Lecture 3 – Blockciphers and key recovery security

CS-466 Applied Cryptography

Adam O'Neill

Setting the Stage

Perfect security => keys as long as messages.

Setting the Stage

Perfect security => keys as long as messages.

To get around this, from now on we move to the setting of computationally-bounded adversaries.

Setting the Stage

Perfect security => keys as long as messages.

To get around this, from now on we move to the setting of computationally-bounded adversaries.

Today: first lower-level primitive, **blockciphers**

Let $n \geq 1$ be an integer. Let $X_1, \ldots, X_n$ and Y be (non-empty) sets.

By $f\colon X_1 \times \cdots \times X_n \to Y$ we denote that f is a function that

- Takes inputs $x_1, \ldots, x_n$, where $x_i \in X_i$ for $1 \leq i \leq n$
- and returns an output $y = f(x_1, \ldots, x_n) \in Y$.

We call n the number of inputs (or arguments) of f. We call $X_1 \times \cdots \times X_n$ the domain of f and Y the range of f.

Example: Define $f\colon \mathbf{Z}_2 \times \mathbf{Z}_3 \to \mathbf{Z}_3$ by $f(x_1, x_2) = (x_1 + x_2) \bmod 3$. This is a function with $n = 2$ inputs, domain $\mathbf{Z}_2 \times \mathbf{Z}_3$ and range $\mathbf{Z}_3$.

Functions with multiple inputs

What is a Blockcipher?

Let

$$E\colon \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$$

key length → k; block-length → ℓ

be a function taking a key K and input x to return output $E(K,x)$. For each key K we let $E_K\colon \{0,1\}^\ell \to \{0,1\}^\ell$ be the function defined by

$$E_K(x) = E(K,x)\ .$$

- $E_K(\cdot)$ is a permutation $\forall K$ → on $\{0,1\}^\ell$
- $E_K(\cdot)$, $E_K^{-1}(\cdot)$ is efficiently computable

Blockcipher Examples

$k, \ell \in \mathbb{N}$

$\mathcal{E}_K(x) = x \quad \forall K \in \{0,1\}^k \quad \forall x \in \{0,1\}^\ell$

