	$2^{126.1}$ time + some data overheads	2011

No direct attacks of practical interest known
Side-channel attacks do exist, need to implement carefully

Instantiating PRF with AES

Recall our multi-message encryption:

$\underline{Enc_K(m):}$
$r \leftarrow U_n$
Return (r, $m \oplus f_K(r)$)

Instantiating PRF with AES

Recall our multi-message encryption:

<u>$Enc_K(m)$:</u>
$r \leftarrow U_n$
Return ($r, m \oplus AES_K(r)$)

As fast as AES!
Only encrypts messages of n bits

This is provably multi-message secure
if AES is secure PRF

We will make this assumption, and trust that no cryptanalysts can't find better attacks

Two encryption applications

We'll look closely at two encryption applications:

- **Length-preserving encryption**
  - Useful for cases where ciphertexts must be same length as plaintexts.
  - Should only be used when absolutely needed

- **Length-extending encryption (used for TLS)**
  - Insecure variants: CTR mode, ECB mode, CBC mode
  - We'll build secure ones in a few lectures

Example: Credit card number encryption

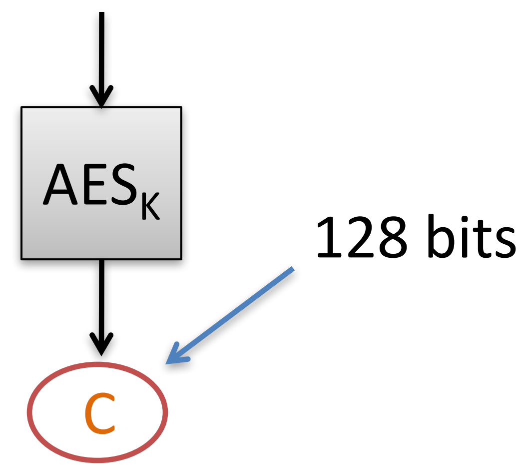
Jane Doe	1343-1321-1231-2310
Thomas Ristenpart	9541-3156-1320-2139
John Jones	5616-2341-2341-1210
Eve Judas	2321-4232-1340-1410

Database schemas and software require <= 16 decimal digits and valid Luhn checksum

$AES_K : \{0,1\}^{128} \longrightarrow \{0,1\}^{128}$

Ciphertexts are too big for replacing plaintext within database!

M = 2321-4232-1345-1415



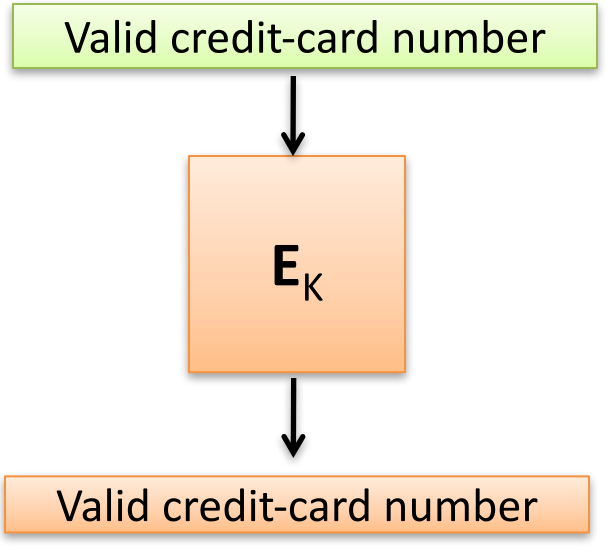
Example: Credit card number encryption

Jane Doe	
Thomas Ristenpart	
John Jones	
Eve Judas	

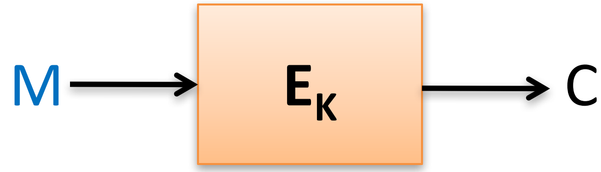
Database schemas and software require <= 16 decimal digits and valid Luhn checksum

Encryption tool whose ciphertexts are also credit-card numbers

$$E_K : [0..9]^{16} \longrightarrow [0..9]^{16}$$



Format-preserving encryption (FPE)



Enjoys format N

Enjoys format N

Disk sectors / payment card numbers just two examples
Some others:

1) Valid addresses for a certain country

2) 4096-byte disk sectors

3) Assigned Social Security Numbers (9 digits, without leading 8 or 9)

4) Composition of (1) and (3)

How to build FPE on 48 bits?

