Encryption and Attacks

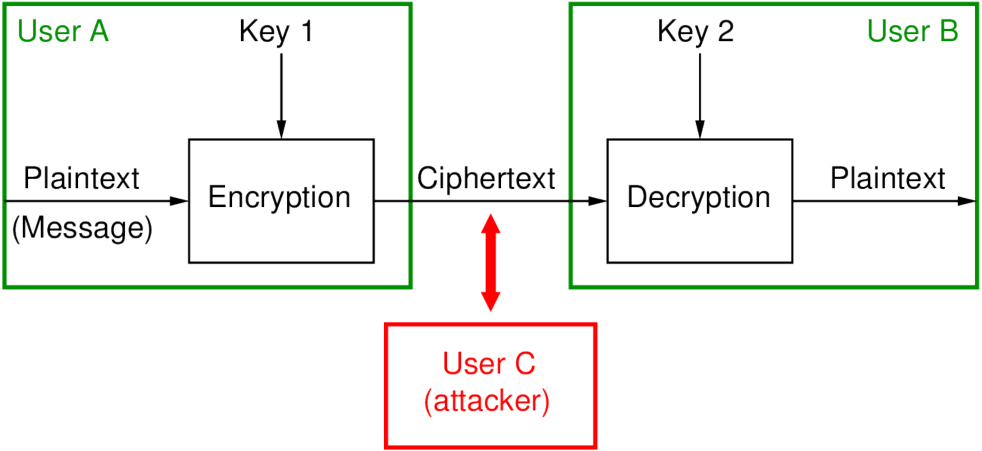
Cryptography

School of Engineering and Technology
CQUniversity Australia

Prepared by Steven Gordon on 04 Jan 2022,
encryption.tex, r1965

Contents

Model of Encryption for Confidentiality

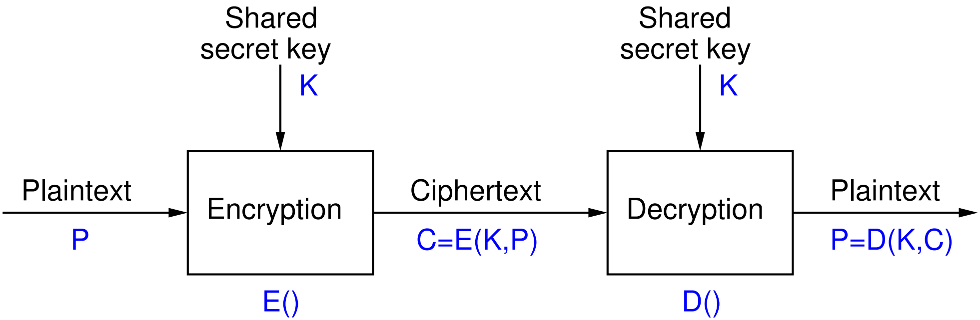


Characterising Ciphers by Number of Keys

Symmetric sender/receiver use same key (single-key, secret-key, shared-key, conventional)

Public-key sender/receiver use different keys (asymmetric)

Symmetric Key Encryption for Confidentiality



Common Operations in Symmetric Ciphers

Substitution replace one element in plaintext with another

Permutation re-arrange elements (also called transposition)

Product systems multiple stages of substitutions and permutations, e.g. Feistel network, Substitution Permutation Network (SPN)

Characterising Ciphers by Processing Plaintext

Block cipher process one block of elements at a time, typically 64 or 128 bits

Stream cipher process input elements continuously, e.g. 1 byte at a time, by XOR plaintext with keystream

Two Important Symmetric Key Block Ciphers

Data Encryption Standard (DES) Became a US government standard in 1977 and widely used for more than 20 years; key is too short

Advanced Encryption Standard (AES) Standardised a replacement of DES in 1998, and now widely used. Highly recommended for use.

