CRYPTOGRAPHY

Doç. Dr. Sıddıka Berna Örs Yalçın

Room Number: 2318

email: siddika.ors@itu.edu.tr

web page: http://web.itu.edu.tr/∼orssi/

References

1. Douglas R. Stinson, Cryptography Theory and Practice, Third Edition, CRC Press, November 2005.

2. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of Applied Cryptography, CRC Press, ISBN: 0-8493-8523-7, October 1996, 816 pages.

Content

1. Classical cryptography: introduction: some simple cryptosystems
2. Cryptanalysis of simple cryptosystems
3. Shannon's theory: probability theory, entropy, properties of entropy
4. Product cryptosystems
5. Block ciphers: substiturion-permutation network
6. Linear cryptanalysis
7. Differential cryptanalysis
8. The data encryption standard (DES)
9. Advanced encryption standard (AES), modes of operation
10. Hash functions: collision-free hash functions, authentication codes
11. The RSA system and factoring: introduction to public-key cryptography
12. Public-key cryptosystems based on discrete logarithm problem: the ElGamal cryptosystem
13. Finite field and elliptic curve systems
14. Signature schemes: introduction, the ElGamal signature scheme
15. The digital signature algorithm (DSA), the elliptic curve digital signature algorithm (ECDSA)

Grading

1st Homework	2-5th week	15 %
1st Midterm	6th week	15 %
2nd Homework	7-10th week	15 %
2nd Midterm	11th week	15 %
3rd Homework	12th week-final exam	15 %
Final		40 %

History of Cryptography

hieroglyphs - around 2000 B.C.

漢 汉
字 字

ideogram - ancient Chinese

Clay tablets from Mesopotamia

ABCDEFGHIJKLMNOPQRSTUVWXYZ
ZYXWVUTSRQPONMLKJIHGFEDCBA

Atbash cipher - around 500 to 600 BC

Scytale - Spartan

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I	J
3	K	L	M	N	O
4	P	Q	R	S	T
5	U	V	W	X	Y/Z

Polybius Square - Greek method

Steganography

- physically concealed beneath wax on wooden tablets

- a tattoo on a slave's head concealed by regrown hair

Usage of Cryptography

- Past
  - Military, Diplomatic Service, Government
  - was used as a tool to protect national secrets and strategies
- Now
  - Private sector
  - is used to protect information in digital form and to provide security services

CRYPTOGRAPHY

- is the study of mathematical techniques related to aspects of information security such as
  - confidentiality,
  - data integrity,
  - entity authentication,
  - data origin authentication.

- is about the prevention and detection of cheating and other malicious activities.

Basic Terminology: Domains

The cryptosystem is a five- tuple $\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D}$

- $\mathcal{P}$: a set called the plaintext space.
- $\mathcal{C}$: a set called the ciphertext space.
- $\mathcal{K}$: a set called the key space.
- For each $K \in \mathcal{K}$, there is an encryption rule $e_K \in \mathcal{E}$ and a corresponding decryption rule $d_K \in \mathcal{D}$. Each $e_K : \mathcal{P} \to \mathcal{C}$ and $d_K : \mathcal{C} \to \mathcal{P}$ are functions such that $d_K(e_K(x)) = x$ for every element $x \in \mathcal{P}$.

Shift Cipher

- Let $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}$. For $0 \leq K \leq 25$, define $e_K(x) = (x + K)$ mod 26 and $d_K(y) = (y - K)$ mod 26

- Since there are only 26 possible keys, it is easy to try every possible K until a meaningful plaintext is obtained.

- $K = 3 \Longrightarrow$ is called Ceaser cipher (~55 BC)

Caesar Cipher

ABCDEFGHIJKLMNOPQRSTUVWXYZ
CDEFGHIJKLMNOPQRSTUVWXYZAB

Example 1 :

$x =$	THISC	IPHER	ISCER	TAINL	YNOTS	ECURE
$y = e_K(x) =$	WKLVF	LSKHU	LVFHU	WDLQO	BQRWV	HFXUH

Cryptanalysis

The practice of changing ciphertext into plaintext without complete knowledge of the cipher.

First method : Frequency analysis - Arabic author, Qalqashandi

If a cryptosystem is to be of practical use:

1. Each e_K and d_K should be efficiently computable.

2. An opponent, upon seeing a ciphertext string y should be unable to determine the key K or the plaintext string x.

Types of Attacks (1/2)

Kerckhof's principle:the attacker has full knowledge of the encryption algorithm, and only the key of the cryptosystem is unknown.

The aim of the attacker is to read the encrypted messages, which in many cases is achieved by finding the secret key of the system.

The efficiency of the attack is measured by

- the amount of plaintext- ciphertext pairs required,
- time spent for their analysis
- the success probability of the attack

Usually the starting point of a cryptanalytic attack is the ability, to distinguish the output of a cipher from the output of a random permutation.

Types of Attacks (2/2)

- Ciphertext- Only
- Known Plaintext
- Chosen Plaintext
- Chosen Ciphertext
- Adaptive Chosen Plaintext or Ciphertext
- Related Key
- Partial Knowledge of the Key

The Goals of Cryptanalytic Attacks (1/2)

- Distinguishing Attacks
- Partial Knowledge of the Plaintext
- Decryption
- Encryption (Forgery)
- Partial Key Recovery
- Total Key Recovery

The Goals of Cryptanalytic Attacks (2/2)

The cipher can be considered broken if:

- its output can be distinguished from a random permutation
- the secret key is found
- it is possible to derive secret elements of a cipher

A cipher is broken, if a person who uses it decides to stop doing so because he/she does not trust its security anymore.

People expect that a good cipher is a one for which the best attack is an exhaustive search for the key.

A cipher is considered broken if a weakness in it is found which requires the changes of the design.

Cryptology

- Exhaustive search: trying all possible keys
- Cryptanalysis: the study of mathematical techniques to break the system
- Cryptology: cryptography + cryptanalysis
- Cryptosystem: a set of cryptographic primitives, symmetric key and public key

Cryptanalysis of Example 1

WKLVF	LSKHU	LVFHU	WDLQO	BQRWV	HFXUH
VJKUE	KQJGT	KUEGT	VCKPN	APQVU	GEWTG
UIJTD	JPIFS	JTDFS	UBJOM	ZOPUT	FDVSF
THISC	IPHER	ISCER	TAINL	YNOTS	ECURE

$K = 3$

Mono- alphabetic Substitution Cipher

- Let $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$. $\mathcal{K}$ consists of all possible permutations of the 26 symbols. For each permutation $\pi \in \mathcal{K}$, define

  $e_\pi(x) = \pi(x)$ and $d_\pi(y) = \pi^{-1}(y)$ where π^{-1} is the inverse permutation to π.

- If the alphabet is the English alphabet, then the size of the key space is $26! \approx 4 \times 10^{26}$

- The distribution of letter frequencies is preserved in the ciphertext.

Example

$$\pi = \begin{array}{l} \text{A B C D E F G H I J K L M N O P Q R S T U V W X Y Z} \\ \text{B D F H J L N P R T V X Z B E G I K M O Q S U W A Y} \end{array}$$

$m =$	THISC	IPHER	ISCER	TAINL	YNOTS	ECURE
$c = e_\pi(m) =$	OPRMF	RGPJK	RMFJK	OBRBX	ABEOM	JFQKJ

Cryptanalysis of Mono- alphabetic Substitution Cipher (1/8)

Ciphertext:

VGQBQHWHUXYQVULRZUGVWBUYVHKYZTHXQBNOZYBVYVURVEQBO
YQHYVTMXTZRQHULVULYQBZOWBOZGYQBKBYYBOTZGYQBHKEQHM
BYYQHYUZYHNZOWRZDKWMBPHWBZDYVGHUXZUBNVTQBTYZWBRV
EQBOYQBTBHUWLBYHYYQBVOPBHUVULQBTQZDKWTDMTYVYDYBY
QBGZDOYQKBYYBOZGYQBHKEQHMBYUHPBKXGZOHUWTZYQBZYQBO

Letter Frequency in the English Language

E T A O I N S R H L D C U M F P G W Y B V K X J Q Z

Letter Frequency in the ciphertext

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	13	0	2	1	0	3	7	0	0	3	2	2	1	4	1	9	2	0	4	6	6	4	2	13	7

e_π(E) = B or Y and e_π(T) = B or Y

Cryptanalysis of Mono- alphabetic Substitution Cipher (2/8)

Digraphs in the ciphertext with B

- - QBQ- - - - - - - - - - - - - - - - WBU- - - - - - - - - QBN- - YBV- - -
- - - QBO- - - - - - - - - - - - - - - - - - - QBZ- WBO- - - QBKBYYBO
- - - - QBH- - - - MBY- - - - - - - - - - - - - - - - - MBP- WBZ- - - - -
- - - UBN- - QBT- - WBR- - QBO- QBTBH- - LBY- - - QBV- PBH-
- - - QBT- - - - - - - - - - - - - - YBYQBG- - - - - KBYYBO- - - QBH-
- - - MBY- - PBK- - - - - - - - - - QBZ- QBO-

The Digraph Frequencies in the English Language
th he an in er on re ed nd ha at en es of nt ea ti to io le is ou ar as de rt ve

Digraph Frequency in the ciphertext with B

BG	BH	BK	BN	BO	BP	BR	BQ	BT	BU	BV	BY	BZ	KB	LB	MB	PB	QB
1	4	1	2	6	1	1	1	3	1	2	6	3	2	1	3	2	15

TB	UB	WB	YB
1	1	4	4

e_π(HE) = QB $\Rightarrow$ e_π(H) = Q e_π(ER) = BO or BY $\Rightarrow$ e_π(R) = O or Y

Cryptanalysis of Mono- alphabetic Substitution Cipher (3/8)

The Trigraph Frequencies in the English Language

the and tha ent ion tio for nde has nce tis oft men

Trigraphs in the ciphertext such as xQB and Byz

- GQB- - - - - - - - - - - - - - - - - - BUY- - - - - - - XQBNO- - BVY- -
- - EQBOY- - - - - - - - - - - - - - - - - - YQBZO- - - - - YQB- - - - BO
T- - YQBHK- BPH- BZD- - - -
- - - - BNVTQBTY- - BRV- - - - YQB- BHU- - BYH- YQBVO- BHU
- - LQB- - - - - - - - - - - - - - - - BYQBGZ- - - - - - - - BOZ- YQBHK
- - - - BYU- - BKX- - - - - - - - YQB- YQBOT

The Trigraph Frequencies in the ciphertext such as xQB and Byz

BGZ	BHK	BHU	BKX	BNO	BNV	BOT	BOY	BOZ	BPH	BRV	BTY
1	2	1	1	1	1	2	1	1	1	1	1

$e_\pi(\text{THE}) = \text{YQB} \Rightarrow e_\pi(\text{T}) = \text{Y} \Rightarrow e_\pi(\text{R}) = \text{O}$

$e_\pi(\text{ENT}) = \text{BTY or BUY or BVY} \Rightarrow e_\pi(\text{N}) = \text{T or U or V}$

Cryptanalysis of Mono- alphabetic Substitution Cipher (4/8)

Digraphs in the ciphertext with Y

- - - - - - - - - - XYQ- - - - - - - - - - - UYV- KYZ- - - - - - - ZYBVYV-
- - - - - OYQHYV- - - - - - - - - - - - - LYQ- - - - - - - GYQ- - BYYB-
- - GYQ- - - - - - - BYYQHYUZYH- - - - - - - - - - - - - - - - - DYV- -
- - - - - - - - - - TYZ- - - - - - - OYQ- - - - - - - BYHYYQ- - - - - - -
- - - - - - - - - - - - - - - TYVYDYBYQ- - - - OYQ- BYYB- - GYQ- - -
- - - - BYU- - - - - - - - - - - - - ZYQ- ZYQ- - -

Digraph Frequency in the ciphertext with Y

BY	DY	HY	GY	KY	OY	TY	UY	VY	XY	YB	YD	YH	YQ	YU	YV	YZ	ZY
4	2	2	1	1	3	2	1	2	1	3	1	1	9	2	4	2	4

e_π(TH) = YQ e_π(ET) = BY e_π(TI or TO) = YV

Cryptanalysis of Mono- alphabetic Substitution Cipher (5/8)

Trigraphs in the ciphertext such as Yxy

- - - - - - - - - - - YQV- - - - - - - - - - - YVH- - - - - - - - - - - YBVYVU
- - - - - - YQHYVT- -
- - - - - - - - - - - - - - - YQHYUZYHN- - - - - - - - - - - - - - - - YVG-
- - - - - - - - - - - YZW- - - - - - - - - - - - - - - - - - YHY- - - - - - - - -
- - - - - - - - - - - - - - - - YVYDY- - - - - - - - YQK- - - - - - - - - - - -
- - - - - YUH- -

The Trigraph Frequencies in the ciphertext such as YQx and Yxy

YBV	YDY	YHN	YHY	YQV	YQH	YQK	YUH	YUZ	YVG	YVH	YVT
1	1	1	1	1	2	1	1	1	1	1	1

$e_\pi(\text{THA}) = \text{YQH} \Rightarrow e_\pi(\text{A}) = \text{H}$

$e_\pi(\text{TIO or TIS}) = \text{YVG or YVT or YVU} \Rightarrow e_\pi(\text{I}) = \text{V}$

Cryptanalysis of Mono- alphabetic Substitution Cipher (6/8)

Digraphs in the ciphertext with H

- - - - - HWHU- - - - - - - - - - - - - - - - VHK- - THX- - - - - - - - - -
- - - - - - - - - HY- - - - - - - - HU- - - - - - - - - - - - - - - - - - - -
- - - - - - HK- - HM- - - - - - - - - HN- - - - - - - - - - PHW- - - - - GH
U- HU- - - - HY- - - - - - - HU
- HK
- - HM- - UHP- - - - - - HU

Digraph Frequency in the ciphertext with H

GH	HK	HM	HN	HP	HU	HW	HX	HY	PH	TH	UH	WH	VH
1	3	2	1	1	5	2	1	2	1	1	1	1	1

$e_\pi(\mathsf{AN}) = \mathsf{HU} \Rightarrow e_\pi(\mathsf{N}) = \mathsf{U}$

The Trigraph Frequencies in the ciphertext such as HUx

HUL	HUW	HUV	HUX
1	2	1	2

$e_\pi(\mathsf{AND}) = \mathsf{HUW}$ or $\mathsf{HUX} \Rightarrow e_\pi(\mathsf{D}) = \mathsf{W}$ or X

Cryptanalysis of Mono- alphabetic Substitution Cipher (7/8)

B $\rightarrow$ E, Y $\rightarrow$ T, Q $\rightarrow$ H, O $\rightarrow$ R, H $\rightarrow$ A, V $\rightarrow$ I, W $\rightarrow$ D

I- HEHADA- - THI- - - - - - IDE- TIA- T- - A- HE- R- TEITI-
VGQBQHWHUXYQVULRZUGVWBUYVHKYZTHXQBNOZYBVYVU
- I- HERTHATI- - - - - - HA- - I- - THE- RDER- - THE- ETTER
RVEQBOYQHYVTMXTZRQHULVULYQBZOWBOZGYQBKBYYBO
- - - THEA- - HA- ETTHAT- - TA- - RD- - - - D- E- ADE- - TI- A
TZGYQBHKEQHMBYYQHYUZYHNZOWRZDKWMBPHWBZDYVGH
- - - - E- I- HE- T- DE- I- HERTHE- EA- D- ETATTHEIR- EA-
UXZUBNVTQBTYZWBRVEQBOYQBTBHUWLBYHYYQBVOPBHU
I- - HE- H- - - D- - - - TIT- TETHE- - - RTH- ETTER- - THEA-
VULQBTQZDKWTDMTYVYDYBYQBGZDOYQKBYYBOZGYQBHK
- HA- ET- A- E- - - - RA- D- - THE- THER-

Cryptanalysis of Mono- alphabetic Substitution Cipher (8/8)

K $\rightarrow$ L, Z $\rightarrow$ O, G $\rightarrow$ F, D $\rightarrow$ U, T $\rightarrow$ S, M $\rightarrow$ B, E $\rightarrow$ P, R $\rightarrow$ C, U $\rightarrow$ N, X $\rightarrow$ Y, L $\rightarrow$ G, N $\rightarrow$ W, R $\rightarrow$ C, P $\rightarrow$ M

IFHEHADANYTHINGCONFIDENTIALTOSAYHEWROTEITIN
VGQBQHWHUXYQVULRZUGVWBUYVHKYZTHXQBNOZYBVYVU
CIPHERTHATISBYSOCHANGINGTHEORDEROFTHELETTER
RVEQBOYQHYVTMXTZRQHULVULYQBZOWBOZGYQBKBYYBO
SOFTHEALPHABETTHATNOTAWORDCOULDBEMADEOUTIFA
TZGYQBHKEQHMBYYQHYUZYHNZOWRZDKWMBPHWBZDYVGH
NYONEWISHESTODECIPHERTHESEANDGETATTHEIRMEAN
UXZUBNVTQBTYZWBRVEQBOYQBTBHUWLBYHYYQBVOPBHU
INGHESHOULDSUBSTITUTETHEFOURTHLETTEROFTHEAL
VULQBTQZDKWTDMTYVYDYBYQBGZDOYQKBYYBOZGYQBHK
PHABETNAMELYFORANDSOTHEOTHERS

IF HE HAD ANYTHING CONFIDENTIAL TO SAY HE WROTE IT IN CIPHER THAT IS BY SO CHANGING THE ORDER OF THE LETTERS OF THE ALPHABET THAT NOT A WORD COULD BE MADE OUT IF ANYONE WISHES TO DECIPHER THESE AND GET AT THEIR MEANING HE SHOULD SUBSTITUTE THE FOURTH LETTER OF THE ALPHABET NAMELY FOR AND SO THE OTHERS

Affine Cipher (1/2)

Let $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$ and let

$\mathcal{K} = \{(a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26} : gcd(a, 26) = 1\}$.