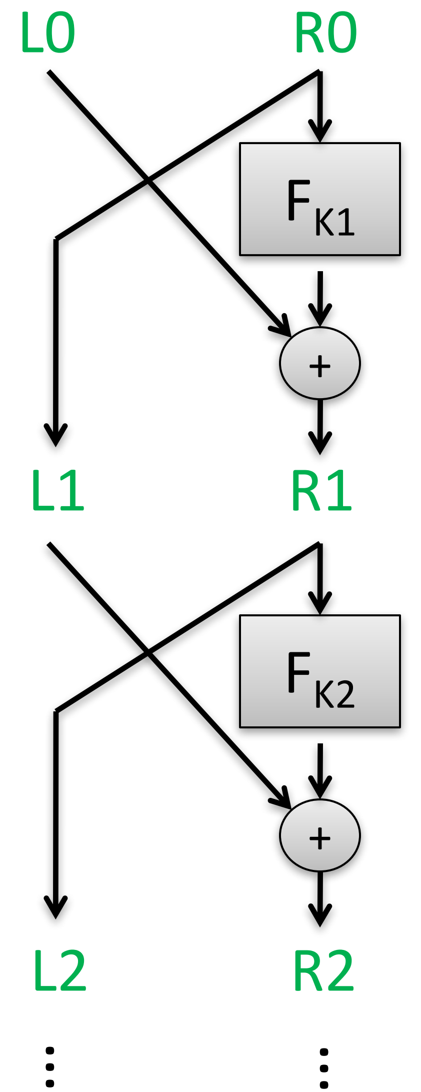
Special case of FFX encryption

Input M = 48 bits
L0 = 24 bits
R0 = 24 bits

$F_{K1}(R) = AES(K, 1 \; || \; R)$
$F_{K2}(R) = AES(K, 2 \; || \; R)$
…

Take XOR mod 2^{24}

Use 10 rounds



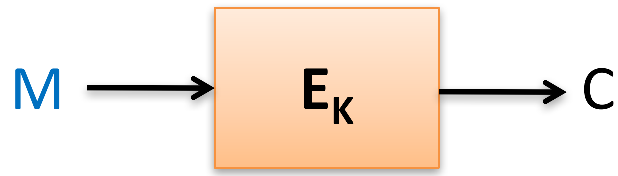
Balanced Feistel security in theory

- Luby & Rackoff showed that if round functions are PRFs and n is relatively large, then
  - 3 rounds suffice for chosen-plaintext attack security in sense of pseudorandom permutation
  - 4 rounds suffice for chosen-ciphertext attack security pseudorandom permutation
  - Proofs hold up to $q \approx 2^{n/4}$

- Sometimes n is not very large:
  - FFX designers suggested 10 rounds as heuristic

FPE now widely used in practice

Security problems with length-preserving encryption?



Looks like a random string

But determinism has problems:

	Plaintext	Ciphertext
Jane Doe	1343-1321-1231-2310	1049-9310-3210-4732
Thomas Ristenpart	9541-3156-1320-2139	7180-4315-4839-0142
John Jones	2321-4232-1340-1410	5731-8943-1483-9015
Eve Judas	1343-1321-1231-2310	1049-9310-3210-4732

Length-extending encryption security

- Not a bit of information about plaintext leaked
  - Equality of plaintexts hidden
  - Even in case of active attacks (we'll get to this)
    - Padding oracles we will see later
- Eventually: authenticity of messages as well
  - Decryption should reject modified ciphertexts