Common Symmetric Key Block Ciphers

Cipher	Year	Designers	Block Size	Key Size	Design
DES	1977	IBM/NSA	64	56	Feistel
IDEA	1991	Lai and Massey	64	128	Other
Blowfish	1993	Schneier	64	32-448	Feistel
RC5	1994	Rivest	64, 128	-2040	Feistel-like
CAST-128	1996	Adams and Tavares	64	40-128	Feistel
Twofish	1998	Schneier et al	128	128, 192, 256	Feistel
Serpent	1998	Anderson et al	128	128, 192, 256	SPN
CAST-256	1998	Adams and Tavares	128	-256	Feistel
RC6	1998	Rivest et al	128	128, 192, 256	Feistel
AES	1998	Rijmen and Daemen	128	128, 192, 256	SPN
3DES	1998	NIST	64	56,112,168	Feistel
Camellia	2000	Mitsubishi/NTT	128	128, 192, 256	Feistel

Contents

Aims and Knowledge of the Attacker

- Study of ciphers and attacks on them is based on assumptions and requirements
  - Assumptions about what attacker knows and can do, e.g. intercept messages, modify messages
  - Requirements of the system/users, e.g. confidentiality, authentication
- Normally assumed attacker knows cipher
  - Keeping internals of algorithms secret is hard
  - Keeping which algorithm used secret is hard
- Attacker also knows the ciphertext
- Attacker has two general approaches
  - "Dumb": try all possible keys, i.e. brute force
  - "Smart": use knowledge of algorithm and ciphertext/plaintext to discover unknown information, i.e. cryptanalysis

Worst Case Brute Force Time for Different Keys

Key length	Key space	Worst case time at speed: 10^9/sec	10^{12}/sec	10^{15}/sec
32	2^{32}	4 sec	4 ms	4 us
56	2^{56}	833 days	20 hrs	72 sec
64	2^{64}	584 yrs	213 days	5 hrs
80	2^{80}	10^7 yrs	10^4 yrs	38 yrs
100	2^{100}	10^{13} yrs	10^{10} yrs	10^7 yrs
128	2^{128}	10^{22} yrs	10^{19} yrs	10^{16} yrs
192	2^{192}	10^{41} yrs	10^{38} yrs	10^{35} yrs
256	2^{256}	10^{60} yrs	10^{57} yrs	10^{54} yrs
26!	2^{88}	10^{10} yrs	10^7 yrs	10^4 yrs

Classifying Attacks Based Upon Information Known

1. Ciphertext Only Attack
2. Known Plaintext Attack
3. Chosen Plaintext Attack
4. Chosen Ciphertext Attack
5. Chosen Text Attack

Ciphertext Only Attack

- Attacker knows:
  - encryption algorithm
  - ciphertext
- Hardest type of attack
- If cipher can be defeated by this, then cipher is weakest

Known Plaintext Attack

- Attacker knows:
  - encryption algorithm
  - ciphertext
  - one or more plaintext–ciphertext pairs formed with the secret key
- E.g. attacker has intercept past ciphertext *and* somehow discovered their corresponding plaintext
- All pairs encrypted with the same secret key (which is unknown to attacker)

Chosen Plaintext Attack

- Attacker knows:
  - encryption algorithm
  - ciphertext
  - plaintext message chosen by attacker, together with its corresponding ciphertext generated with the secret key

Chosen Ciphertext Attack

- Attacker knows:
  - encryption algorithm
  - ciphertext
  - ciphertext chosen by attacker, together with its corresponding decrypted plaintext generated with the secret key
- Attackers aim is to find the secret key (not the plaintext)

General Measures of Security

Unconditionally Secure Ciphertext does not contained enough information to derive plaintext or key

- One-time pad is only unconditionally secure cipher (but not very practical)

Computationally Secure If:

- cost of breaking cipher exceeds value of encrypted information
- or time required to break cipher exceeds useful lifetime of encrypted information
- Hard to estimate value/lifetime of some information
- Hard to estimate how much effort needed to break cipher

Common Metrics for Attacks

Time: usually measured as *number of operations*, since real time depends on implementation and computer specifics

- Operations are encrypts or decrypts; ignore other processing tasks
- E.g. worst case brute force of k-bit key takes 2^k (decrypt) operations