For $K = (a, b) \in \mathcal{K}$, define

$e_K(x) = y \equiv (ax + b) \bmod 26$ and $d_K(y) = x \equiv a^{-1}(y - b) \bmod 26$.

In order that decryption is possible, for any $y \in \mathbb{Z}_{26}$, $y \equiv (ax + b) \bmod 26$ must have a unique solution for $x \iff gcd(a, 26) = 1$

Affine Cipher (2/2)

$y \equiv (ax + b) \bmod 26$ is equivalent to $y - b \bmod 26 \equiv ax \bmod 26$

As $y \in \mathbb{Z}_{26}$, $y - b \bmod 26 \in \mathbb{Z}_{26}$.

It suffices to study the congruence $y \equiv ax \bmod 26$.

If $gcd(a, 26) = d > 1$, then $0 \equiv ax \bmod 26$ has to distinct solutions in $\mathbb{Z}_{26}$, namely $x = 0$ and $x = \frac{26}{d}$.

In this case $e(x) = ax + b \bmod 26$ is not an injective function and hence not a valid encryption function.

Since $26 = 2 \times 13$, $a = 1, 3, 4, 7, 9, 11, 15, 17, 19, 21, 23, 25$, b can be any element in $\mathbb{Z}_{26}$. Hence affine cipher has $12 \times 26 = 312$ possible keys.

Multiplicative Inverse

$a \in \mathbb{Z}_n$, *multiplicative inverse* of a mod n, denoted a^{-1} mod n. $aa^{-1} \equiv 1 \bmod n$.

If p is prime, then every non- zero element of $\mathbb{Z}_p$ has a unique multiplicative inverse.

Algorithm 1 Multiplicative Inverse

Require: $a \in \mathbb{Z}_n$, n is a positive integer

Ensure: a^{-1} mod n

1: Use the extended Euclidean algorithm to find integers x and y such that $ax + ny = d$, where $d = gcd(a, n)$.

2: If $d > 1$, then a^{-1} mod n does not exist. Otherwise return x .

Extended Euclidean (1/2)

Algorithm 2 Extended Euclidean

Require: two non- negative integers a and b with $a \geq b$.

Ensure: $d = gcd(a, b)$ and integers x, y satisfying $ax + by = d$.

1: If $b = 0$ then set $d \leftarrow a$, $x \leftarrow 1$, $y \leftarrow 0$ and return (d, x, y) .

2: Set $x_2 \leftarrow 1$, $x_1 \leftarrow 0$, $y_2 \leftarrow 0$, $y_1 \leftarrow 1$.

3: **while** $b > 0$ **do**

4: $q \leftarrow \lfloor \frac{a}{b} \rfloor$, $r \leftarrow a - qb$, $x \leftarrow x_2 - qx_1$, $y \leftarrow y_2 - qy_1$.

5: $a \leftarrow b$, $b \leftarrow r$, $x_2 \leftarrow x_1$, $x_1 \leftarrow x$, $y_2 \leftarrow y_1$ and $y_1 \leftarrow y$.

6: **end while**

7: Set $d \leftarrow a$, $x \leftarrow x_2$, $y \leftarrow y_2$ and return (d, x, y) .

Extended Euclidean (2/2)

Example $gcd(81, 57) = ?$

$$
\begin{aligned}
81 &= 57 \cdot 1 + 24 \\
57 &= 24 \cdot 2 + 9 \\
24 &= 9 \cdot 2 + 6 \\
9 &= 6 \cdot 1 + 3 \\
6 &= 3 \cdot 2
\end{aligned}
$$

$81x + 57y = 3$

3=9- 6

3=9- (24- 9 · 2) · 1=9 · 3- 24

3=(57- 24 · 2) · 3- 24=57 · 3- 24 · 7

3=57 · 3- (81- 57) · 7=81 · - 7+57 · 10

Cryptanalysis of the Affine Cipher (1/3)

Ciphertext:

KADHLFMLNMFKVERSLDYAREHFSOORLDYAREHKRWKNHDS

XFSFUUDSRLDYAREDSTADLAMRKKREHFERRSLVORONHDS

XKARUVEPNMFTARERFSOFERTAVMRSNPIREHIRKTRRSFS

OFSODHERMFKDQRMZYEDPR

Letter Frequency in the English Language

E T A O I N S R H L D C U M F P G W Y B V K X J Q Z

Letter Frequency in the ciphertext

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
6	0	0	8	8	8	0	5	1	0	6	4	4	3	4	2	0	16	8	2	2	2	0	1	2	0

Cryptanalysis of the Affine Cipher (2/3)

$E_e(\mathsf{E}) = \mathsf{R} \rightarrow 17 \equiv a4 + b \bmod 26$

$E_e(\mathsf{T}) = \mathsf{D} \rightarrow 3 \equiv a19 + b \bmod 26$ then $a = 6$, $gcd(6, 26) = 2 > 1$.

$E_e(\mathsf{T}) = \mathsf{E} \rightarrow 4 \equiv a19 + b \bmod 26$ then $a = 13$, $gcd(13, 26) = 13 > 1$.

$E_e(\mathsf{T}) = \mathsf{F} \rightarrow 5 \equiv a19 + b \bmod 26$ then $a = 20$, $gcd(20, 26) = 2 > 1$.

$E_e(\mathsf{T}) = \mathsf{S} \rightarrow 18 \equiv a19 + b \bmod 26$ then $a = 7$, $gcd(7, 26) = 1$ and $b = 9$.

Decrypted message

PVOWESTEITSPYDQFEORVQDWSFXXQEORVQDWPQNPIWO

FCSFSJJOFQEORVQDOFUVOEVTQPPQDWSDQQFEYXQXIW

OFCPVQJYDMITSUVQDQSFXSDQUVYTQFIMLQDWLQPUQQ

FSFXSFXOWDQTSPOBQTGRDOMQ

Cryptanalysis of the Affine Cipher (3/3)

$E_e(\mathsf{T}) = \mathsf{K} \rightarrow 10 \equiv a19 + b \bmod 26$ then $a = 3$, $gcd(7, 26) = 1$ and $b = 5$.

Decrypted message

THISCALCULATORENCIPHERSANDDECIPHERSTEXTUSI

NGANAFFINECIPHERINWHICHLETTERSAREENCODEDUS

INGTHEFORMULAWHEREANDAREWHOLENUMBERSBETWEE

NANDANDISRELATIVELYPRIME

THIS CALCULATOR ENCIPHERS AND DECIPHERS TEXT USING AN AFFINE CIPHER IN WHICH LETTERS ARE ENCODED USING THE FORMULA WHERE AND ARE WHOLE NUMBERS BETWEEN AND AND IS RELATIVELY PRIME

Alberti Cipher

All of the Western European governments used cryptography

Venice created an elaborate organization in 1452.

Leon Battista Alberti was known as "The Father of Western Cryptology" in part because of his development of polyalphabetic substitution.

Formula

The larger one is called Stabilis [stationary or fixed], the smaller one is called Mobilis [movable]

Polyalphabetic substitution is any technique which allows different ciphertext symbols to represent the same plaintext symbol.

Vigenère Cipher

$m \geq 0$ and $m \in \mathbb{Z}$.
Let $\mathcal{P} = \mathcal{C} = \mathcal{K} = (\mathbb{Z}_{26})^m$. For a key $K = (k_1, k_2, \ldots, k_m))$, we define
$e_K = (x_1, x_2, \ldots, x_m) = (x_1 + k_1, x_2 + k_2, \ldots, x_m + k_m)$
$d_K = (y_1, y_2, \ldots, y_m) = (y_1 - k_1, y_2 - k_2, \ldots, y_m - k_m)$

Example: $m = 6$. The keyword is *CIPHER*. $K = (2, 8, 15, 7, 4, 17)$.

message: namedafterblaisedevigenere

13	0	12	4	3	0	5	19	4	17	1	11	0	8	18	4	3	4	21	8
2	8	15	7	4	17	2	8	15	7	4	17	2	8	15	7	4	17	2	8
15	8	1	11	7	17	7	1	19	24	5	2	2	16	7	11	7	21	23	16

ciphertext: piblhrhbtyfccqhlhvxqvlrvtm

The number of possible keywords of length m is 26^m.

An alphabetic character can be mapped to one of m possible alphabetic characters.

Cryptanalysis of Vigenère Cipher

The first step is to determine the keyword length, m.

- Kasiski test : 1854 - Charles Babbage and 1863 - Friedrich Kasiski
- the index of coincidence

Kasiski Test

Two identical segments of plaintext will be encrypted to the same ciphertext whenever their occurrence in the plaintext is Δ positions apart, where $\Delta \equiv 0 \bmod m$.

- Search the ciphertext for pairs of identical segments of length at least three.
- Record the distance between the starting positions of the two segments.
- If we obtain several such distances, say $\Delta_1, \Delta_2, \ldots$ we would conjecture that m divides all of the Δ_i's, $m \mid gcd\,(\Delta_1, \Delta_2, \ldots,)$

The reason this test works is that if a repeated string occurs in the plaintext, and the distance between them is a multiple of the keyword length, m, the keyword letters will line up in the same way with both occurrences of the string.

Example for Kasiski Test (1/2)

ciphertext:

knwllrficfxykvvjsehqslrkkiaasjcgwlaiqbtpxzpkxwlvtbwhxzujnzstjicnmeibwlsifmgvj

kjmalxkgzhvjkjmpstycjtaxycbcvxrywgkgfwtsiidcltvykknpucfpmlpwygaivhveqeoiiv

ptzirplvlxrvbwlmiompumeiptzlfwtszysubxaykgbwljfwziopvvbtyswvpthpgjiqlxecut
scwqzpuhjqbwlskjmgz

Example for Kasiski Test (2/2)

ciphertext string	occurs at (index)	spacing	factors
MEI	64 172	108	2 3 4 6 9 12 18 27 36 54 108
BWL	67 163	96	2 3 4 6 8 12 16 24 32 48 96
BWL	67 193	126	2 3 6 7 9 14 18 21 42 63 126
BWLS	67 235	168	2 3 4 6 7 8 12 14 21 24 28 42 56 84 168
WLS	68 236	168	2 3 4 6 7 8 12 14 21 24 28 42 56 84 168
VJKJM	75 87	12	2 3 4 6 12
JKJM	76 88	12	2 3 4 6 12
KJM	77 89	12	2 3 4 6 12
KJM	77 239	162	2 3 6 9 18 27 54 81 162
KJM	89 239	150	2 3 5 6 10 15 25 30 50 75 150
FWTS	113 179	66	2 3 6 11 22 33 66
WTS	114 180	66	2 3 6 11 22 33 66
VPT	150 210	60	2 3 4 5 6 10 12 15 20 30 60
PTZ	151 175	24	2 3 4 6 8 12 24
BWL	163 193	30	2 3 5 6 10 15 30
BWL	163 235	72	2 3 4 6 8 9 12 18 24 36 72
BWL	193 235	42	2 3 6 7 14 21 42

Keyword length $m = 6$.

Index of Coincidence (1/3)

1920 - Friedman

Definition: Suppose $x = x_1 x_2 \ldots x_n$ is a string of n alphabetic characters. The *index of coincidence* of x, denoted by $I_c(x)$, is defined to be the probability that two random elements of x are identical.

Suppose the frequencies of A, B, C, ..., Z in x are $f_0, f_1, \ldots, f_{25}$. We can choose two elements of x in $\binom{n}{2} = n(n-1)$ ways. For each i, $0 \leq i \leq 25$ there are $\binom{f_i}{2} = f_i\,(f_i - 1)$ ways of choosing both elements to be i.

$$I_c(x) = \frac{\sum_{i=0}^{25} \binom{f_i}{2}}{\binom{n}{2}} = \frac{\sum_{i=0}^{25} f_i\,(f_i-1)}{n(n-1)} = \frac{\sum_{i=0}^{25} f_i^2 - \sum_{i=0}^{25} f_i}{n(n-1)} = \frac{\sum_{i=0}^{25} f_i^2 - n}{n(n-1)}$$

$n \to \infty \Rightarrow I_c(x) \to \frac{\sum_{i=0}^{25} f_i^2}{n^2}$, $p_i = \frac{f_i}{n}$ then $I_c(x) \to \sum_{i=0}^{25} p_i^2$

Index of Coincidence (2/3)

letter	probability	letter	probability	letter	probability
A	0.0856	B	0.0139	C	0.0279
D	0.0378	E	0.1304	F	0.0289
G	0.0199	H	0.0528	I	0.0627
J	0.0013	K	0.0042	L	0.0339
M	0.0249	N	0.0707	O	0.0797
P	0.0199	Q	0.0012	R	0.0677
S	0.0607	T	0.1045	U	0.0249
V	0.0092	W	0.0149	X	0.0017
Y	0.0199	Z	0.0008		

$n \to \infty \Rightarrow I_c(x) \approx \sum_{i=0}^{25} p_i^2 = 0.065$

The same reasoning applies if x is a ciphertext string obtained using any monoalphabetic cipher.

Index of Coincidence (3/3)

$y = y_1 y_2 \dots y_n$ constructed by Vigenère Cipher.

m substrings of y, $\bar{y_1}$, $\bar{y_2}$, ..., $\bar{y_m}$ by writing out the ciphertext in columns in a rectangular array of dimensions $m \times (n/m)$.

Example: $n = 15$ and $m = 3$

$$\begin{array}{ccccc} y_1 & y_4 & y_7 & y_{10} & y_{13} \\ y_2 & y_5 & y_8 & y_{11} & y_{14} \\ y_3 & y_6 & y_9 & y_{12} & y_{15} \end{array} \Rightarrow \begin{array}{rcl} \bar{y_1} & = & y_1 y_4 y_7 y_{10} y_{13} \\ \bar{y_2} & = & y_2 y_5 y_8 y_{11} y_{14} \\ \bar{y_3} & = & y_3 y_6 y_9 y_{12} y_{15} \end{array}$$

If $I_c(\bar{y_i}) \approx 0.065$ then m is the keyword length.

If m is not the keyword length then $\bar{y_i}$ s are random

$\sum_{i=0}^{25} f_i = n$, in random text $f_0 = f_1 = \cdots = f_{25}$, $26 f_i = n$, $f_i = \frac{n}{26}$,

$I_c(x) = \frac{26 f_i^2}{n^2} = \frac{1}{26} = 0.038$. The two values 0.065 and 0.038 are sufficiently far apart that we will often be able to determine the correct keyword length.

Permutation Cipher

Alter the plaintext characters positions by rearranging them using a permutation.

Let m be a positive integer. Let $\mathcal{P} = \mathcal{C} = (\mathbb{Z}_{26})^m$ and let $\mathcal{K}$ consist of all permutations of $1, \ldots, m$. For a key π, we define

$$e_\pi(x_1, \ldots, x_m) = \left(x_{\pi(1)}, \ldots, x_{\pi(m)}\right)$$

and

$$d_\pi(y_1, \ldots, y_m) = \left(y_{\pi^{-1}(1)}, \ldots, y_{\pi^{-1}(m)}\right)$$

where π^{-1} is the inverse permutation to π.

Example for Permutation Cipher

$m = 6$

Encryption: $\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 4 & 3 & 1 & 6 & 2 & 5 \end{pmatrix}$

plaintext: he walked up and down the passage two or three times

plaintext is divided into groups of 6:

hewalk edupan ddownt hepass agetwo orthre etimes

ciphertext: WLEHKAUADENPONDDTWPSEHSAEWGAOTTRROEHIETESM

Decryption: $\pi^{-1} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 5 & 2 & 1 & 6 & 4 \end{pmatrix}$

Remarks for Permutation Cipher

The Permutation Cipher is not monoalphabetic.

In the example the first e is encrypted as L, the second e is encrypted as U and the third e is encrypted as S.

This encryption does not change the frequency of alphabetic characters but the positions of the letters.

The different number of keys are $m!$.

Product Cryptosystems 1/2

introduced by Shannon in 1949

For simplicity; $\mathcal{C} = \mathcal{P}$: *endomorphic cryptosystem*

Suppose $S_1 = (\mathcal{P}, \mathcal{P}, \mathcal{K}_1, \varepsilon_1, \mathcal{D}_1)$ and $S_2 = (\mathcal{P}, \mathcal{P}, \mathcal{K}_2, \varepsilon_2, \mathcal{D}_2)$ are two endomorphic cryptosystems.

Product cryptosystem of S_1 and S_2=$S_1 \times S_2 = (\mathcal{P}, \mathcal{P}, \mathcal{K}_1 \times \mathcal{K}_2, \varepsilon, \mathcal{D})$.

A key of the product cryptosystem: $K = (K_1, K_2)$, where $K_1 \in \mathcal{K}_1$ and $K_2 \in \mathcal{K}_2$.

$e_{(K_1,K_2)}(x) = e_{K_2}\left(e_{K_1}(x)\right)$ and $d_{(K_1,K_2)}(y) = d_{K_1}\left(d_{K_2}(y)\right)$.