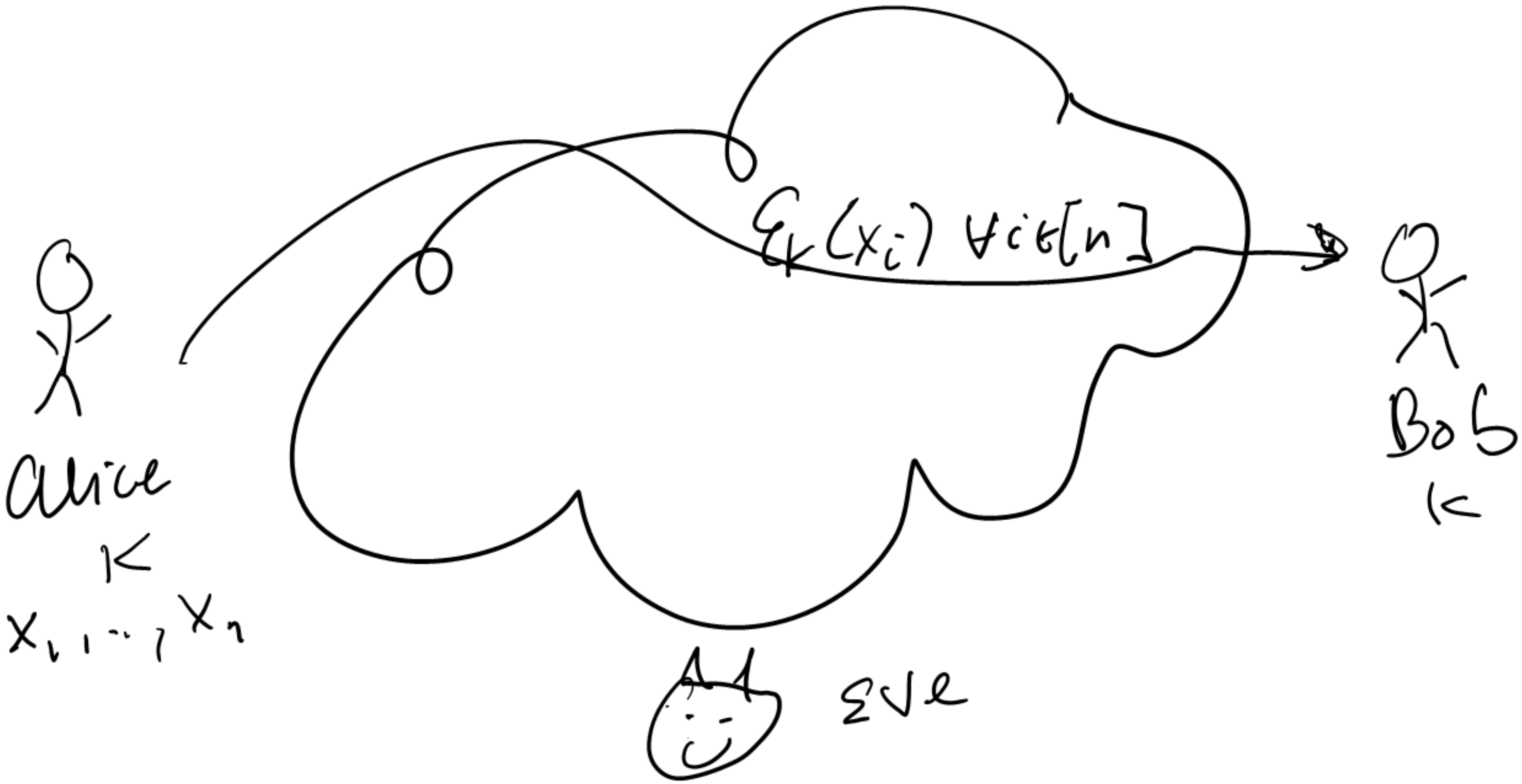
$\mathcal{E} \quad k = \ell$

$\mathcal{E}_K(x) = K \oplus x \qquad \mathcal{E}_K^{-1}(c) = c \oplus K$

$= K \oplus x \oplus K$

$\mathcal{E}_K(x) = \overline{x} \qquad \mathcal{E}_K^{-1}(c) = \overline{c}$

Blockcipher Usage



For security:

- hard for Eve to guess K
- detect common message characters
- recover any of the x_i's.

Blockcipher Usage

Shannon's design criteria

confusion : Every bit of the key should influence the entire output

diffusion : Every bit of the input should influence the entire output

One can easily see that the simple examples we gave do not meet these!!?

DES

- The first example of a “general purpose” blockcipher realizing Shannon’s criteria

DES

1970s

- The first example of a "general purpose" blockcipher realizing Shannon's criteria
- Influenced by earlier designs of Feistel and Coppersmith

DES

- The first example of a "general purpose" blockcipher realizing Shannon's criteria
- Influenced by earlier designs of Feistel and Coppersmith
- Still important and even used today

History of DES

1972 – NBS (now NIST) asked for a block cipher for standardization

1974 – IBM designs Lucifer

Lucifer eventually evolved into DES.

Widely adopted as a standard including by ANSI and American Bankers association

Used in ATM machines

Replaced (by AES) in 2001.

DES Parameters

Key Length $k = 56$

Block length $\ell = 64$

So,

$$\mathsf{DES}\colon \{0,1\}^{56} \times \{0,1\}^{64} \to \{0,1\}^{64}$$

$$\mathsf{DES}^{-1}\colon \{0,1\}^{56} \times \{0,1\}^{64} \to \{0,1\}^{64}$$

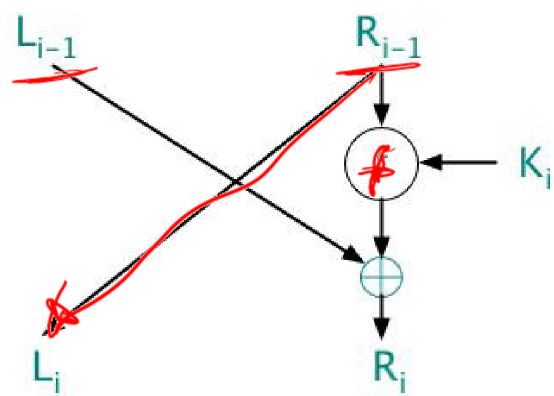
DES Construction

function $\mathsf{DES}_K(M)$ // $|K| = 56$ and $|M| = 64$
  $(K_1, \ldots, K_{16}) \leftarrow KeySchedule(K)$ // $|K_i| = 48$ for $1 \leq i \leq 16$
  $M \leftarrow IP(M)$
  Parse M as $L_0 \| R_0$ // $|L_0| = |R_0| = 32$
  for $i = 1$ to 16 do
    $L_i \leftarrow R_{i-1}$; $R_i \leftarrow f(K_i, R_{i-1}) \oplus L_{i-1}$
  $C \leftarrow IP^{-1}(L_{16} \| R_{16})$
  return C