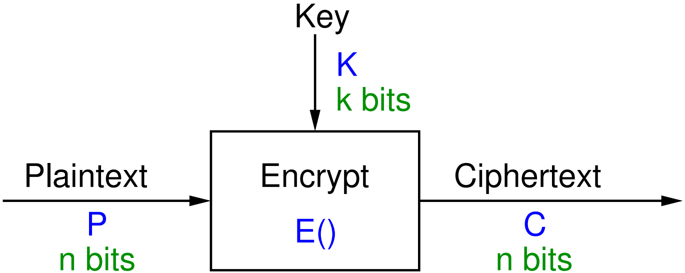
Amount of Memory: temporary data needed to be stored during attack

Known information: number of known plaintext/ciphertext values attacker needs to know in advance to perform attack

Contents

Block Cipher with n bit blocks

- Encrypt a block of plaintext as a whole to produce same sized ciphertext
- Typical block sizes are 64 or 128 bits
- Modes of operation used to apply block ciphers to larger plaintexts



Simple Ideal 2-bit Block Cipher 1

Encryption Cipher 1

P	K0	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	K16	K17	K18	K19	K20	K21	K22	K23
00	10	10	00	10	11	10	11	00	01	01	00	01	11	01	00	00	10	11	11	01	00	01	10	11
01	00	11	10	11	00	00	10	01	10	00	10	11	10	00	11	01	01	01	00	11	11	10	01	01
10	11	00	11	01	10	01	00	10	11	10	01	10	01	11	01	11	00	10	01	00	10	00	11	00
11	01	01	01	00	01	11	01	11	00	11	11	00	00	10	10	10	11	00	10	10	01	11	00	10

Encrypt with Ideal Cipher 1 (exercise)

Encrypt the message *Tokyo* using the above ideal 2-bit block cipher 1 with key K6.

Issues When Applying Block Ciphers

- Encoding/decoding: independent of block cipher, which operate only in binary values
- Mode of operation: typically independent of block cipher, which operate only on a single block
- Repetition of plaintext blocks: undesirable. Make block size larger and use mode of operation that obscures repetition
- Key space: larger block size needed to allow more keys in ideal block cipher
- Implementing an ideal block cipher: how are they generated? can all values be stored?

Simple Ideal 2-bit Block Cipher 2

Encryption Cipher 2

P	K0	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	K16	K17	K18	K19	K20	K21	K22	K23
00	01	01	00	10	11	00	11	11	01	10	01	00	00	10	01	11	11	01	11	10	00	10	00	10
01	10	11	01	01	11	10	10	01	10	11	11	01	11	00	00	00	01	00	10	01	10	00	11	11
10	11	00	11	00	10	11	01	10	00	01	10	10	10	11	11	01	00	10	00	11	01	01	01	00
11	00	10	10	11	01	01	00	00	11	00	00	11	01	01	10	10	10	11	01	00	11	11	10	01

What is plaintext with key K13, ciphertext 11 with ideal cipher 2? (question)

What is plaintext with key K13, ciphertext 11 with ideal cipher 2?

What is plaintext with key K4, ciphertext 11 with ideal cipher 2? (question)

What is plaintext with key K4, ciphertext 11 with ideal cipher 2?

Simple Ideal 2-bit Block Cipher 2 (fixed)

Encryption Cipher 2

P	K0	K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	K13	K14	K15	K16	K17	K18	K19	K20	K21	K22	K23
00	01	01	00	10	11	00	11	11	01	10	01	00	00	10	01	11	11	01	11	10	00	10	00	10
01	10	11	01	01	00	10	10	01	10	11	11	01	11	00	00	00	01	00	10	01	10	00	11	11
10	11	00	11	00	10	11	01	10	00	01	10	10	10	11	11	01	00	10	00	11	01	01	01	00
11	00	10	10	11	01	01	00	00	11	00	00	11	01	01	10	10	10	11	01	00	11	11	10	01