Product Cryptosystems 2/2

$$\begin{aligned} d_{(K_1,K_2)}\left(e_{(K_1,K_2)}(x)\right) &= d_{(K_1,K_2)}\left(e_{K_2}\left(e_{K_1}(x)\right)\right) \\ &= d_{K_1}\left(d_{K_2}\left(e_{K_2}\left(e_{K_1}(x)\right)\right)\right) \\ &= d_{K_1}\left(e_{K_1}(x)\right) \\ &= x \end{aligned} .$$

Cryptosystems have the probability distributions associated with their keyspaces.

$\mathbf{Pr}[(K_1, K_2)] = \mathbf{Pr}[K_1] \times \mathbf{Pr}[K_2]$.

Choose K_1 and K_2 independently, using the probability distributions defined on $\mathcal{K}_1$ and $\mathcal{K}_2$.

Note that the product of a substitution cipher with another substitution cipher is another substitution cipher, so for practical purposes, we want to alternate.

Multiplicative Cipher 1/2

Let $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$ and let $\mathcal{K} = \{a \in \mathbb{Z}_{26} : \gcd(a, 26) = 1\}$.

For $a \in \mathcal{K}$, define $e_a(x) = ax \bmod 26$ and $d_a(y) = a^{-1}y \bmod 26$

$(x, y \in \mathbb{Z}_{26})$.

Suppose **M** is the Multiplicative Cipher and **S** is the Shift Cipher, then **M** $\times$ **S**=**S** $\times$ **M**=Affine Cipher.

Proof:

S: $e_K(x) = (x + K) \bmod 26$, $K \in \mathbb{Z}_{26}$.

M: $e_K(x) = (ax) \bmod 26$, $a \in \mathbb{Z}_{26}$ and $\gcd(a, 26) = 1$.

M $\times$ **S**: $e_{(a,K)}(x) = (ax + K) \bmod 26$.

Multiplicative Cipher 2/2

The probability of a key in Affine Cipher is $\frac{1}{312} = \frac{1}{12} \times \frac{1}{26}$.

S $\times$ **M**: $e_{(K,a)}(x) = a(x + K) \bmod 26 = (ax + aK) \bmod 26$.

The key is (a, aK) in **M** $\times$ **S**. $aK = K_1 \Rightarrow K = a^{-1}K_1$.

M $\times$ **S**=**S** $\times$ **M** $\rightarrow$ **M** and **S** are *commute*. But not all pairs of cryptosystems commute.

The product operation is always *associative*.

Product Cipher

$$\underbrace{\mathbf{S} \times \mathbf{S} \times \cdots \times \mathbf{S}}_{\text{n times}} = S^n$$

If $\mathbf{S}^2 = \mathbf{S}$, then $\mathbf{S}$ is *idempotent cryptosystem*.

If a cryptosystem is not idempotent, then there is a potential increase in security by iterating it several times.

Taking the product of substitution- type ciphers with permutation- type ciphers is a commonly used technique.

Introduction to Block Cipher 1/2

Iterated cipher: The cipher requires the specification of a *round function* and a *key schedule* and the encryption of a plaintext will proceed through N_r similar *rounds*.

K: a random binary key.

N_r *round keys* (*subkeys*): $K^1, \cdots, K^{N_r}$.

Key schedule: the list of round keys $\longrightarrow$ constructed from K using a fixed, public algorithm.

$\omega^r = g\left(\omega^{r-1}, K^r\right)$

ω^r: next state, ω^{r-1}: current state, K^r: round key, g: round function

ω^0: plaintext, x, ω^{N_r}: ciphertext, y

Introduction to Block Cipher 2/2

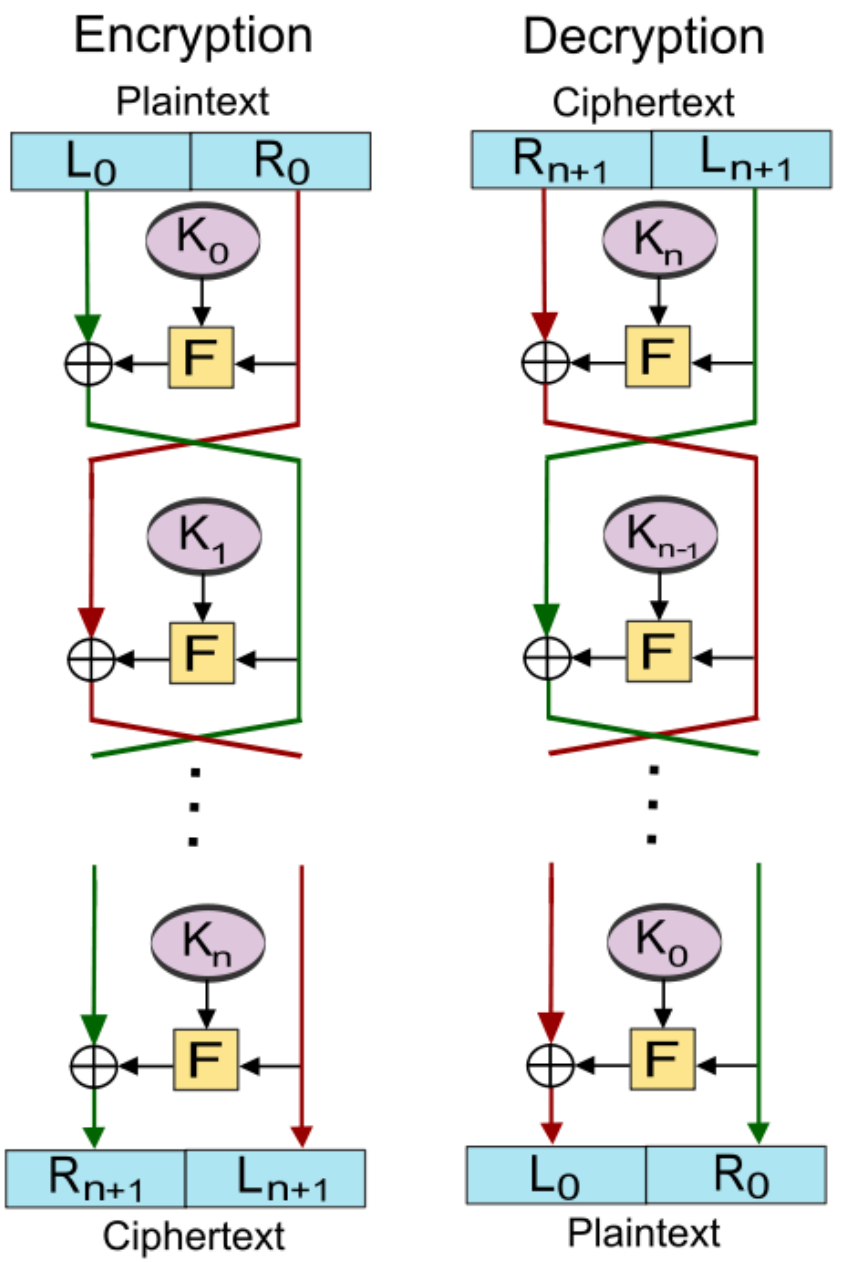
In order for decryption to be possible, the function g must be injective (one-to- one) if its second argument is fixed.

$g^{-1}\left(g\left(\omega, a\right), a\right) = \omega$ for all ω and a.

$\omega^{r-1} = g^{-1}\left(\omega^{r}, K^{r}\right)$

Feistel Cipher

- named after the German-born physicist and cryptographer Horst Feistel who did pioneering research while working for IBM (USA)
- advantage: encryption and decryption operations are very similar, even identical in some cases
- requires only a reversal of the key schedule
- the size of the code or circuitry required to implement such a cipher is nearly halved.



Feistel Cipher - Construction details

Let F be the round function and let $K_0, K_1, \ldots, K_n$ be the sub-keys for the rounds respectively.

Encryption:

1. Split the plaintext block into two equal pieces, (L_0, R_0)
2. For $i = 0, 1, \ldots, n$, compute $L_{i+1} = R_i$, $R_{i+1} = L_i \oplus F(R_i, K_i)$.
3. The ciphertext is $\Big(R_{n+1}, L_{n+1}\Big)$.

Decryption:

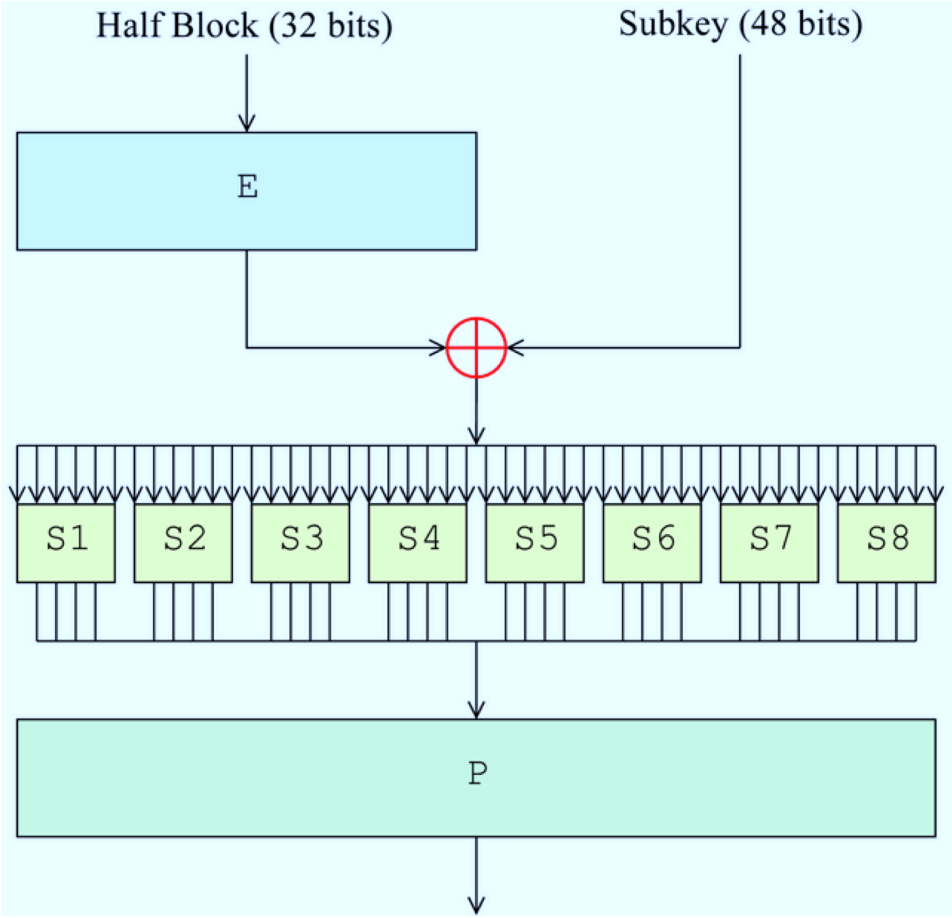
1. Split the ciphertext block into two equal pieces $\Big(R_{n+1}, L_{n+1}\Big)$
2. For $i = n, n-1, \ldots, 0$, compute $R_i = L_{i+1}$, $L_i = R_{i+1} \oplus F\Big(L_{i+1}, K_i\Big)$.
3. The plaintext is (L_0, R_0).

One advantage of the Feistel model compared to a substitution-permutation network is that the round function does not have to be invertible.

Note the reversal of the subkey order for decryption; this is the only difference between encryption and decryption.

Data Encryption Standard

- Selected by the NBS as an official FIPS for the US in 1976.
- Was initially controversial because of classified design elements, a relatively short key length, and suspicions about a NSA backdoor.
- Insecure due to the 56-bit key size being too small
- In January, 1999, distributed.net and the Electronic Frontier Foundation collaborated to publicly break a DES key in 22 hours and 15 minutes.
- The algorithm is believed to be practically secure in the form of Triple DES, although there are theoretical attacks.



Triple Data Encryption Standard

uses a "key bundle" which comprises three DES keys, K_1, K_2 and K_3, each of 56 bits.

The encryption algorithm is: $ciphertext = E_{K_3}\left(D_{K_2}\left(E_{K_1}\left(plaintext\right)\right)\right)$

Decryption is the reverse: $plaintext = D_{K_1}\left(E_{K_2}\left(D_{K_3}\left(ciphertext\right)\right)\right)$

Keying options

1. Keying option 1: All three keys are independent.

   strongest, with 3 56 = 168 independent key bits.

2. Keying option 2: K1 and K2 are independent, and K3 = K1.

   provides less security, with 2 56 = 112 key bits.

3. Keying option 3: All three keys are identical, i.e. K1 = K2 = K3.

   equivalent to DES, with only 56 key bits

Substitution- Permutation Networks (SPNs)

ℓ and m are positive integers

$x = (x_1 x_2 \cdots x_{\ell m})_2$ and $y = (y_1 y_2 \cdots y_{\ell m})_2$

ℓm:block length

S- box: $\pi_S : \{0,1\}^\ell \longrightarrow \{0,1\}^\ell$ is substitution. It is used to replace ℓ bits with a different set of ℓ bits.

$\pi_P : \{1, \ldots, \ell m\} \longrightarrow \{1, \ldots, \ell m\}$ is a permutation. It is used to permute ℓm bits.

$x = (x_1 x_2 \cdots x_{\ell m}) = x_{(1)} \| \cdots \| x_{(m)}$ for $1 \leq i \leq m$

$x_{(i)} = \left(x_{(i-1)\ell+1}, \ldots, x_{i\ell} \right)$.

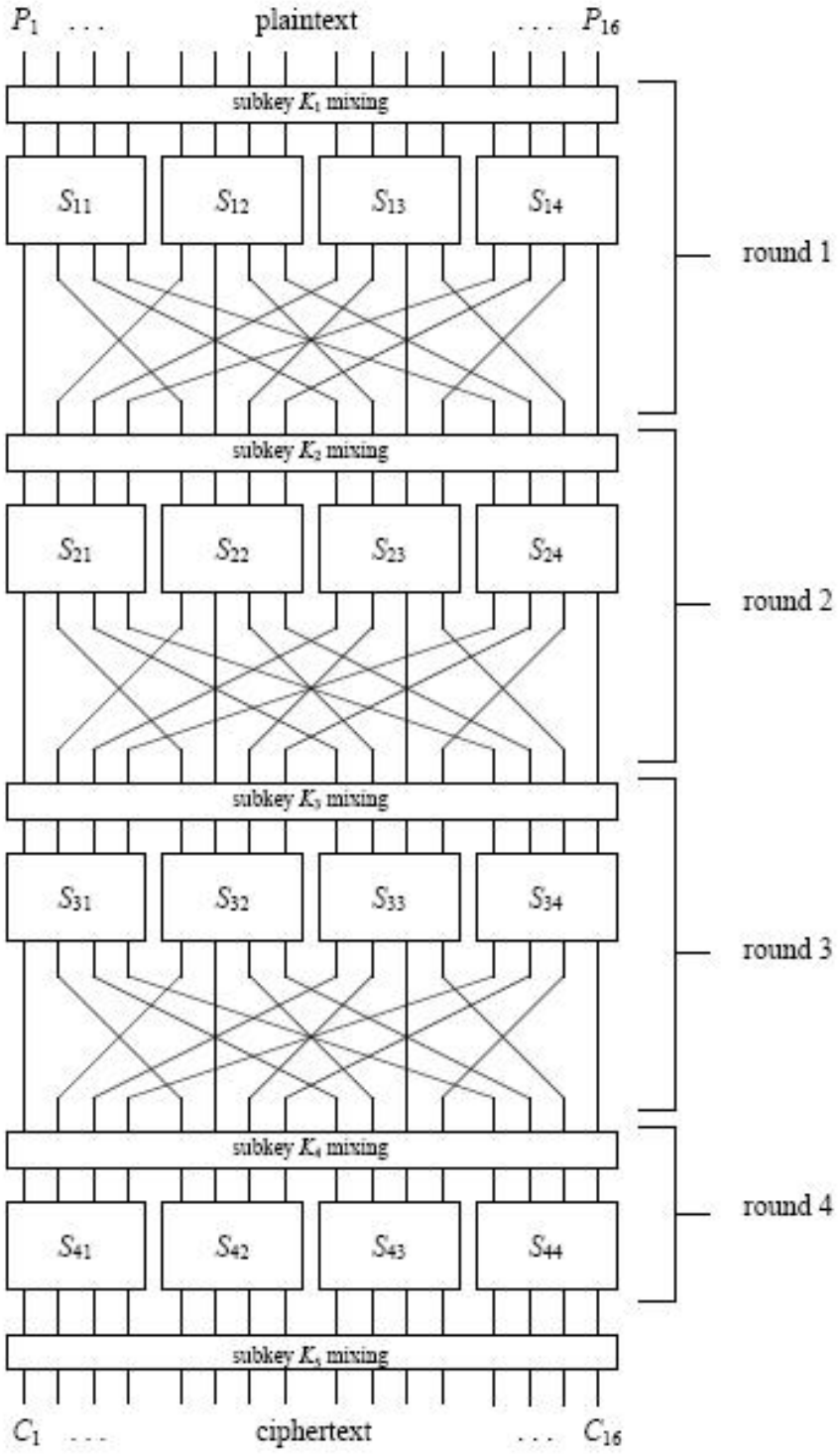
The very first and last operations are XORs with subkeys: *whitenning*.

$\ell = m = N_r = 4$.

z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(z)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

z	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi_P(z)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

Key schedule: $K = (k_1, \ldots, k_{32}) \in \{0,1\}^{32}$.
For $1 \leqslant r \leqslant 5$, $K^r = \left(k_{4r-3}, \ldots, k_{4r+12}\right)$.