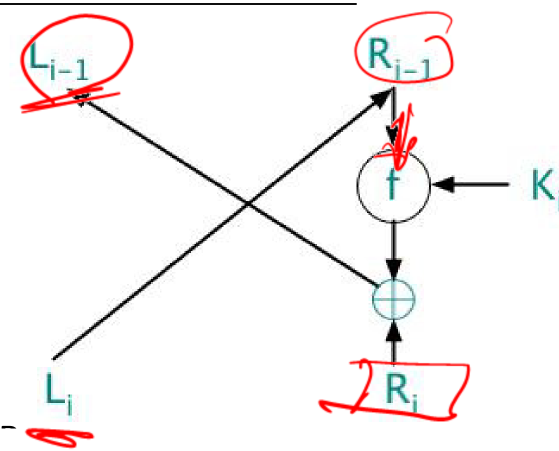
Computes one Feistel round

16 Feistel rounds

Round i:



Invertible given K_i:



Key-Recovery Security

Let E: $\{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$ be a blockcipher. It is known to the adversary A.

Def: We say that $K' \in \{0,1\}^k$ is *consistent* with $(M_1, C_1), \ldots, (M_q, C_q)$ if $E(K', M_i) = C_i$ for all $1 \leq i \leq q$.

we will be concerned with whether the adversary can recover a CONSISTENT KEY after seeing some communication or other data

Note that it seems more imprtant whether the adversary recovers the TARGET (actual) key used for the communication but one can show with enough commrcation the only consistent key is the target key

Adversary's Goal, Capability, and Resources

We always judge an adversary in terms of goal capabilities and resources

goal: What does the adversary have to do to WIN (break the scheme)

capabilities: How can the adversary interact with the scheme? What is it allowed to choose and what is unknown?

resources: Running-time, code size memory usage (POWER + MONEY!!) HOW MANY TIMES IT CALLS a function (oracle)

Formalizing Cryptographic Games

We will use cryptographic games following a particular format. Every game is played by an ADVERSARY and is specified by the following:

* Initialize procedure : Called at the beginning of the game
  result passed to adversary

* Functions $F_1, \ldots, F_n$: Given to the adversary as oracles

* Finalize Procedure : Called at the end of the game
  adversary's output passed to finalize

Consistent-key recovery game

The Game

how clever A is

Let $E: \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$ be a blockcipher and A an adversary.

Game KR_E
 	
procedure Initialize
$K \xleftarrow{\$} \{0,1\}^k$; $i \leftarrow 0$
 	
procedure Fn(M)
$i \leftarrow i+1$; $M_i \leftarrow M$
$C_i \leftarrow E(K, M_i)$
Return C_i
	procedure Finalize(K')
	$\mathrm{win} \leftarrow$ true
	For $j = 1, \ldots, i$ do
	    If $E(K', M_j) \neq C_j$ then $\mathrm{win} \leftarrow$ false
	    If $M_j \in \{M_1, \ldots, M_{j-1}\}$ then $\mathrm{win} \leftarrow$ false
	Return win

$M \in \{0,1\}^\ell$

depends on: - the performance in the game

$$\mathbf{Adv}^{\mathrm{kr}}_E(A) = \Pr[\mathrm{KR}^A_E \Rightarrow \text{true}]$$

- high advantage function means the adversary is doing well, the scheme is vulnerable to the adversary's attack
- low advantage means the scheme resists this specific attack

Exhaustive Key Search

Let $T_1, \ldots, T_{2^k}$ be a list of all k bit keys and let $\langle i \rangle$ denote the ℓ-bit binary representation of integer i. Let $1 \leq q \leq 2^\ell$ be a parameter.

The q-query exhaustive key-search adversary

Adversary EKSq:

enumeration of $\{0,1\}^\ell$

For i=1 to q do:

$y_i \leftarrow Fn(\langle i \rangle)$ // call $\langle i \rangle = x_i$

For j=1 to 2^k

if $\forall i \in [q]$ $E_{T_j}(x_i) = y_i$

Return T_j

Time_E is the time to compute E

q queries to Fn

running-time: $2^k \cdot q \cdot \text{Time}_E$

Exhaustive Key-Search on DES

DES can be computed at 1.6 Gbits/sec in hardware.

DES plaintext = 64 bits

Chip can perform $(1.6 \times 10^9)/64 = 2.5 \times 10^7$ DES computations per second

Expect A_{eks} $(q = 1)$ to succeed in 2^{55} DES computations, so it takes time

$$\frac{2^{55}}{2.5 \times 10^7} \approx 1.4 \times 10^9 \text{ seconds}$$

$$\approx 45 \text{ years!}$$

Key Complementation $\Rightarrow$ 22.5 years

But this is prohibitive. Does this mean DES is secure?

Parallelizing the Attack

Observation: The E computations can be performed in parallel!