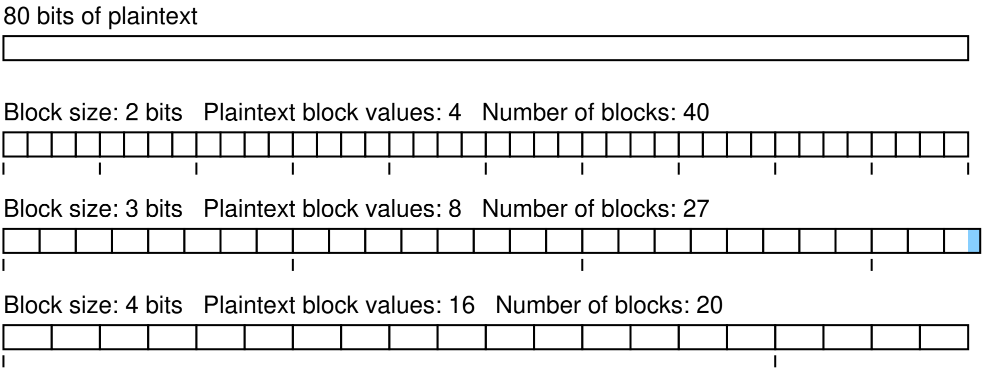
How many bits are needed to represent the key in cipher 2? (question)

The example 2-bit ideal block cipher 2 (as well as cipher 1) list 24 different keys (or mappings from plaintext to ciphertext). How many bits are needed to represent a key for this cipher?

How to reduce repetition of plaintext blocks? (question)

With a 2-bit ideal block cipher, with a long plaintext, many of plaintext blocks will repeat. This is bad for security (see Modes of Operation). What can you change in the design of an ideal block cipher that reduces repetition of plaintext blocks?

Impact of Block Sizes for 80 bit Plaintext



General n-bit Ideal Block Cipher

- n-bit block cipher takes n bit plaintext and produces n bit ciphertext
- 2^n possible different plaintext blocks
- Encryption must be reversible (decryption possible)
- Number of permutations of plaintext (and number of keys) is $2^n!$
- Design trade-offs:
  - Large block size to reduce plaintext repetitions (64-bits is good)
  - Key space large enough to avoid brute force, but small enough to make distribution practical
  - Small block size to simplify implementation

Ideal 64-bit Block Cipher (exercise)

Consider an ideal 64-bit block cipher. How many different different keys are possible? How many bits are needed to store a single key? How much space is required to store the mappings?

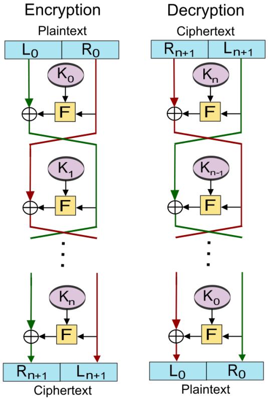
Feistel Structure for Block Ciphers

- Ideal block ciphers are not practical
- Feistel proposed applying two or more simple ciphers in sequence so final result is cryptographically stronger than component ciphers
- n-bit block length; k-bit key length; 2^k transformations
- Feistel cipher alternates: substitutions, transpositions (permutations)
- Applies concepts of diffusion and confusion
- Applied in many ciphers today
- Approach:
  - Plaintext split into halves
  - Subkeys (or round keys) generated from key
  - Round function, F, applied to right half
  - Apply substitution on left half using XOR
  - Apply permutation: interchange to halves

Diffusion and Confusion

- Diffusion
  - Statistical nature of plaintext is reduced in ciphertext
  - E.g. A plaintext letter affects the value of many ciphertext letters
  - How: repeatedly apply permutation (transposition) to data, and then apply function
- Confusion
  - Make relationship between ciphertext and key as complex as possible
  - Even if attacker can find some statistical characteristics of ciphertext, still hard to find key
  - How: apply complex (non-linear) substitution algorithm

Feistel Encryption and Decryption



Credit: Amirki, `https://commons.wikimedia.org/wiki/File:Feistel_cipher_diagram_en.svg`, CC BY-SA 3.0