P_1 . . . plaintext . . . P_{16}
subkey K_1 mixing
S_{11} S_{12} S_{13} S_{14}
round 1
subkey K_2 mixing
S_{21} S_{22} S_{23} S_{24}
round 2
subkey K_3 mixing
S_{31} S_{32} S_{33} S_{34}
round 3
subkey K_4 mixing
S_{41} S_{42} S_{43} S_{44}
round 4
subkey K_5 mixing
C_1 . . . ciphertext . . . C_{16}

Substitution- Permutation Networks (SPNs)

- The design is simple and very efficient, in both hardware and software.
  - In software, S- box $\longrightarrow$ look- up table. Memory= $\ell 2^{\ell}$.
  - In hardware, needs smaller implementation.

In Example: Memory for S- box= $\ell 2^{\ell} = 4 \times 2^4 = 2^6$.

If the S- box would be 16 bits to 16 bits, then Memory= $\ell 2^{\ell} = 16 \times 2^{16} = 2^{20}$.

A practical secure SPN would have

- a larger key size
- a larger block length
- larger S- Box
- more rounds

$\Longrightarrow$ Advanced Encryption Standard (AES)

Many variations of SPNs are possible

use more than one S- box	Data Encryption Standard (DES)
include an invertible linear transformation in each round	Advanced Encryption Standard (AES)

Linear Cryptanalysis

Take advantage of high probability occurrences of linear expressions involving

- plaintext bits
- ciphertext bits
- subkey bits

a known plaintext attack: that is, it is premised on the attacker having information on a set of plaintexts and the corresponding ciphertexts.

Linear Cryptanalysis

The basic idea is to approximate the operation of a portion of the cipher with an expression that is linear where the linearity refers to a mod-2 bit wise operation.

$$\mathbf{Pr}\left[X_{i_1} \oplus X_{i_2} \oplus \cdots \oplus X_{i_u} \oplus Y_{j_1} \oplus Y_{j_2} \oplus \cdots \oplus Y_{j_v} = 0\right] = p_L \tag{1}$$

The approach in linear cryptanalysis is to determine expressions of the form above which have a high or low probability of occurrence.

If a cipher displays a tendency for equation (1) to hold with high probability or not hold with high probability, this is evidence of the cipher's poor randomization abilities.

It is deviation or bias from the probability of 1/2 is exploited.

linear probability bias: the amount by which the probability of a linear expression holding deviates from 1/2.

Steps of the Linear Cryptanalysis 1/2

1. Suppose that it is possible to find a probabilistic linear relationship between a subset of plaintext bits and a subset of state bits immediately preceding the substitutions performed in the last round.

2. Assume that an attacker has a large number of plaintext- ciphertext pairs, all of which are encrypted using the same unknown key K .

3. Decrypt all the ciphertexts, using all possible candidate keys for the last round of the cipher.

4. For each candidate key, we compute the values of the relevant state bits involved in the linear relationship given by Eq. 1.

5. Determine if the above mentioned linear relationship holds.

Steps of the Linear Cryptanalysis 2/2

6. Whenever it does, we increment a counter corresponding to the particular candidate key.

7. The candidate key that has a frequency count that is furthest from 1/2 times the number of pairs contains the correct values for these key bits.

Meaning of the p_L

- Equation (1) implicitly has subkey bits involved. If the sum of the involved subkey bits is "0", the bias of (1) will have the same sign as the bias of the expression involving the subkey sum and if the sum of the involved subkey bits is "1", the bias of (1) will have the opposite sign as the bias of the expression involving the subkey sum

- $p_L = 1$ implies that linear expression (1) is a perfect representation of the cipher behavior and the cipher has a catastrophic weakness.

- $p_L = 0$, then (1) represents an affine relationship.

How do we construct expressions which are highly linear and hence can be exploited?

This is done by considering the properties of the cipher's only nonlinear component: S- box.

It is possible to concatenate linear approximations of the S- boxes together so that intermediate bits can be canceled out and we are left with a linear expression which has a large bias and involves only plaintext and the last round input bits.

The Pilling- up Lemma

$X_1, X_2, \ldots$: independent random variables and $X_i = 0$ or 1 .

$\mathbf{Pr}[X_i = 0] = p_i$ and $\mathbf{Pr}[X_i = 1] = 1 - p_i$.

$i \neq j \rightarrow$ The independence of X_i and X_j implies that

$$\mathbf{Pr}\left[X_i = 0, X_j = 0\right] = p_i p_j$$

$$\mathbf{Pr}\left[X_i = 0, X_j = 1\right] = p_i \left(1 - p_j\right)$$

$$\mathbf{Pr}\left[X_i = 1, X_j = 0\right] = \left(1 - p_i\right) p_j$$

$$\mathbf{Pr}\left[X_i = 1, X_j = 1\right] = \left(1 - p_i\right) \left(1 - p_j\right) .$$

The Pilling- up Lemma

$X_i \oplus X_j = 0 \Rightarrow X_i = X_j$: linear expression

$$\mathbf{Pr}\left[X_i \oplus X_j = 0\right] = p_i p_j + (1 - p_i)\left(1 - p_j\right)$$

$X_i \oplus X_j = 1 \Rightarrow X_i \neq X_j$: affine expression

$$\mathbf{Pr}\left[X_i \oplus X_j = 1\right] = p_i\left(1 - p_j\right) + (1 - p_i)\, p_j$$

The *bias* of X_i: $\epsilon_i = p_i - \frac{1}{2}$.

Observe the following facts:

$-\frac{1}{2} \leq \epsilon_i \leq \frac{1}{2}$

$\mathbf{Pr}\left[X_i = 0\right] = \frac{1}{2} + \epsilon_i$

$\mathbf{Pr}\left[X_i = 1\right] = \frac{1}{2} - \epsilon_i$

The Pilling- up Lemma

For $i_1 < i_2 < \cdots < i_k$, let $\epsilon_{i_1,i_2,\ldots,i_k}$ denote the bias of the random variable $X_{i_1} \oplus \cdots \oplus X_{i_k}$.

$\mathbf{Pr}\left[X_{i_1} \oplus X_{i_2} = 0\right] = \frac{1}{2} + \epsilon_{i_1,i_2} = \left(\frac{1}{2} + \epsilon_1\right)\left(\frac{1}{2} + \epsilon_2\right) + \left(\frac{1}{2} - \epsilon_1\right)\left(\frac{1}{2} - \epsilon_2\right) = \frac{1}{2} + 2\epsilon_{i_1}\epsilon_{i_2}$.

LEMMA 3.1 (Pilling- up lemma) Let $\epsilon_{i_1,i_2,\ldots,i_k}$ denote the bias of the random variable $X_{i_1} \oplus \cdots \oplus X_{i_k}$. Then

$$\epsilon_{i_1,i_2,\ldots,i_k} = 2^{k-1} \prod_{j=1}^{k} \epsilon_{i_j}$$

.

CORROLLARY 3.2 Let $\epsilon_{i_1,i_2,\ldots,i_k}$ denote the bias of the random variable $X_{i_1} \oplus \cdots \oplus X_{i_k}$. Suppose that $\epsilon_{i_j} = 0$ for some j. Then $\epsilon_{i_1,i_2,\ldots,i_k} = 0$.

In developing the linear approximation of a cipher, the X_i values will actually represent linear approximations of the S- boxes.

Concatenation of Linear Expressions

Consider four independent random binary variables, X_1, X_2, X_3 and X_4. Let $\mathbf{Pr}[X_1 \oplus X_2 = 0] = 1/2 + \epsilon_{1,2}$ and $\mathbf{Pr}[X_2 \oplus X_3 = 0] = 1/2 + \epsilon_{2,3}$.

$\mathbf{Pr}[X_1 \oplus X_3 = 0] = \mathbf{Pr}[(X_1 \oplus X_2) \oplus (X_2 \oplus X_3) = 0]$.

We are combining linear expressions to form a new linear expression.

$\mathbf{Pr}[X_1 \oplus X_3 = 0] = 1/2 + 2\epsilon_{1,2}\epsilon_{2,3}$.

$\epsilon_{1,3} = 2\epsilon_{1,2}\epsilon_{2,3}$.

The expression $X_1 \oplus X_2 = 0$ and $X_2 \oplus X_3 = 0$ are analogous to linear approximation of S- boxes and $X_1 \oplus X_3 = 0$ is analogous to a cipher approximation.

Linear Approximations of S- boxes

S- box $\pi_S : \{0,1\}^m \rightarrow \{0,1\}^n$

input m- tuble $X = (x_1, \ldots, x_m)$ and output n- tuble $Y = (y_1, \ldots, y_n)$

These n random variables are not independent

- if $(y_1, \ldots, y_n) \neq \pi_S(x_1, \ldots, x_m)$ then

  $\mathbf{Pr}[X_1 = x_1, \ldots, X_m = x_m, Y_1 = y_1, \ldots, Y_n = y_n] = 0$

- if $(y_1, \ldots, y_n) = \pi_S(x_1, \ldots, x_m)$ then

  $\mathbf{Pr}[X_1 = x_1, \ldots, X_m = x_m] = 2^{-m}$ and

  $\mathbf{Pr}[Y_1 = y_1, \ldots, Y_n = y_n | X_1 = x_1, \ldots, X_m = x_m,] = 1$

  Hence $\mathbf{Pr}[X_1 = x_1, \ldots, X_m = x_m, Y_1 = y_1, \ldots, Y_n = y_n] = 2^{-m}$

Bias of Linear Expression $X_2 \oplus X_3 \oplus Y_1 \oplus Y_3 \oplus Y_4 = 0$

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4	$X_2 \oplus X_3$	$Y_1 \oplus Y_3 \oplus Y_4$	$X_1 \oplus X_4$	Y_2	$X_3 \oplus X_4$	$Y_1 \oplus Y_4$
0	0	0	0	1	1	1	0	0	0	0	1	0	1
0	0	0	1	0	1	0	0	0	0	1	1	1	0
0	0	1	0	1	1	0	1	1	0	0	1	1	0
0	0	1	1	0	0	0	1	1	1	1	0	0	1
0	1	0	0	0	0	1	0	1	1	0	0	0	0
0	1	0	1	1	1	1	1	1	1	1	1	1	0
0	1	1	0	1	0	1	1	0	1	0	0	1	0
0	1	1	1	1	0	0	0	0	1	1	0	0	1
1	0	0	0	0	0	1	1	0	0	1	0	0	1
1	0	0	1	1	0	1	0	0	0	0	0	1	1
1	0	1	0	0	1	1	0	1	1	1	1	1	0
1	0	1	1	1	1	0	0	1	1	0	1	0	1
1	1	0	0	0	1	0	1	1	1	1	1	0	1
1	1	0	1	1	0	0	1	1	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1	0	1	0
1	1	1	1	0	1	1	1	0	0	0	1	0	1

For exactly 12 out the 16 cases the expression above hold true.

The probability bias is 12/16- 1/2=1/4.

For equation $X_1 \oplus X_4 = Y_2$ the probability bias is 0.

For equation $X_3 \oplus X_4 = Y_1 \oplus Y_4$ the probability bias is 2/16- 1/2=- 3/8.

Linear Approximation Table

Input Sum	Output Sum 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	+8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	−2	−2	0	0	−2	+6	+2	+2	0	0	+2	+2	0	0
2	0	0	−2	−2	0	0	−2	−2	0	0	+2	+2	0	0	−6	+2
3	0	0	0	0	0	0	0	0	+2	−6	−2	−2	+2	+2	−2	−2
4	0	+2	0	−2	−2	−4	−2	0	0	−2	0	+2	+2	−4	+2	0
5	0	−2	−2	0	−2	0	+4	+2	−2	0	−4	+2	0	−2	−2	0
6	0	+2	−2	+4	+2	0	0	+2	0	−2	+2	+4	−2	0	0	−2
7	0	−2	0	+2	+2	−4	+2	0	−2	0	+2	0	+4	+2	0	+2
8	0	0	0	0	0	0	0	0	−2	+2	+2	−2	+2	−2	−2	−6
9	0	0	−2	−2	0	0	−2	−2	−4	0	−2	+2	0	+4	+2	−2
A	0	+4	−2	+2	−4	0	+2	−2	+2	+2	0	0	+2	+2	0	0
B	0	+4	0	−4	+4	0	+4	0	0	0	0	0	0	0	0	0
C	0	−2	+4	−2	−2	0	+2	0	+2	0	+2	+4	0	+2	0	−2
D	0	+2	+2	0	−2	+4	0	+2	−4	−2	+2	0	+2	0	0	+2
E	0	+2	+2	0	−2	−4	0	+2	−2	0	0	−2	−4	+2	−2	0
F	0	−2	−4	−2	−2	0	+2	0	0	−2	+4	−2	−2	0	+2	0

Properties of the Table

- Each element in the table represents the number of matches between the linear equation represented in hexadecimal as "Input Sum" and the sum of the output bits represented in hexadecimal as "Output Sum" minus 8.

  Example: Input Sum=A and Output Sum=6 then expression that is considered is $X_1 \oplus X_3 \oplus Y_2 \oplus Y_3 = 0$

- Hence, dividing an element value by 16 gives the probability bias for the particular linear combination of input and output bits.

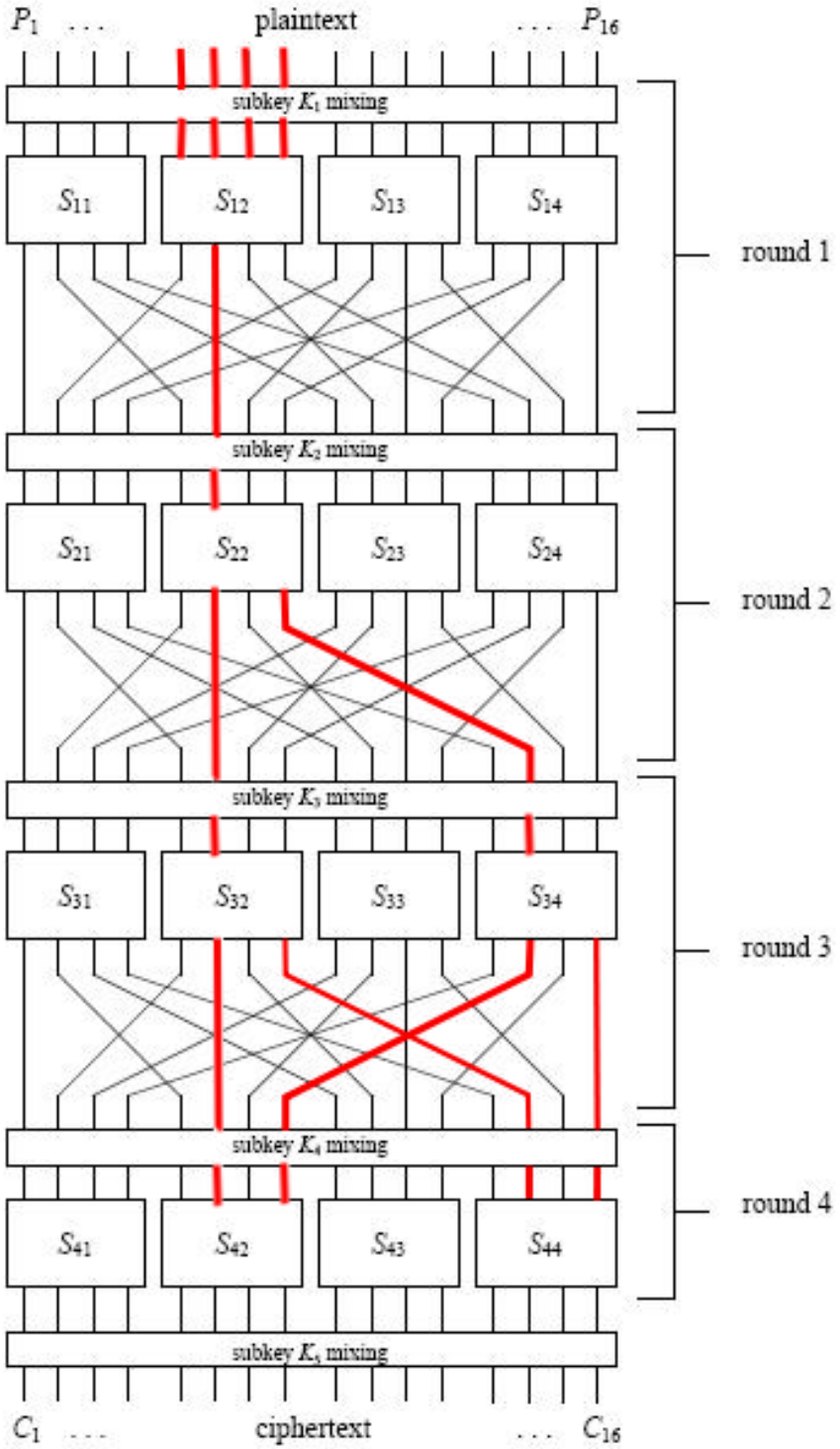
- The linear combination involving no output bits (column 0) will always equal the linear combination of no input bits (row 0) resulting in a bias of +1/2 and a table value of +8 in the top left corner.

- The sum of any row or any column must be either +8 or - 8.

Constructing Linear Approximations for the Complete Cipher

Once the linear approximation information has been compiled for the S- boxes in an SPN, we have the data to proceed with determining linear approximations of the overall cipher of the form of equation (1). This can be achieved by concatenating appropriate linear approximations of S- boxes.