In 1993, Wiener designed a dedicated DES-cracking machine:

- $1 million
- 57 chips, each with many, many DES processors
- Finds key in 3.5 hours

Effective key-length

DES: 64-bit keys   64 is actual key length

Suppose there is an attack in time 2^{20}

20-bit effective key length

Increasing Key-Length

Can one use DES to design a new blockcipher with longer effective key-length?

2DES

→ 56+56

Block cipher $2DES : \{0,1\}^{112} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ is defined by

$$2DES_{K_1 K_2}(M) = DES_{K_2}(DES_{K_1}(M))$$

$Time_{2DES}$

→ key-length : 112

→ running-time of EKS^{1}_{2DES} ← 1-query exhaustive search adversary against 2DES

$2^{112} \cdot Time_{2DES}$

effective-length ???

Meet-in-the-Middle Attack

effective - key length 57

DES

3DES

Block ciphers

$$3DES3 : \{0,1\}^{168} \times \{0,1\}^{64} \to \{0,1\}^{64}$$

$$3DES2 : \{0,1\}^{112} \times \{0,1\}^{64} \to \{0,1\}^{64}$$

are defined by

$$3DES3_{K_1 \parallel K_2 \parallel K_3}(M) = DES_{K_3}(DES^{-1}_{K_2}(DES_{K_1}(M))$$

$$3DES2_{K_1 \parallel K_2}(M) = DES_{K_2}(DES^{-1}_{K_1}(DES_{K_2}(M))$$

Meet-in-the-middle attack on 3DES3 reduces its "effective" key length to 112.

DESX

$$DESX_{KK_1K_2}(M) = K_2 \oplus DES_K(K_1 \oplus M)$$

- Key length $= 56 + 64 + 64 = 184$
- "effective" key length $= 120$ due to a 2^{120} time meet-in-middle attack

But again a non-generic attack does better!

DESX

$$DESX_{KK_1K_2}(M) = K_2 \oplus DES_K(K_1 \oplus M)$$

- Key length $= 56 + 64 + 64 = 184$
- "effective" key length $= 120$ due to a 2^{120} time meet-in-middle attack

But again a non-generic attack does better!

DESX

$$DESX_{KK_1K_2}(M) = K_2 \oplus DES_K(K_1 \oplus M)$$

- Key length $= 56 + 64 + 64 = 184$
- "effective" key length $= 120$ due to a 2^{120} time meet-in-middle attack

But again a non-generic attack does better!

https://www.iacr.org/archive/eurocrypt2000/1807/18070595-new.pdf

Generic vs non-generic attacks

The power of non-generic attacks

[Biryukov-Wagner 2000] attacks DESX using *slide attacks*

Known plaintext - $2^{32.5}$ data, $2^{87.5}$ time

Ciphertext only - $2^{32.5}$ data, 2^{95} time

https://www.iacr.org/archive/eurocrypt2000/1807/18070595-new.pdf

The power of non-generic attacks

[Biryukov-Wagner 2000] attacks DESX using *slide attacks*

Known plaintext - $2^{32.5}$ data, $2^{87.5}$ time

Ciphertext only - $2^{32.5}$ data, 2^{95} time

https://www.iacr.org/archive/eurocrypt2000/1807/18070595-new.pdf

[Ashur-Posteuca 2018] attacks DES, 3DES using *linear trails*

Known plaintext - $2^{50.9}$

2^{64} time and data

https://eprint.iacr.org/2018/1219.pdf

Increasing Block-Length?

Not only do non-generic attacks kill DES-constructs effective key-length, we will see later we also want longer block-length.

This seems much harder to do using DES.

Motivated the search for a new blockcipher.

AES History

1998: NIST announces competition for a new block cipher

- key length 128
- block length 128
- faster than DES in software

Submissions from all over the world: MARS, Rijndael, Two-Fish, RC6, Serpent, Loki97, Cast-256, Frog, DFC, Magenta, E2, Crypton, HPC, Safer+, Deal

2001: NIST selects Rijndael to be AES.

AES Construction

function $\mathsf{AES}_K(M)$
    $(K_0, \ldots, K_{10}) \leftarrow \mathit{expand}(K)$
    $s \leftarrow M \oplus K_0$
    for $r = 1$ to 10 do
        $s \leftarrow S(s)$
        $s \leftarrow \mathit{shift\text{-}rows}(s)$
        if $r \leq 9$ then $s \leftarrow \mathit{mix\text{-}cols}(s)$ fi
        $s \leftarrow s \oplus K_r$
    end for
    return s

- Fewer tables than DES
- Finite field operations

Is Key-Recovery Security Enough?