Using the Feistel Structure

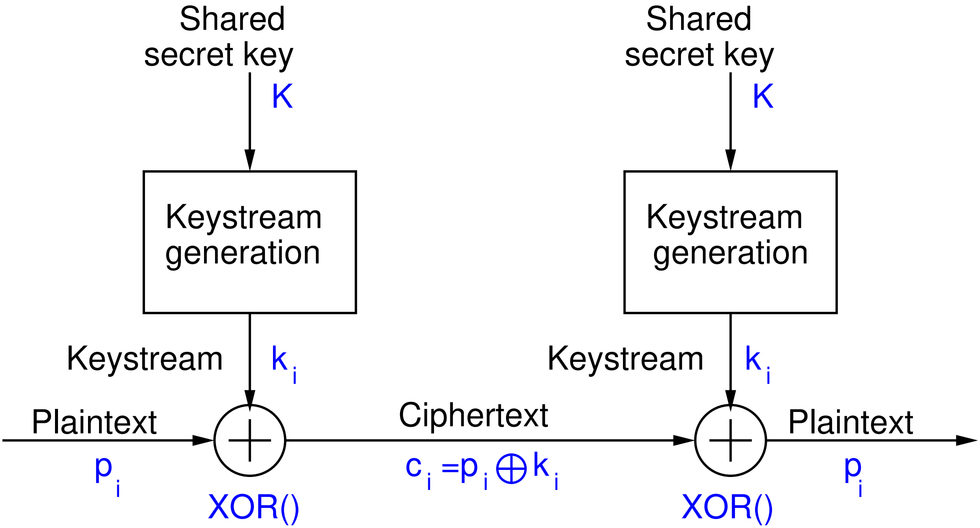
- Exact implementation depends on various design features
  - Block size, e.g. 64, 128 bits: larger values leads to more diffusion
  - Key size, e.g. 128 bits: larger values leads to more confusion, resistance against brute force
  - Number of rounds, e.g. 16 rounds
  - Subkey generation algorithm: should be complex
  - Round function F: should be complex
- Other factors include fast encryption in software and ease of analysis
- Trade-off: security vs performance

Contents

Stream Ciphers

- Encrypts a digital data stream one bit or one byte at a time
- One time pad is example; but practical limitations
- Typical approach for stream cipher:
  - Key (K) used as input to bit-stream generator algorithm
  - Algorithm generates cryptographic bit stream (k_i) used to encrypt plaintext
  - k_i is XORed with each byte of plaintext P_i
  - Users share a key; use it to generate keystream

Stream Cipher Encrypt and Decrypt



Key Re-use in Stream Ciphers

- Encrypting two different plaintexts with the same key leads to key re-use attack
  - Attacker intercepts two ciphertexts: $C_1 = P_1 \oplus k_1$ and $C_2 = P_2 \oplus k_1$
  - Properties of XOR: commutative and $A \oplus A = 0$
  - Attacker performs XOR on two ciphertexts
  - $C_1 \oplus C_2 = P_1 \oplus k_1 \oplus P_2 \oplus k_1 = P_1 \oplus P_2$
  - Even without knowing P_1 or P_2, attacker can easily use frequency analysis to discover both
- Solution: Use additional IV that changes for every encryption

When can key re-use attack be successful if IV is used? (question)

If a stream cipher is using a n-bit IV, but the same key, under what conditions is a key re-use attack possible? Assume the IV increments every time an encrypt operation is performed.

Contents

DES and Real Brute Force Attacks

- DES is 64-bit block cipher with 56-bit (effective) key length
- Developed in 1977, recommended standard until 1990's
- Brute force: 2^{56} operations
- Hardware built to perform brute force attack
  - 1998: DeepCrack
  - 2006: COPACABANA

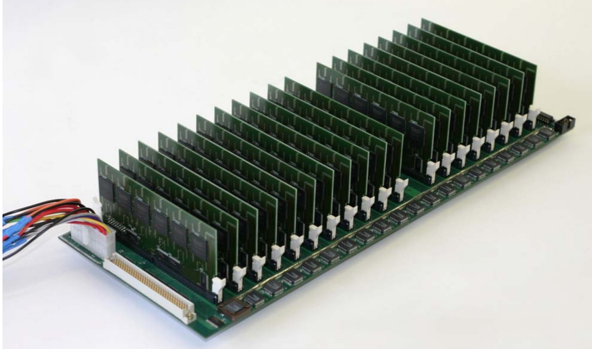
Paul Kocher and DeepCrack

- Developed by EFF
- Cost less than $US250,000
- 80×10^9 keys/sec
- Solved DES challenge in 56 hours
- See `www.cryptography.com` and `www.eff.org`

Credit: Wikimedia, CC0 1.0 Public Domain `https://commons.wikimedia.org/wiki/File:Paul_kocher_deepcrack.jpg`

COPACABANA by SciEngines, 2006

- Joint effort by SciEngines and German universities
- 120 FPGA, 400×10^6 keys/sec/FPGA
- For comparison, a Pentium 4: 2×10^6 keys/sec
- Brute force DES in 8.6 days
- Cost about $US10,000
- See `www.sciengines.com`



Credit: Copyright SciEngines GMBH

Can We Estimate Cost Today?