By constructing a linear approximation involving plaintext bits and data bits from the output of the second last round of S- boxes, it is possible to attack the cipher by recovering a subset of the subkey bits that follow the last round.



We would like to use as less as possible S-Boxes. The S-Boxes that are used are called active S-Boxes.

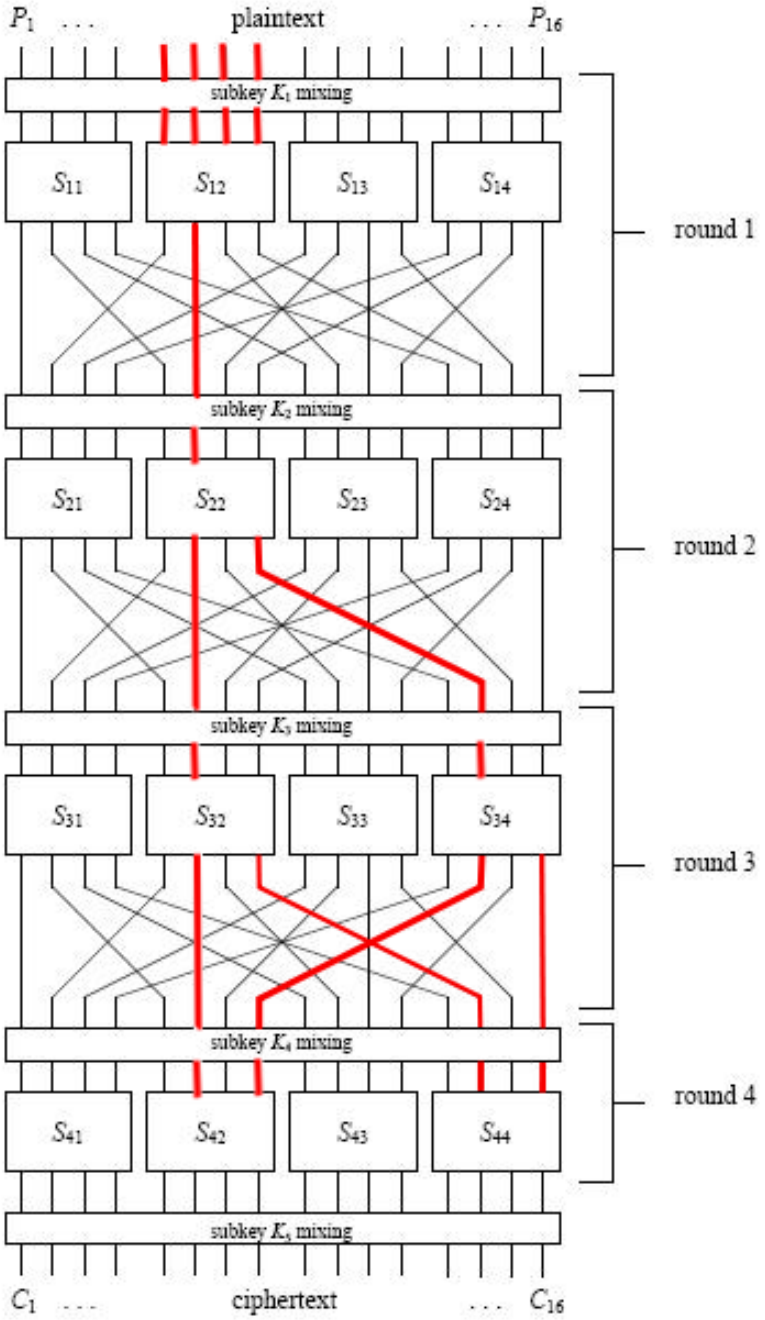
The permutation layer distributes all the outputs of one S-Box to different S-Boxes at the next round. Hence, the best choice is to use just one output bit of an S-Box.

We consider S_{12} for the first round.

Only Y_1, Y_2, Y_3 or Y_4 should be involved in the expression that is used for S_{12}.

The choices are as follows:

	1	2	4	8
9	0	-2	0	-4
A	+4	-2	-4	+2
B	+4	0	+4	0
C	-2	+4	-2	+2
D	+2	+2	-2	-4
F	-2	-4	-2	0



S_{12}: $X_1 \oplus X_3 \oplus X_4 = Y_2$ with probability 12/16 and bias +1/4

The choices for S_{22} are as follows:

	5	D
4	-4	-4

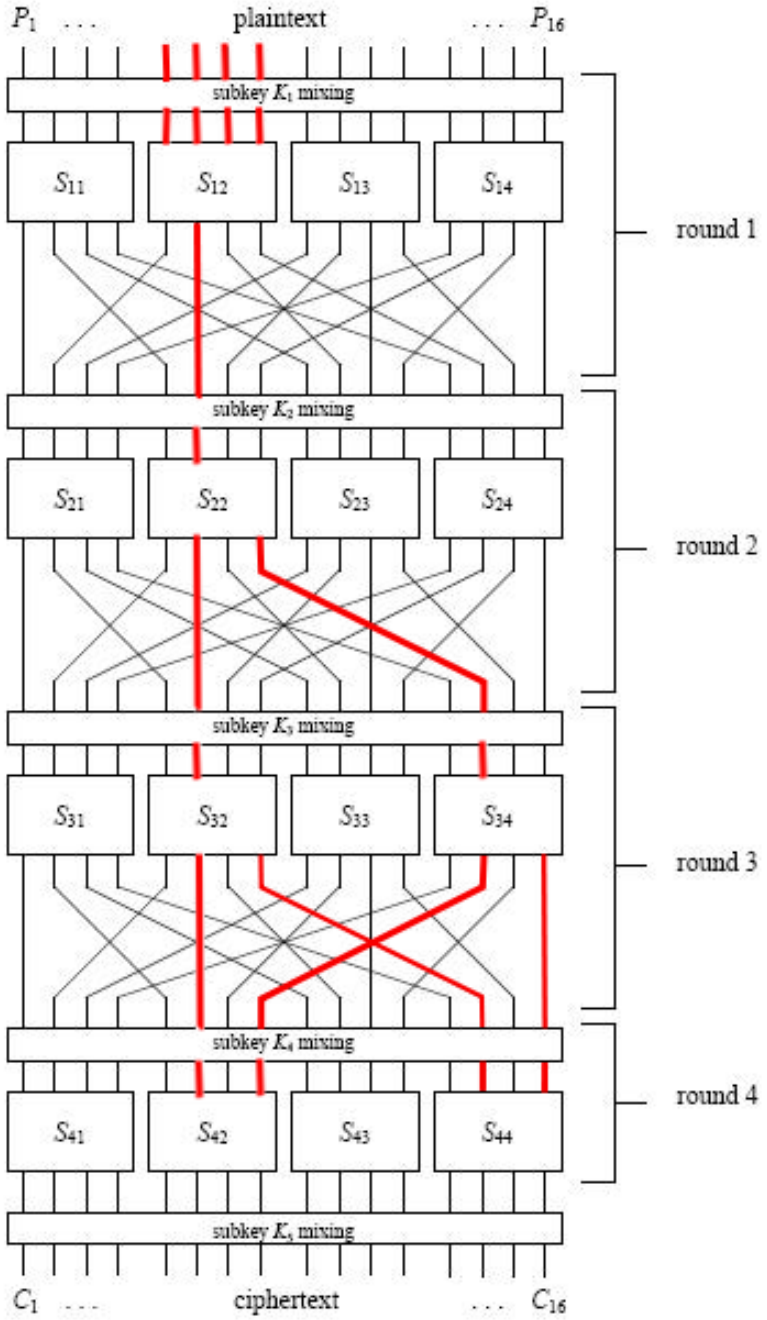
S_{22}: $X_2 = Y_2 \oplus Y_4$ with probability 4/16 and bias - 1/4

The choices for S_{32} and S_{34} are as follows:

	5	D
4	-4	-4

S_{32}: $X_2 = Y_2 \oplus Y_4$ with probability 4/16 and bias - 1/4

S_{34}: $X_2 = Y_2 \oplus Y_4$ with probability 4/16 and bias - 1/4



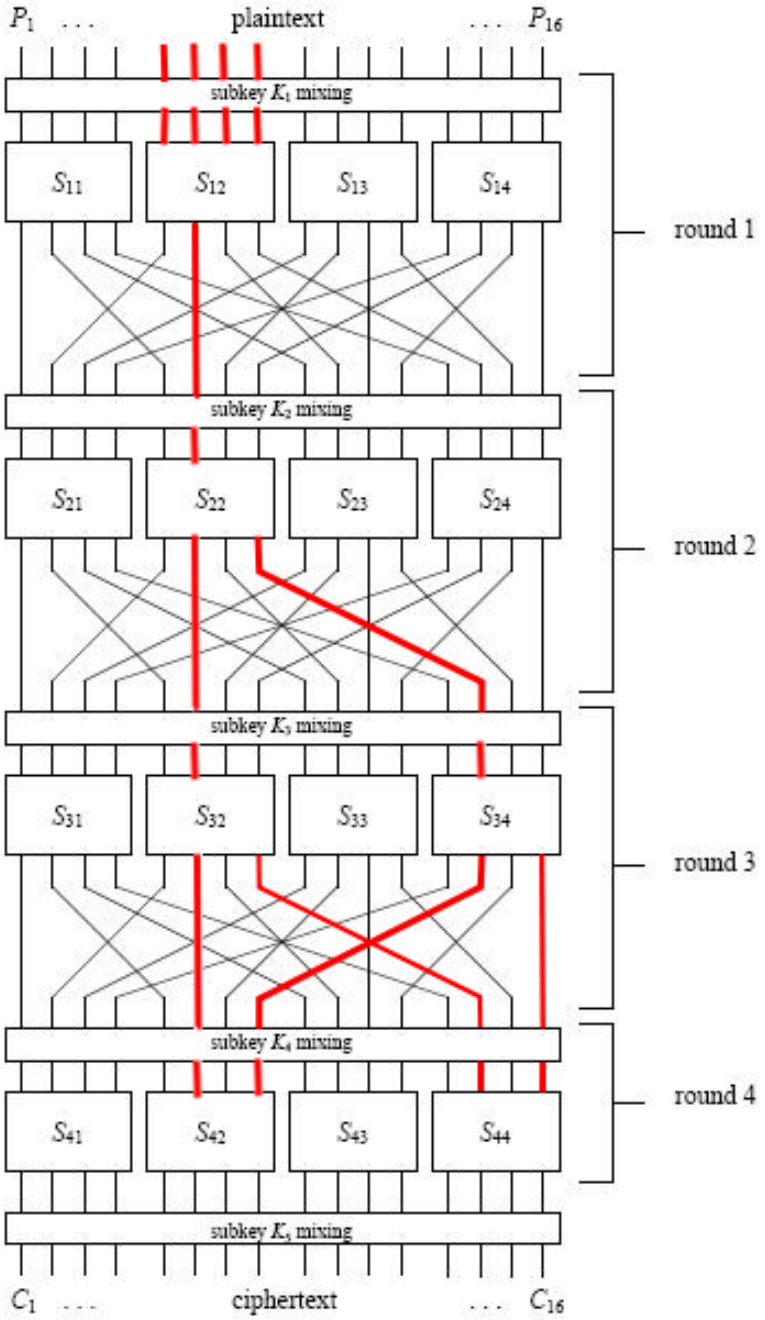
U_i (V_i) represent the 16- bit block of bits at the input (output) of the round i S- boxes.
$U_{i,j}$ ($V_{i,j}$) represent the j- th bit of block U_i (V_i).
K_i represents the subkey block of bits XORed at the input to round i.
1st round, S_{12}: $X_1 \oplus X_3 \oplus X_4 = Y_2$

$$\begin{aligned} V_{1,6} &= U_{1,5} \oplus U_{1,7} \oplus U_{1,8} \\ &= P_5 \oplus K_{1,5} \oplus P_7 \oplus K_{1,7} \oplus P_8 \oplus K_{1,8} \end{aligned} \quad (2)$$

with bias 1/4.
2nd round, S_{22}: $X_2 = Y_2 \oplus Y_4$

$$\begin{aligned} V_{2,6} \oplus V_{2,8} &= U_{2,6} \\ &= V_{1,6} \oplus K_{2,6} \end{aligned} \quad (3)$$

with bias -1/4.



3rd round, S_{32}: $X_2 = Y_2 \oplus Y_4$, S_{34}: $X_2 = Y_2 \oplus Y_4$

$$\begin{aligned} V_{3,6} \oplus V_{3,8} &= U_{3,6} \\ (U_{4,6} \oplus K_{4,6}) \oplus (U_{4,14} \oplus K_{4,14}) &= V_{2,6} \oplus K_{3,6} \end{aligned} \quad (4)$$

with bias -1/4

$$\begin{aligned} V_{3,14} \oplus V_{3,16} &= U_{3,14} \\ (U_{4,8} \oplus K_{4,8}) \oplus (U_{4,16} \oplus K_{4,16}) &= V_{2,8} \oplus K_{3,14} \end{aligned} \quad (5)$$

with bias -1/4.

Now we combine Eq. 2, 3, 4 and 5.

$$U_{4,6} \oplus U_{4,14} \oplus K_{3,6} \oplus K_{4,6} \oplus K_{4,14} = V_{2,6}$$

$$U_{4,8} \oplus U_{4,16} \oplus K_{3,14} \oplus K_{4,8} \oplus K_{4,16} = V_{2,8}$$

$$V_{2,6} \oplus V_{2,8} = V_{1,6} \oplus K_{2,6}$$

$$U_{4,6} \oplus U_{4,14} \oplus K_{3,6} \oplus K_{4,6} \oplus K_{4,14} \oplus U_{4,8} \oplus U_{4,16} \oplus K_{3,14} \oplus K_{4,8} \oplus K_{4,16} = V_{1,6} \oplus K_{2,6}$$

$$U_{4,6} \oplus U_{4,14} \oplus U_{4,8} \oplus U_{4,16} \oplus K_{3,6} \oplus K_{4,6} \oplus K_{4,14} \oplus K_{3,14} \oplus K_{4,8} \oplus K_{4,16} \oplus K_{2,6} = V_{1,6}$$

$$V_{1,6} = P_5 \oplus K_{1,5} \oplus P_7 \oplus K_{1,7} \oplus P_8 \oplus K_{1,8}$$

$$P_5 \oplus P_7 \oplus P_8 \oplus U_{4,6} \oplus U_{4,14} \oplus U_{4,8} \oplus U_{4,16} \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{3,6} \oplus K_{4,6} \oplus K_{4,14} \oplus K_{3,14} \oplus K_{4,8} \oplus K_{4,16} \oplus K_{2,6} = 0$$

By application of the Piling- Up Lemma, the above expression holds with bias $2^3 \frac{1}{4}(-\frac{1}{4})^3 = -\frac{1}{2^5}$.

$$P_5 \oplus P_7 \oplus P_8 \oplus U_{4,6} \oplus U_{4,14} \oplus U_{4,8} \oplus U_{4,16} \oplus \sum K = 0$$

where $\sum K = K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} \oplus K_{3,6} \oplus K_{3,14} \oplus K_{4,6} \oplus K_{4,8} \oplus K_{4,14} \oplus K_{4,16}$

and $\sum K$ is fixed at either 0 or 1 depending on the key of the cipher. Now since $\sum K$ is fixed, we note that

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 = 0 \tag{6}$$

must hold with a probability of either 15/32 or (1- 15/32) = 17/32, depending on whether $\sum K = 0$ or 1, respectively. In other words, we now have a linear approximation of the first three rounds of the cipher with a bias of magnitude -1/32. We must now discuss how such a bias can be used to determine some of the key bits.

Extracting Key Bits

Once an $R-1$ round linear approximation is discovered for a cipher of R rounds with a suitably large enough linear probability bias, it is conceivable to attack the cipher by recovering bits of the last subkey.

In the case of our example cipher, it is possible to extract bits from subkey K_5 given a 3 round linear approximation.

We shall refer to the bits to be recovered from the last subkey as the **target partial subkey**. Specifically, the target partial subkey bits are the bits from the last subkey associated with the S- boxes in the last round influenced by the data bits involved in the linear approximation.

The process followed involves partially decrypting the last round of the cipher.

Specifically, for all possible values of the target partial subkey, the corresponding ciphertext bits are exclusive- ORed with the bits of the target partial subkey and the result is run backwards through the corresponding S- boxes.

This is done for all known plaintext/ciphertext samples and a count is kept for each value of the target partial subkey. The count for a particular target partial subkey value is incremented when the linear expression holds true for the bits into the last rounds S- boxes (determined by the partial decryption) and the known plaintext bits.

The target partial subkey value which has the count which differs the greatest from half the number of plaintext/ciphertext samples is assumed to represent the correct values of the target partial subkey bits.

This works because it is assumed that the correct partial subkey value will result in the linear approximation holding with a probability significantly different from 1/2. (Whether it is above or below 1/2 depends on whether a linear or affine expression is the best approximation and this depends on the unknown values of the subkey bits implicitly involved in the linear expression.)

An incorrect subkey is assumed to result in a relatively random guess at the bits entering the S- boxes of the last round and as a result, the linear expression will hold with a probability close to 1/2.

Lets now put this into the context of our example. The linear expression of (6) affects the inputs to S- boxes S_{42} and S_{44} in the last round.

For each plaintext/ciphertext sample, we would try all 256 values for the target partial subkey $[K_{5,5} \ldots K_{5,8}, K_{5,13} \ldots K_{5,16}]$.

For each partial subkey value, we would increment the count whenever equation (6) holds true, where we determine the value of $[U_{4,5} \ldots U_{4,8}, U_{4,13} \ldots U_{4,16}]$ by running the data backwards through the target partial subkey and S- boxes S_{24} and S_{44}.

The count which deviates the largest from half of the number of plaintext/ciphertex samples is assumed to the correct value.

Whether the deviation is positive or negative will depend on the values of the subkey bits involved in $\sum K$. When $\sum K = 0$, the linear approximation of (6) will serve as the estimate (with probability ¡ 1/2) and when $\sum K = 1$, (6) will hold with a probability ¿ 1/2.

partial subkey $[K_{5,5}...K_{5,8}, K_{5,13}...K_{5,16}]$	| bias |	*partial subkey* $[K_{5,5}...K_{5,8}, K_{5,13}...K_{5,16}]$	| bias |
1 C	0.0031	2 A	0.0044
1 D	0.0078	2 B	0.0186
1 E	0.0071	2 C	0.0094
1 F	0.0170	2 D	0.0053
2 0	0.0025	2 E	0.0062
2 1	0.0220	2 F	0.0133
2 2	0.0211	3 0	0.0027
2 3	0.0064	3 1	0.0050
2 4	**0.0336**	3 2	0.0075
2 5	0.0106	3 3	0.0162
2 6	0.0096	3 4	0.0218
2 7	0.0074	3 5	0.0052
2 8	0.0224	3 6	0.0056
2 9	0.0054	3 7	0.0048

Although the correct target partial subkey has clearly the highest bias, other large bias values occur indicating that the examination of incorrect target partial subkeys is not precisely equivalent to comparing random data to a linear expression (where the bias could be expected to be very close to zero).

Inconsistencies in the experimental biases can occur for several reasons including the S- box properties influencing the partial decryption for different partial subkey values, the imprecision of the independence assumption required for use in the Piling- Up Lemma, and the influence of linear hulls (to be discussed in the next section).

Complexity of Attack

We refer to the S- boxes involved in the linear approximation as active S- boxes.

In Figure 3, the four S- boxes in rounds 1 to 3 influenced by the highlighted lines are active. The probability that a linear expression holds true is related to the linear probability bias in the active S- boxes and the number of active S- boxes.

In general, the larger the magnitude of the bias in the S- boxes, the larger the magnitude of the bias of the overall expression.

Also, the fewer active S- boxes, the larger the magnitude of the overall linear expression bias.

Let ϵ represent the bias from 1/2 of the probability that the linear expression for the complete cipher holds. The number of known plaintexts required in the attack is proportional to ϵ^{-2} and, letting N_L represent the number of known plaintexts required, it is reasonable to approximate N_L by $NL \approx 1/\epsilon^{-2}$.

In practice, it is generally reasonable to expect some small multiple of ϵ^{-2} known plaintexts are required.

The complexity of the cryptanalysis could be characterized in both time and space (or memory) domains, we refer to the data required to mount the attack when considering the complexity of the cryptanalysis.

We assume that if we are able to acquire N_L plaintexts, we are able to process them. Since the bias is derived using the Piling- Up Lemma where each term in the product refers to an S- box approximation, it is easy to see that the bias is dependent on the biases of the S- box linear approximations and the number of active S- boxes involved.

General approaches to providing security against linear cryptanalysis have focused on optimizing the S- boxes (i.e., minimizing the largest bias) and finding structures to maximize the number of active S- boxes.

The design principles of Rijndael are an excellent example of such an approach.

It must be cautioned, however, the concept of a "proof" of security to linear cryptanalysis is usually premised on the nonexistence of highly likely linear

approximations. However, the computation of the probability of such linear approximations is based on the assumption that each S- box approximation is independent (so that the Piling- Up Lemma can be used) and on the assumption that one linear approximation scenario (i.e., a particular set of active S-boxes) is sufficient to determine the best linear expression between plaintext bits and data bits at the input to the last round.

The reality is that the Sbox approximations are not independent and this can have significant impact on the computation of the probability.

Also, linear approximation scenarios involving the same plaintext and last round input bits but different sets of active S- boxes can combine to give a linear probability higher than that predicted by one set of active S- boxes. This concept is referred to as a **linear hull**.(**Odev**)

Most notably for example, a number of linear approximation scenarios may have very small biases and on their own seem to imply that a cipher might be immune to a linear attack. However, when these scenarios are combined, the resulting linear expression of plaintext and last round input bits might have a very high bias. Nevertheless, the approach outlined in this paper, tends to work well for many ciphers because the independence assumption is a reasonable approximation and when one linear approximation scenario of a particular set of active S- boxes has a high bias, it tends to dominate the linear hull.

Differential Cryptanalysis

Differential cryptanalysis exploits the high probability of certain occurrences of plaintext differences and differences into the last round of the cipher.

A system with input $X = [X_1 X_2 \dots X_n]$ and output $Y = [Y_1 Y_2 \dots Y_n]$. Let two inputs to the system be X' and X'' with the corresponding outputs Y' and Y'', respectively.

The input difference is given by $\Delta X = X' \oplus X''$ where $\oplus$ represents a bit-wise exclusive-OR of the n-bit vectors and, hence,

$\Delta X = [\Delta X_1 \Delta X_2 \dots \Delta X_n]$ where $\Delta X_i = X'_i \oplus X''_i$.

Similarly, $\Delta Y = Y' \oplus Y''$ is the output difference and

$\Delta Y = [\Delta Y_1 \Delta Y_2 \dots \Delta Y_n]$

Differential

In an ideally randomizing cipher, the probability that a particular output difference ΔY occurs given a particular input difference ΔX is $\frac{1}{2}^n$ where n is the number of bits of X.

Differential cryptanalysis seeks to exploit a scenario where a particular ΔY occurs given a particular input difference ΔX with a very high probability p_D (i.e., much greater than $\frac{1}{2}^n$). The pair $(\Delta X, \Delta Y)$ is referred to as a differential.

Differential cryptanalysis is a chosen plaintext attack.

The attacker will select pairs of inputs, X' and X'', to satisfy a particular ΔX, knowing that for that ΔX value, a particular ΔY value occurs with high probability.

Investigate the construction of a differential $(\Delta X, \Delta Y)$.

Differential Characteristic

Examine high likely differential characteristics where a differential characteristic is a sequence of input and output differences to the rounds so that the output difference from one round corresponds to the input difference for the next round.

Using the highly likely differential characteristic gives us the opportunity to exploit information coming into the last round of the cipher to derive bits from the last layer of subkeys.

Example 1 for Differentials

$X, Y \in 0, 1, \ldots, 4$

$Y = f(X) = 3X$, $\Delta X = X' - X''$, $Y' = f(X')$, $Y'' = f(X'')$, $f(\Delta X) = 5(X' - X'') = Y' - Y'' = \Delta Y$

		ΔY				
X'	Y'	$\Delta X = 0$	$\Delta X = 1$	$\Delta X = 2$	$\Delta X = 3$	$\Delta X = 4$
0	0	0	3	1	4	2
1	3	0	3	1	4	2
2	1	0	3	1	4	2
3	4	0	3	1	4	2
4	2	0	3	1	4	2

		ΔY				
		0	**1**	**2**	**3**	**4**
	0	5	0	0	0	0
	1	0	0	0	5	0
ΔX	**2**	0	5	0	0	0
	3	0	0	0	0	5
	4	0	5	0	0	0

$Pr(\Delta Y = 1 | \Delta X = 2) = 1,$
$Pr(\Delta Y = 2 | \Delta X = 2) = 0$

Example 2 for Differentials

$X, Y \in 0, 1, \ldots, 4$

$Y = f(X) = 3X + 2,\ \Delta X = X' - X'',\ X' = 4 \rightarrow Y' = f(X') = 4,\ X'' = 3 \rightarrow Y'' = f(X'') = 1,\ f(\Delta X) = 3(X' - X'') + 2 = 0 \neq Y' - Y''$

		ΔY				
X'	Y'	$\Delta X = 0$	$\Delta X = 1$	$\Delta X = 2$	$\Delta X = 3$	$\Delta X = 4$
0	2	0	3	0	4	1
1	1	0	4	2	4	3
2	3	0	2	1	4	1
3	2	0	4	1	0	3
4	4	0	2	1	3	2

		ΔY				
		0	**1**	**2**	**3**	**4**
	0	5	0	0	0	0
	1	0	0	2	1	2
ΔX	**2**	1	3	1	0	0
	3	1	0	0	1	3
	4	0	2	1	2	0

$Pr(\Delta Y = 1 | \Delta X = 2) = \frac{3}{5},$
$Pr(\Delta Y = 2 | \Delta X = 2) = \frac{1}{5}$

Differential of S-Box

Examine the properties of individual S-boxes and use these properties to determine the complete differential characteristic.

Consider the input and output differences of the S-boxes in order to determine a high probability difference pair.

Combining S-box difference pairs from round to round so that the nonzero output difference bits from one round correspond to the non-zero input difference bits of the next round, enables us to find a high probability differential consisting of the plaintext difference and the difference of the input to the last round.

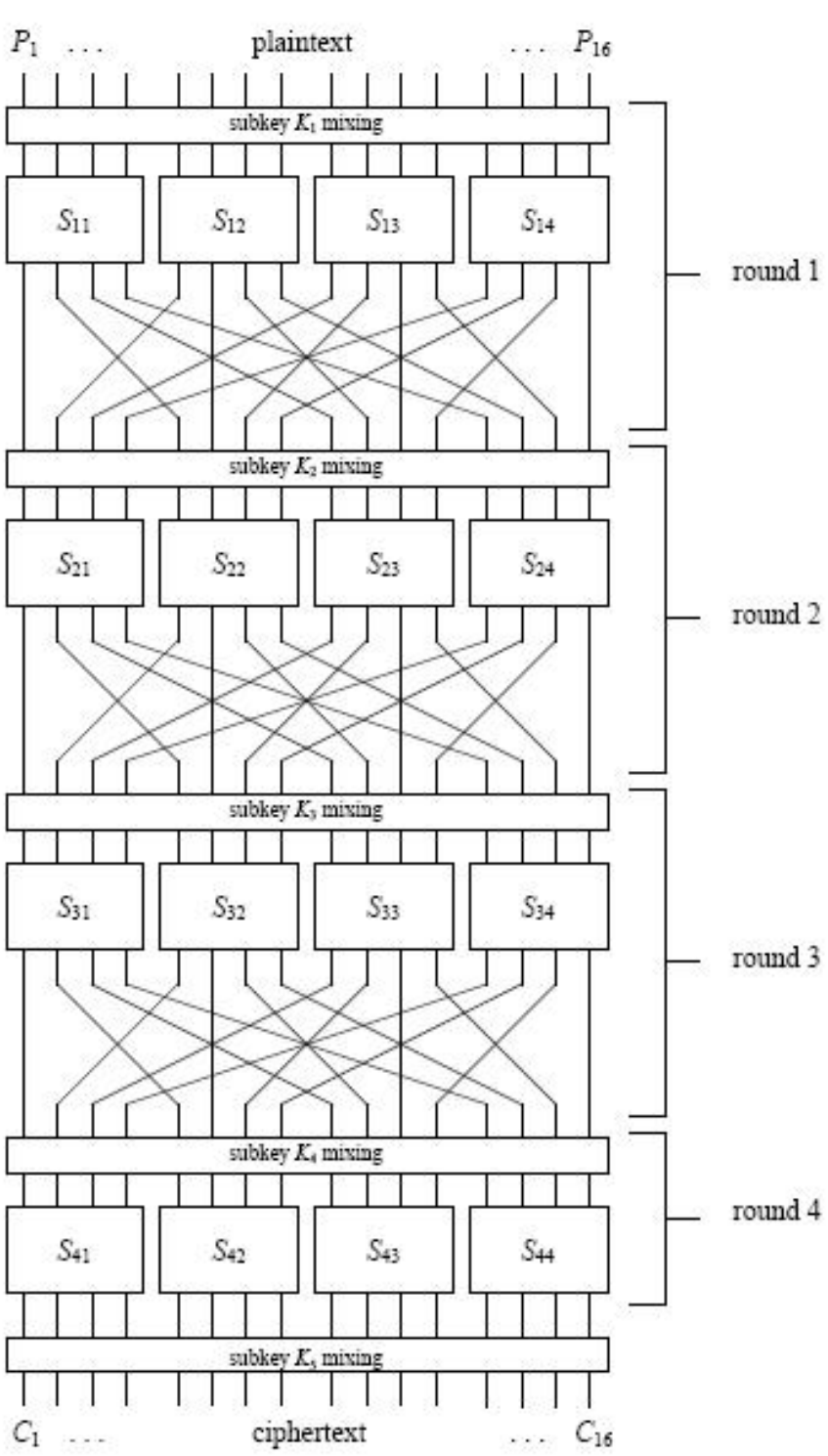
z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$\pi_S(z)$	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

$Pr(\Delta Y|\Delta X)$ can be derived by considering input pairs (X', X'') such that $X' \oplus X'' = \Delta X$.

X	Y	ΔY		
		$\Delta X = 1011$	$\Delta X = 1000$	$\Delta X = 0100$
0000	1110	0010	1101	1100
0001	0100	0010	1110	1011
0010	1101	0111	0101	0110
0011	0001	0010	1011	1001
0100	0010	0101	0111	1100
0101	1111	1111	0110	1011
0110	1011	0010	1011	0110
0111	1000	1101	1111	1001
1000	0011	0010	1101	0110
1001	1010	0111	1110	0011
1010	0110	0010	0101	0110
1011	1100	0010	1011	1011
1100	0101	1101	0111	0110
1101	1001	0010	0110	0011
1110	0000	1111	1011	0110
1111	0111	0101	1111	1011

Input Difference	Output Difference 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