- Moore's law: computers double speed every 1.5 years
- Alternative: computers halve in cost every 1.5 years
- $US10,000 to brute force DES in 2006
- Cost has halved about 10 times
- Cost to brute force DES in 2020: $10

Contents

RIVYERA S3-5000 by SciEngines, 2013

- Rivyera S3 supported up to 128 Xilinx Spartan-3 FPGAs
- Approx $100 per FPGA (XCS5000)
- AES-128 Brute Force
  - 500×10^6 keys per sec
  - 4×10^6 keys per mW
- Biclique Attack
  - 945×10^6 keys per sec
  - 7.3×10^6 keys per mW

Credit: Copyright SciEngines GMBH

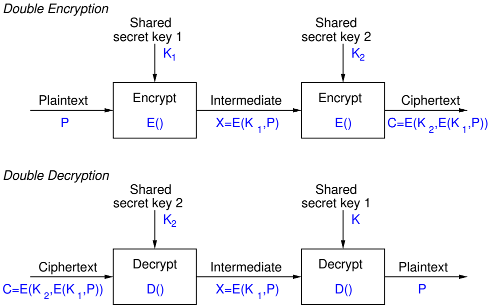
Breaking AES-128 in 2020

- AES-128 has key space of 2^{128}
- 2013: \$US12,800 for 5×10^8 k/s
- Assume: computers double speed every 1.5 years
- 2020: Increase by $2^5 = 32$; 1.6×10^{10} k/s
  - \$12,800: 6.7×10^{20} years
  - \$12,800,000: 6.7×10^{17} years
  - \$12,800,000,000: 6.7×10^{14} years
- Biclique attack about 2 to 4 times faster, but requires 2^{88} known plaintext/ciphertext pairs
- In 2035, cost \$12,800,000,000 to brute force AES-128 in 670,000,000,000 years

Contents

Double Encryption Concept

- Encrypt plaintext with one key, then encrypt output with another key



- Advantage: doubles the key length
  - Single version of cipher has k-bit key
  - Double version of cipher uses two different k-bit keys
  - Worst case brute force: 2^{2k}
- Advantage: uses an existing cipher
- Disadvantage: doubles the processing time
- Problem: double encryption is subject to *meet-in-the-middle* attack

Meet-in-the-Middle Attack

- Double Encryption where key K is k-bits: $C = \mathrm{E}(K_2, \mathrm{E}(K_1, P))$
- Say $X = \mathrm{E}(K_1, P) = \mathrm{D}(K_2, C)$
- Attacker knows two plaintext, ciphertext pairs (P_a, C_a) and (P_b, C_b)
  1. Encrypt P_a using all 2^k values of K_1 to get multiple values of X
  2. Store results in table and sort by X
  3. Decrypt C_a using all 2^k values of K_2
  4. As each decryption result produced, check against table
  5. If match, check current K_1, K_2 on C_b. If P_b obtained, then accept the keys
- With two known plaintext, ciphertext pairs, probability of successful attack is almost 1
- Encrypt/decrypt operations required: $\approx 2 \times 2^k$ (twice as many as single encryption)