S-box difference distribution table



$S_{11} : Pr(\Delta Y = 4 | \Delta X = F) = \frac{6}{16}$

$S_{22} : Pr(\Delta Y = 6 | \Delta X = 8) = \frac{2}{16}$

$S_{32} : Pr(\Delta Y = 6 | \Delta X = 4) = \frac{6}{16}$

$S_{33} : Pr(\Delta Y = 6 | \Delta X = 4) = \frac{6}{16}$

$\Delta P = [1111\,0000\,0000\,0000]$

$\Delta U_1 = [1111\,0000\,0000\,0000]$

$\Delta V_1 = [0100\,0000\,0000\,0000]$

$\Delta U_2 = [0000\,1000\,0000\,0000]$

$\Delta V_2 = [0000\,0110\,0000\,0000]$

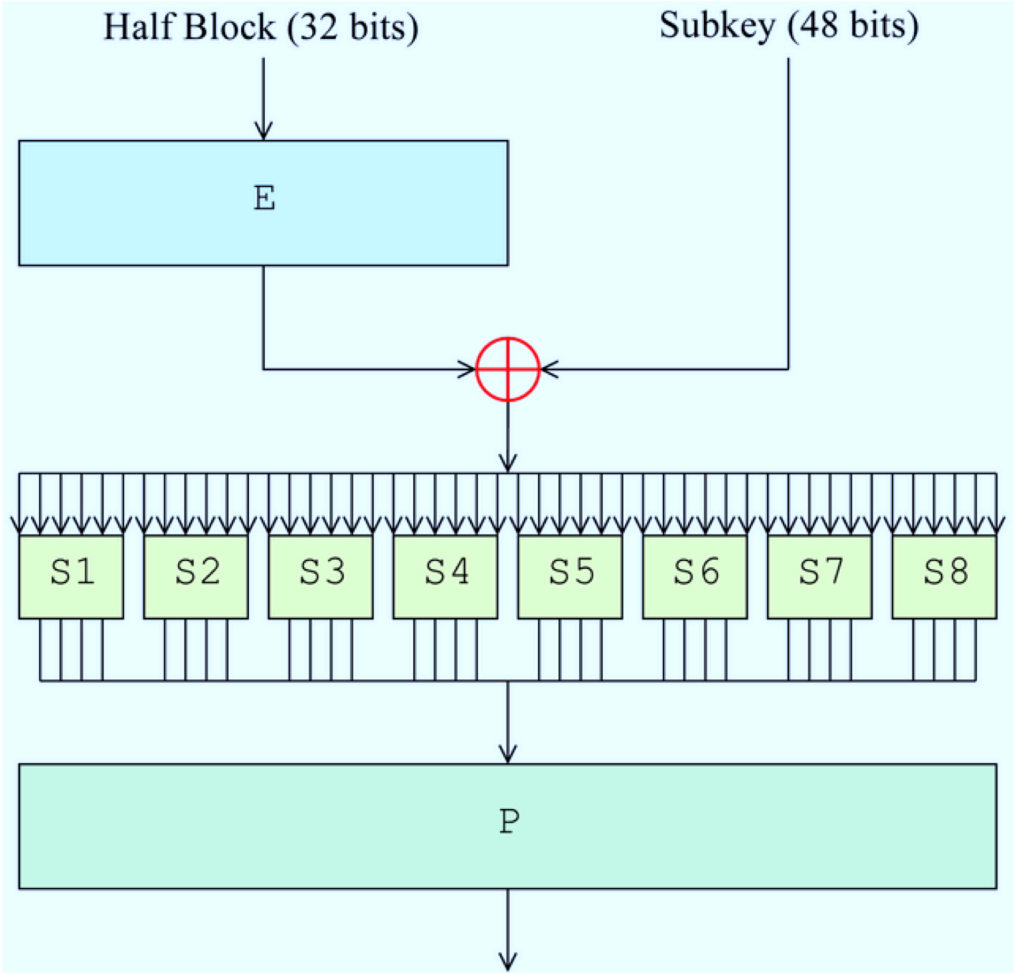
$\Delta U_3 = [0000\,0100\,0100\,0000]$

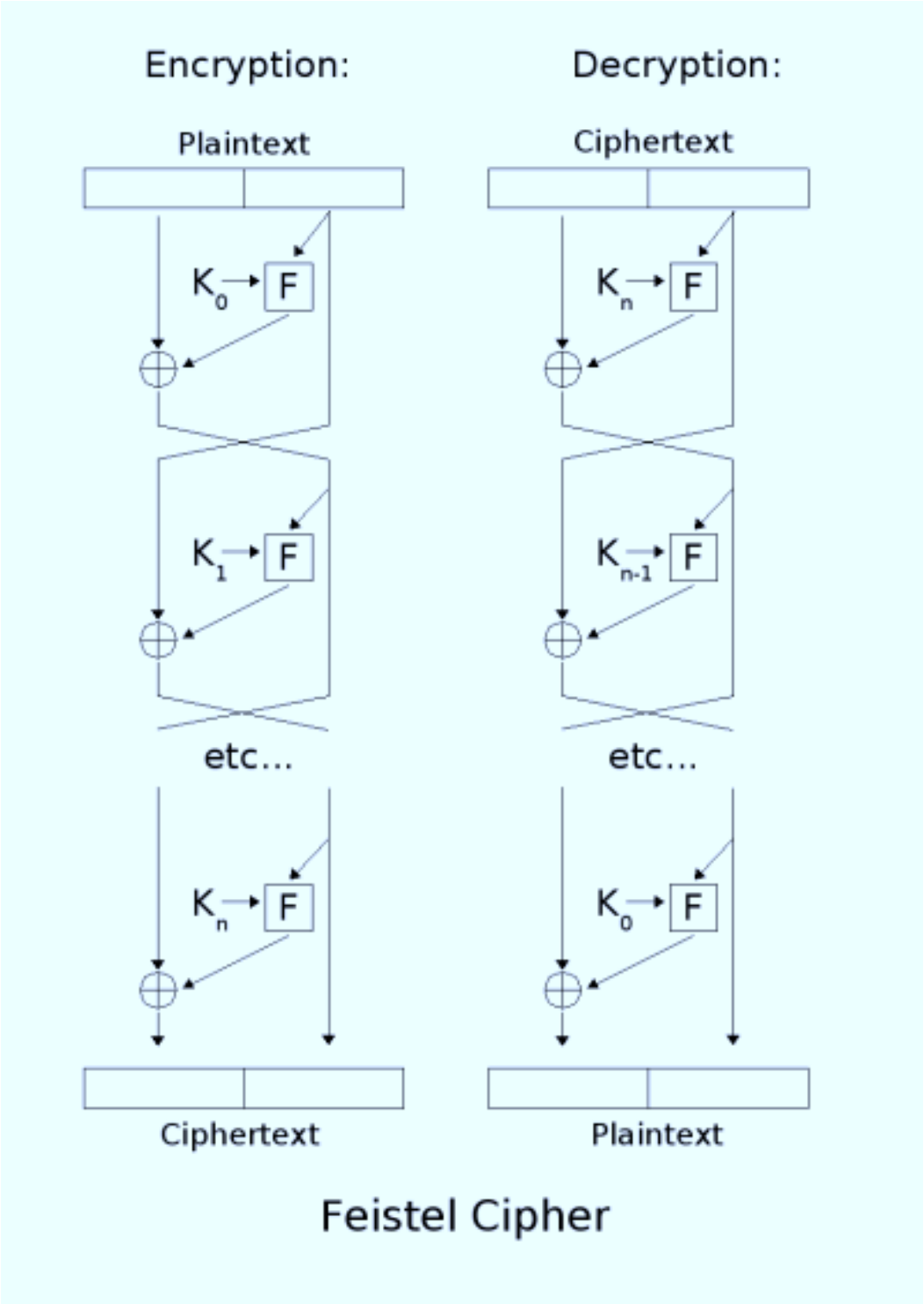
$\Delta V_3 = [0000\,0110\,0110\,0000]$

$\Delta U_4 = [0000\,0110\,0110\,0000]$

$Pr(\Delta P = [1111\,0000\,0000\,0000] | \Delta U_4 = [0000\,0110\,0110\,0000]) = (\frac{6}{16})^3 \times \frac{2}{16} = \frac{27}{4096}$

Data Encryption Standard - DES





Feistel Cipher

Triple DES

Algorithm Triple DES uses a "key bundle" which comprises three DES keys, K_1, K_2 and K_3, each of 56 bits. The encryption algorithm is:

$$ciphertext = E_{K_3}\left(D_{K_2}\left(E_{K_1}\left(plaintext\right)\right)\right)$$

DES encrypt with K_1, DES decrypt with K_2, then DES encrypt with K_3.

Decryption is the reverse:

$$plaintext = D_{K_1}\left(E_{K_2}\left(D_{K_3}\left(ciphertext\right)\right)\right)$$

Decrypt with K_3, encrypt with K_2, then decrypt with K_1.

Each triple encryption encrypts one block of 64 bits of data.

In each case the middle operation is the reverse of the first and last. This improves the strength of the algorithm when using keying option 2, and provides backward compatibility with DES with keying option 3.

1. Keying option 1: All three keys are independent.

2. Keying option 2: K_1 and K_2 are independent, and $K_3 = K_1$.

3. Keying option 3: All three keys are identical, i.e. $K_1 = K_2 = K_3$.

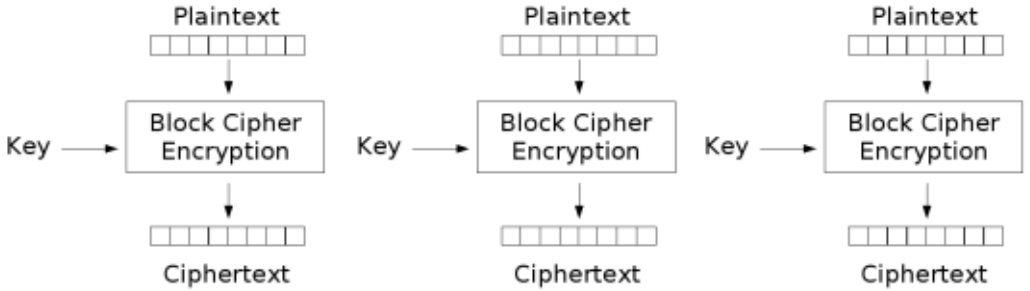
Modes of operation

For messages exceeding block length, n, the message is partitioned into n-bit blocks.

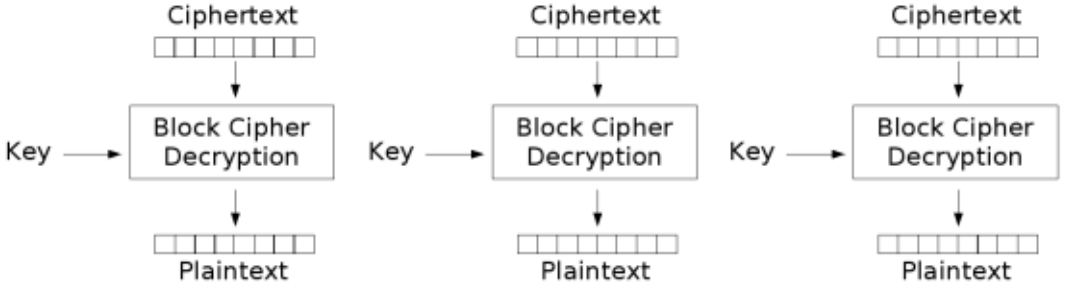
E_K: the encryption function

E^{-1}: the decryption function

$x = x_1, \ldots, x_t$: A plaintext message



Electronic Codebook (ECB) mode encryption

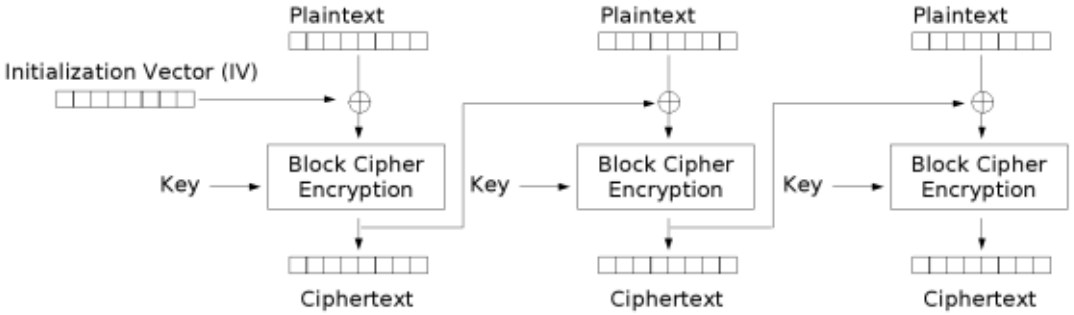


Electronic Codebook (ECB) mode decryption

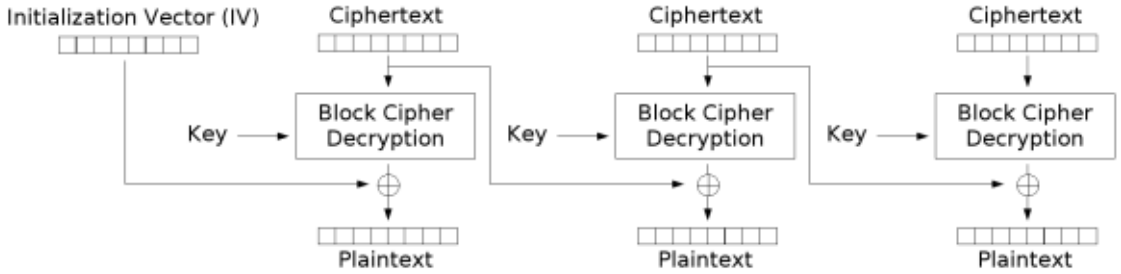
1. Identical plaintext blocks result in identical ciphertext.

2. Chaining dependencies: Reordering ciphertext blocks results in correspondingly re-ordered plaintext blocks.

3. Error propagation: one or more bit errors in a single ciphertext block affect decipherment of that block only.



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

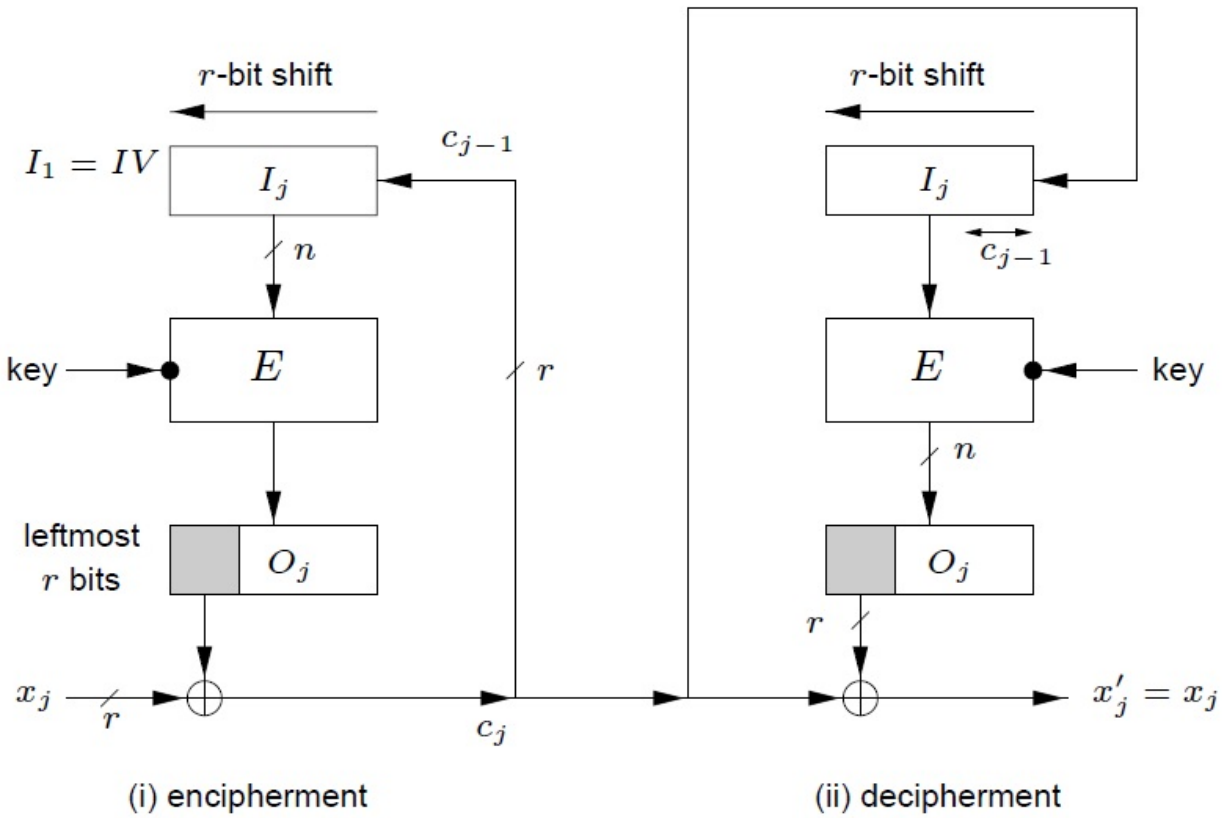
1. Identical plaintexts: identical ciphertext blocks result when the same plaintext is enciphered under the same key and IV.

2. Chaining dependencies: ciphertext c_j depends on x_j and all preceding plaintext blocks.

3. Error propagation: a single bit error in ciphertext block c_j affects decipherment of blocks c_j and c_{j+1}.

4. Error recovery: the CBC mode is self-synchronizing or ciphertext autokey.

Cipher feedback (CFB) Mode



1. Identical plaintexts: changing the IV results in the same plaintext input being enciphered to a different output.

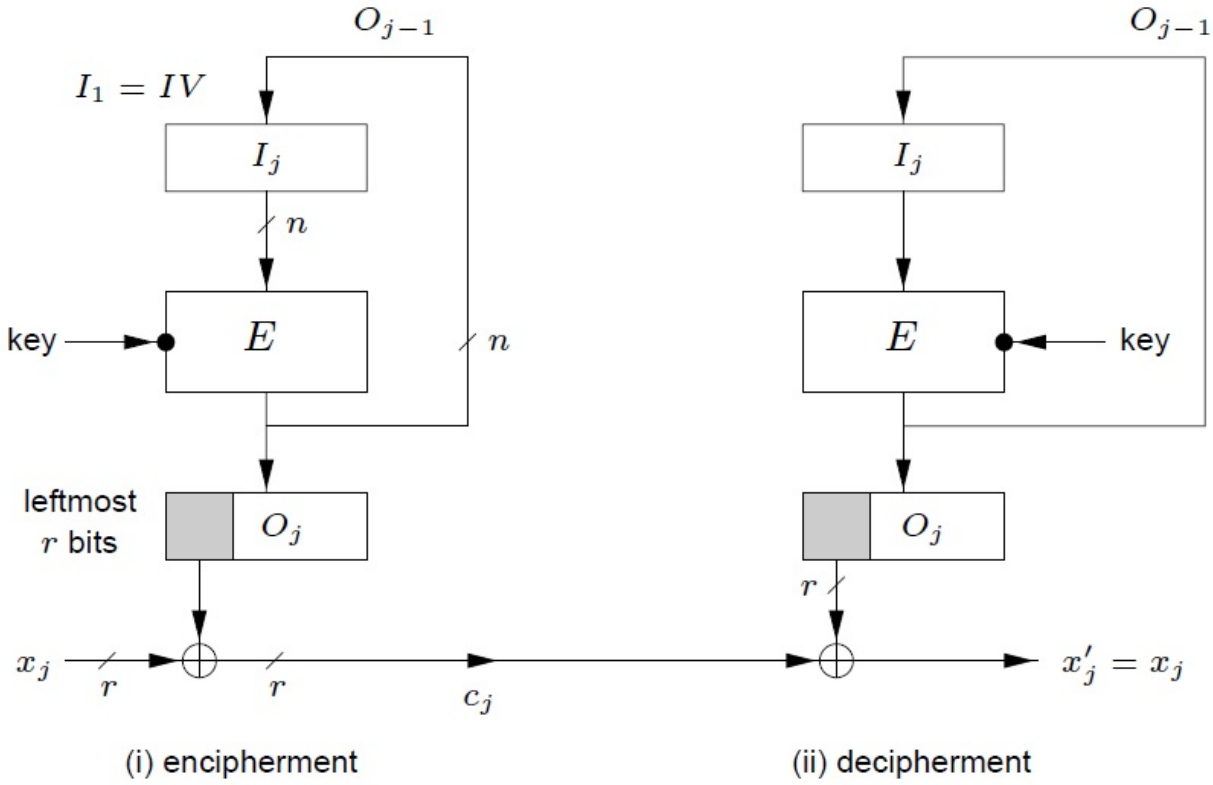
2. Chaining dependencies: ciphertext block c_j to depend on both x_j and preceding plaintext blocks.

3. Error propagation: one or more bit errors in any single r-bit ciphertext block c_j affects the decipherment of that and the next $\lceil \frac{n}{r} \rceil$ ciphertext blocks.

4. Error recovery: the CFB mode is self-synchronizing, but requires $\lceil \frac{n}{r} \rceil$ ciphertext blocks to recover.