Example 5-bit Block Cipher

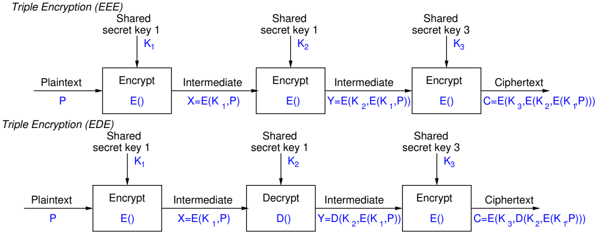
	Ciphertext for key, K:							
P	*000*	*001*	*010*	*011*	*100*	*101*	*110*	*111*
00000	00001	10010	01101	01111	11011	10011	10000	11101
00001	10001	01001	11010	10000	01010	11100	10100	01010
00010	01011	10100	11011	01100	00100	10100	00111	00100
00011	01110	10110	01011	00111	10110	11101	11000	00101
00100	00011	00011	00001	11101	11001	10010	11011	01100
00101	10100	10111	01110	00010	01101	00011	01101	00110
00110	10101	11111	00110	10011	00010	10001	10111	10110
00111	01101	10001	10111	00110	11111	01100	11100	10011
01000	01000	11011	10011	01010	01001	10110	10011	11111
01001	10010	11110	10001	10101	01111	00100	00000	01110
01010	01111	00010	10000	10110	11000	01010	00001	00010
01011	11110	01110	00111	01011	11101	11011	01111	10010
01100	11011	10000	01010	00101	01100	00101	01100	00111
01101	11101	00111	10110	01000	01000	10111	10010	11100
01110	11000	01000	10100	00000	11010	01111	11111	01000
01111	01001	11101	01100	00001	00011	01000	01010	01101
10000	00110	11100	01111	01001	01011	11111	00010	11011
10001	11111	01100	10010	10010	00000	11010	11110	00000
10010	10110	10011	11110	01101	10111	01101	10001	10000
10011	00010	00001	11000	11100	10100	00111	00011	10111
10100	10111	01101	11001	11111	10011	00000	00100	00011
10101	01010	01111	00101	00011	00001	01001	10101	01011
10110	00000	00110	10101	11010	00110	01011	01000	11001
10111	00111	11000	01001	11110	10000	00010	01110	10100
11000	00101	01011	00010	10001	11100	10000	11010	10001
11001	11100	00000	11101	10111	10001	01110	00101	11000
11010	11010	11001	01000	01110	01110	11110	01011	01001
11011	01100	11010	11111	11001	10101	00001	10110	00001
11100	11001	01010	00100	00100	00101	11001	00110	10101
11101	10011	10101	00011	10100	00111	00110	11001	01111
11110	00100	00101	11100	11000	10010	11000	11101	11110
11111	10000	00100	00000	11011	11110	10101	01001	11010

Meet-in-the-Middle Attack (exercise)

The figure on slide 54 shows an example 5-bit block cipher, referred to as *Bob's Cipher*. A double version of Bob's cipher, called *Double-Bob*, was used by two users to exchange multiple encrypted messages using the same 6-bit secret key. You have obtained the plaintext/ciphertext pairs of two of those messages: $(P_1, C_1) = (01101, 11111)$ and $(P_2, C_2) = (11001, 11011)$. Using a meet-in-the-middle attack, find the secret key.

Triple Encryption Concept

- Different variations:
  - Use 2 keys, e.g. Triple-DES 112 bits
  - Use 3 keys, e.g. Triple-DES 168 bits



- Why E-D-E? To be compatible with single DES:

$$C = \mathrm{E}(K_1, \mathrm{D}(K_1, \mathrm{E}(K_1, P))) = \mathrm{E}(K_1, P)$$

- Problem: 3 times slower than single DES

Contents

Cryptanalysis of Triple-DES and AES

Cipher	Method	Key space	Required resources: Time	Memory	Known data
DES	Brute force	2^{56}	2^{56}	-	-
3DES	MITM	2^{168}	2^{111}	2^{56}	2^{2}
3DES	Lucks	2^{168}	2^{113}	2^{88}	2^{32}
AES 128	Biclique	2^{128}	$2^{126.1}$	2^{8}	2^{88}
AES 256	Biclique	2^{256}	$2^{254.4}$	2^{8}	2^{40}

- Known data: chosen pairs of (plaintext, ciphertext)
- Lucks: S. Lucks, Attacking Triple Encryption, in *Fast Software Encryption*, Springer, 1998
- Biclique: Bogdanov, Khovratovich and Rechberger, Biclique Cryptanalysis of the Full AES, in *ASIACRYPT2011*, Springer, 2011