5. Throughput: for $r < n$, throughput is decreased by a factor of $\frac{n}{r}$.

Output feedback (OFB) Mode



1. Identical plaintexts: changing the IV results in the same plaintext being enciphered to a different output.

2. Chaining dependencies: the keystream is plaintext-independent.

3. Error propagation: one or more bit errors in any ciphertext character c_j affects the decipherment of only that character, in the precise bit position(s)

c_j is in error, causing the corresponding recovered plaintext bit(s) to be complemented.

4. Error recovery: the OFB mode recovers from ciphertext bit errors, but cannot selfsynchronize after loss of ciphertext bits, which destroys alignment of the decrypting keystream.

5. Throughput: for $r < n$, throughput is decreased as per the CFB mode. However, in all cases, since the keystream is independent of plaintext or ciphertext, it may be pre-computed (given the key and IV).

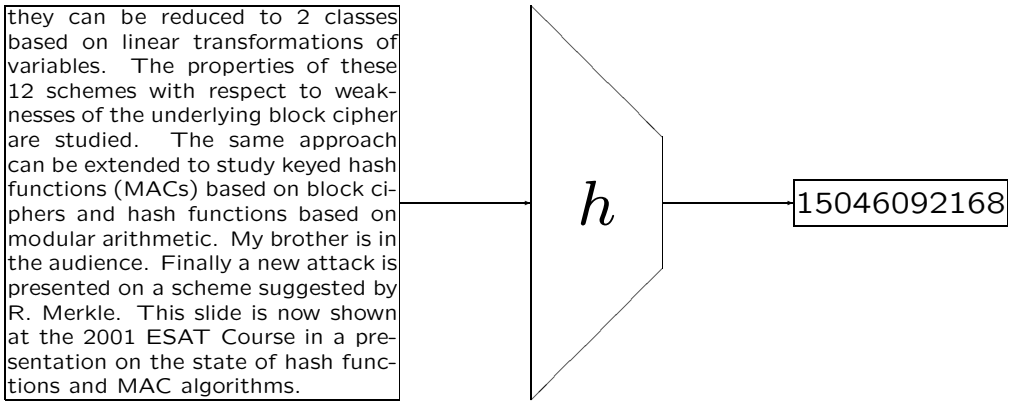
Data Integrity and Source Authentication

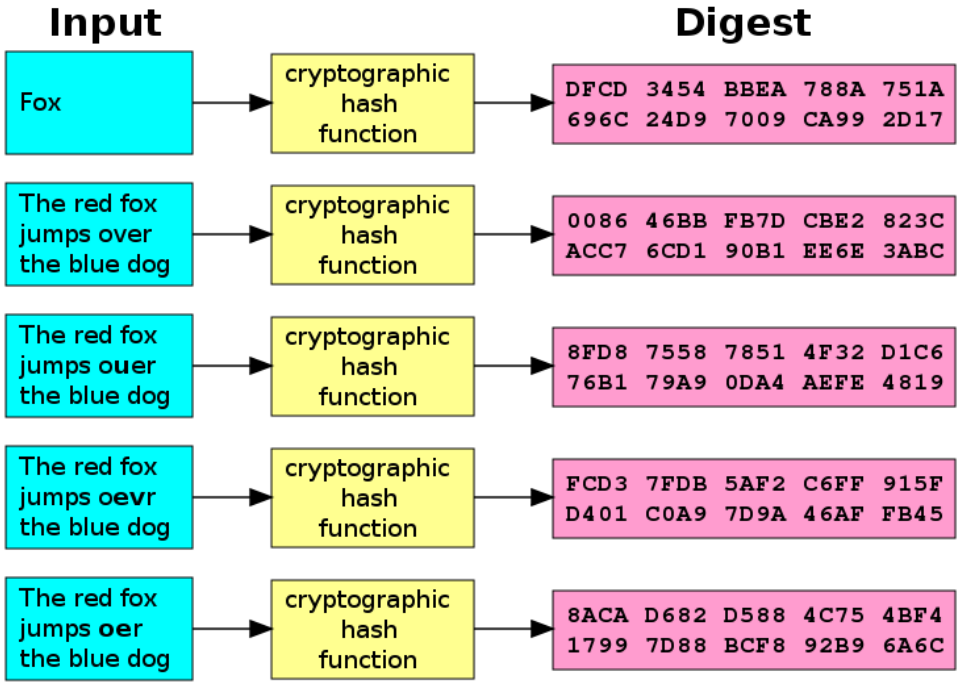
- Encryption does not protect data from modification by another party.

- Need a way to ensure that data arrives at destination in its original form as sent by the sender and it is coming from an authenticated source.

Hash Functions

Definition - hash function





A cryptographic hash function is a deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string, the hash value, such that an accidental or intentional change to the data will change the hash value.

The data to be encoded is often called the message and the hash value is sometimes called the message digest or simply digest.

Properties of Ideal Cryptographic Hash Functions

It is

1. easy to compute the hash value for any given message,
2. infeasible to find a message that has a given hash,
3. infeasible to modify a message without hash being changed,
4. infeasible to find two different messages with the same hash.

Even if the data is stored in an insecure place, its integrity can be checked from time to time by recomputing the digest and verifying that the digest has not changed.

Definition of Hash Family

A hash family is a four tuple $\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{H}$, where teh following conditions are satisfied:

1. $\mathcal{X}$ is a set of possible messages
2. $\mathcal{Y}$ is a finite set of possible message digests
3. $\mathcal{K}$ is a finite set of possible keys
4. For each $K \in \mathcal{K}$, there is a hash function $h_K \in \mathcal{H}$. Each $h_K : \mathcal{X} \to \mathcal{Y}$.

A pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$ is said to be valid under the key K if $h_K(x) = y$.

Let $\mathcal{F}^{\mathcal{X},Y}$ denote the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. Suppose that $|\mathcal{X}| = N$ and $|\mathcal{Y}| = M$. Then $|\mathcal{F}^{\mathcal{X},Y}| = M^N$. Any hash family $\mathcal{F} \subseteq \mathcal{F}^{\mathcal{X},Y}$ is termed an (N, M)-hash family.

MDC (Modification Dedection Code): An unkeyed hash function is a function $h_K : \mathcal{X} \to \mathcal{Y}$, where $|\mathcal{K}| = 1$.

Security of Cryptographic Hash Functions

A cryptographic hash function must be able to withstand all known types of cryptanalytic attacks. As a minimum, it must have the following properties:

1. Preimage resistance: Given a hash y it should be difficult to find any message x such that $y = h(x)$. This concept is related to that of one-way function.

2. Second preimage resistance: Given an input x_1 it should be difficult to find another input x_2 where $x_1 \neq x_2$ such that $h(x_1) = h(x_2)$. This property is sometimes referred to as weak collision resistance.

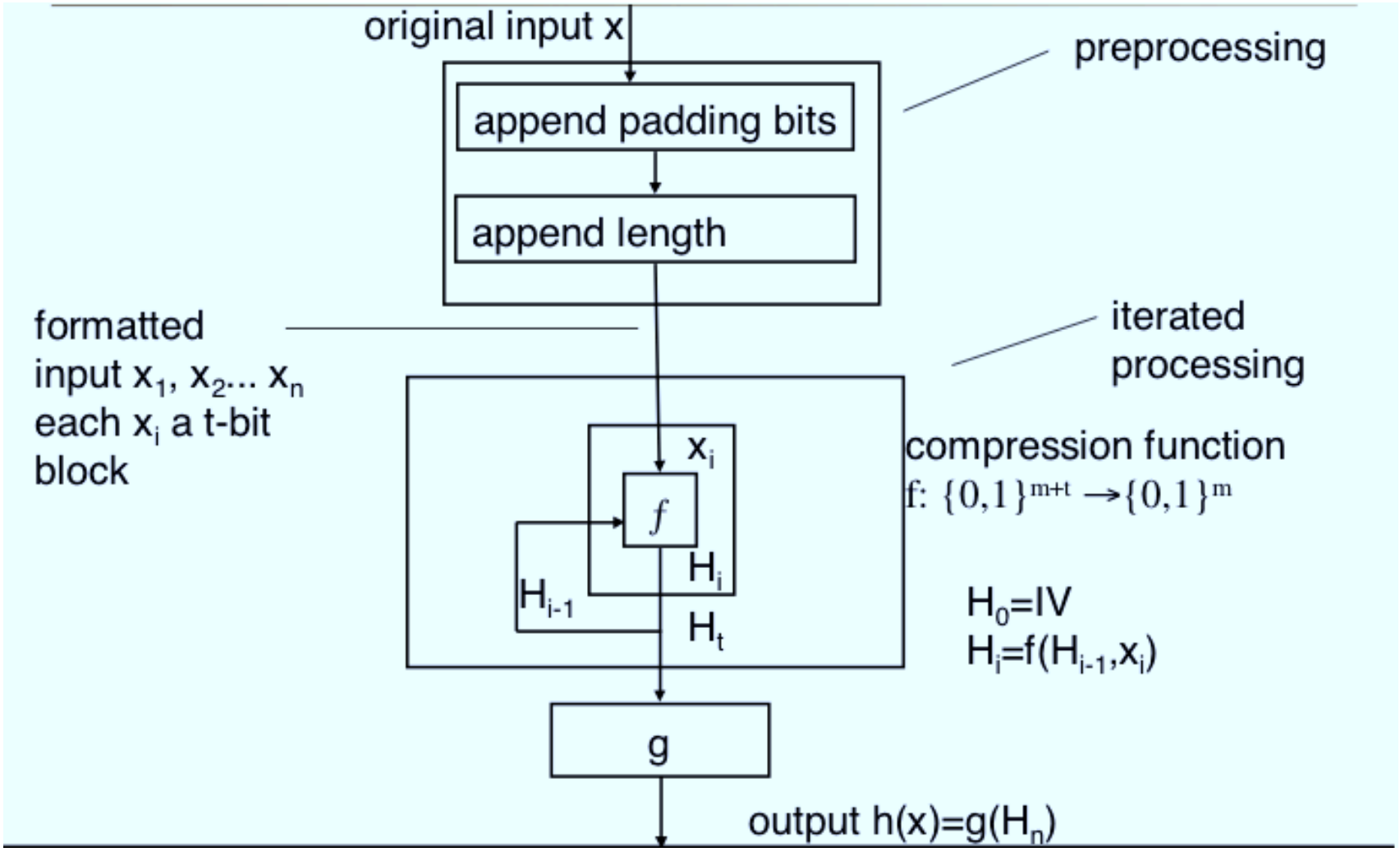
3. Collision resistance: It should be difficult to find two different messages x_1 and x_2 such that $h(x_1) = h(x_2)$. Such a pair is called a cryptographic hash collision. This property is sometimes referred to as strong collision resistance. It requires a hash value at least twice as long as that required for preimage-resistance, otherwise collisions may be found by a birthday attack.

Uses of hash functions

- Message authentication
- Software integrity
- One-time Passwords
- Digital signature
- Timestamping
- Certificate revocation management

Constructing Hash Function From Compression Functions

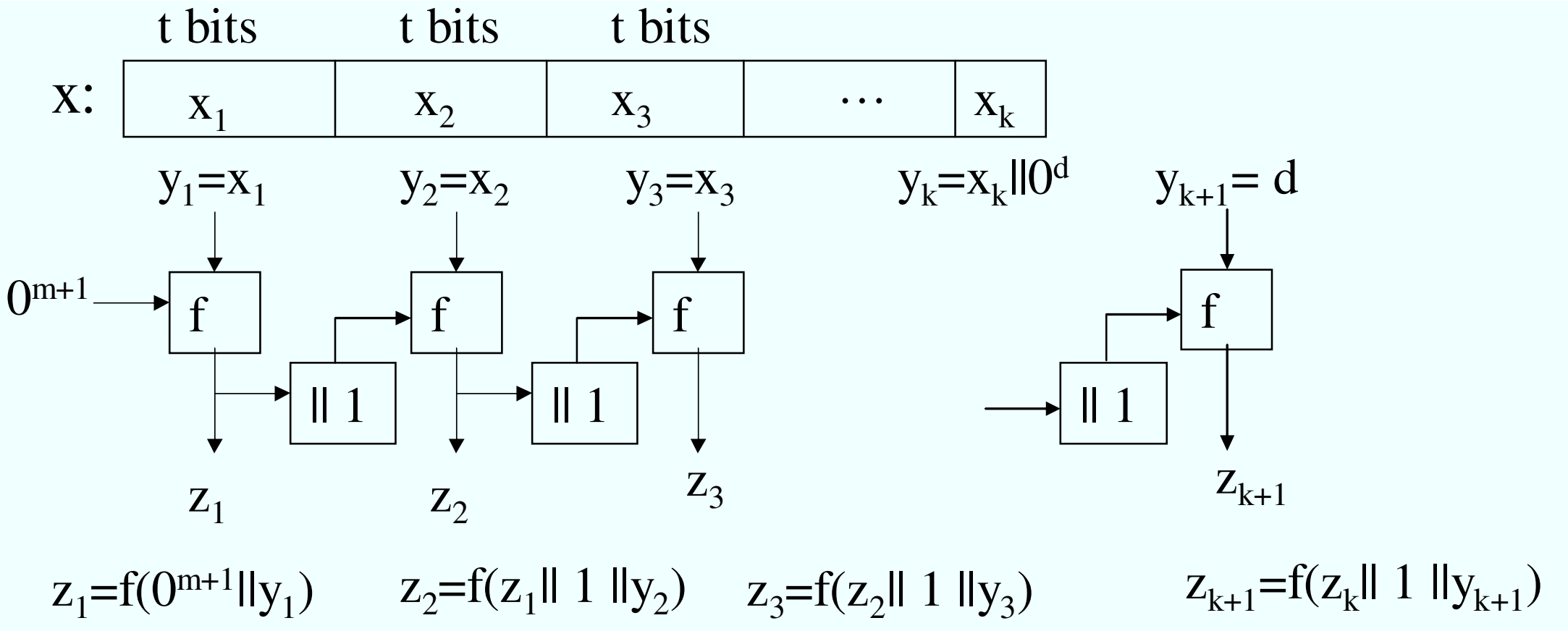
A compression function takes a fixed-length input string and output a shorter string $f : \{0,1\}^{m+t} \rightarrow \{0,1\}^{m}$.



The Merkle-Damgard Construction of Hash Functions

- Goal: construct a hash function $h : \{0,1\}^\star \to \{0,1\}^m$ from a compression function $f : \{0,1\}^{m+t+1} \to \{0,1\}^m$

- Given message x of arbitrary length



Example:

- Compression function: $f : \{0,1\}^{128+512+1} \rightarrow \{0,1\}^{128}$

- Message x has 1000 bits:
  - y_1 is first 512 bits of x
  - y_2 is last 488 bits of $x||0^{24}$
  - y_3 is $0^{480}||$ 32-bit binary representation of 24

- Iteration results
  - $z_1 = f\left(0^{129}||y_1\right)$ z_1 has 128 bits
  - $z_2 = f\left(z_1||1||y_2\right)$
  - $z_3 = f\left(z_2||1||y_3\right)$ z_3 is the message digest $h(x)$

Example:

- Suppose that message x' has 488 bits and $h(x) = h(x')$ (there is a collision for h):
  - y'_1 is $x'||0^{24}$
  - y'_2 is $0^{480}||$ 32-bit binary representation of 24
  - $z'_1 = f\left(0^{129}||y'_1\right)$ z'_1 has 128 bits
  - $z'_2 = f\left(z'_1||1||y'_2\right)$ z'_2 is $h(x')$

- Then $f\left(z'_1||1||y'_2\right) = f(z_2||1||y_3)$ and $y_3 = y'_2$
  - if $z'_1 \neq z_2$ then a collision is found for f
  - if $z'_1 = z_2$ then $f\left(0^{129}||y'_1\right) = f(z_1||1||y_2)$, there is also a collision for f

Security of the Merkle- Damgard Construction

If $f : \{0,1\}^{m+t+1} \rightarrow \{0,1\}^m$ is collision resistant, then the Merkle-Damgard construction $h : \{0,1\}^\star \rightarrow \{0,1\}^m$ is collision resistant.

SHA1 (Secure Hash Algorithm)

- SHA was designed by NIST and is the US federal standard for hash functions, specified in FIPS-180 (1993).

- SHA-1, revised version of SHA, specified in FIPS-180-1 (1995) use with Secure Hash Algorithm).

- It produces 160-bit hash values.

- NIST have issued a revision FIPS 180-2 that adds 3 additional hash algorithms: SHA-256, SHA-384, SHA-512, designed for compatibility with increased security provided by AES.

SHA3 Contest

NIST announced a public competition on Nov. 2, 2007 to develop a new cryptographic hash algorithm. The winning algorithm will be named "SHA-3", and will augment the hash algorithms currently specified in the Federal Information Processing Standard (FIPS) 180-3, Secure Hash Standard.

NIST received 64 entries by October 31, 2008; and selected 51 candidate algorithms to advance to the first round on December 10, 2008, and 14 to advance to the second round on July 24, 2009.

Based on the public feedback and internal reviews of the second-round candidates, NIST selected 5 SHA-3 finalists - BLAKE, Grøstl, JH, Keccak, and Skein to advance to the third (and final) round of the competition on December 9, 2010, which ended the second round of the competition.

A one-year public comment period is planned for the finalists. NIST also plans to host a final SHA-3 Candidate Conference in the spring of 2012 to discuss the public feedback on these candidates, and select the SHA-3 winner later in 2012.

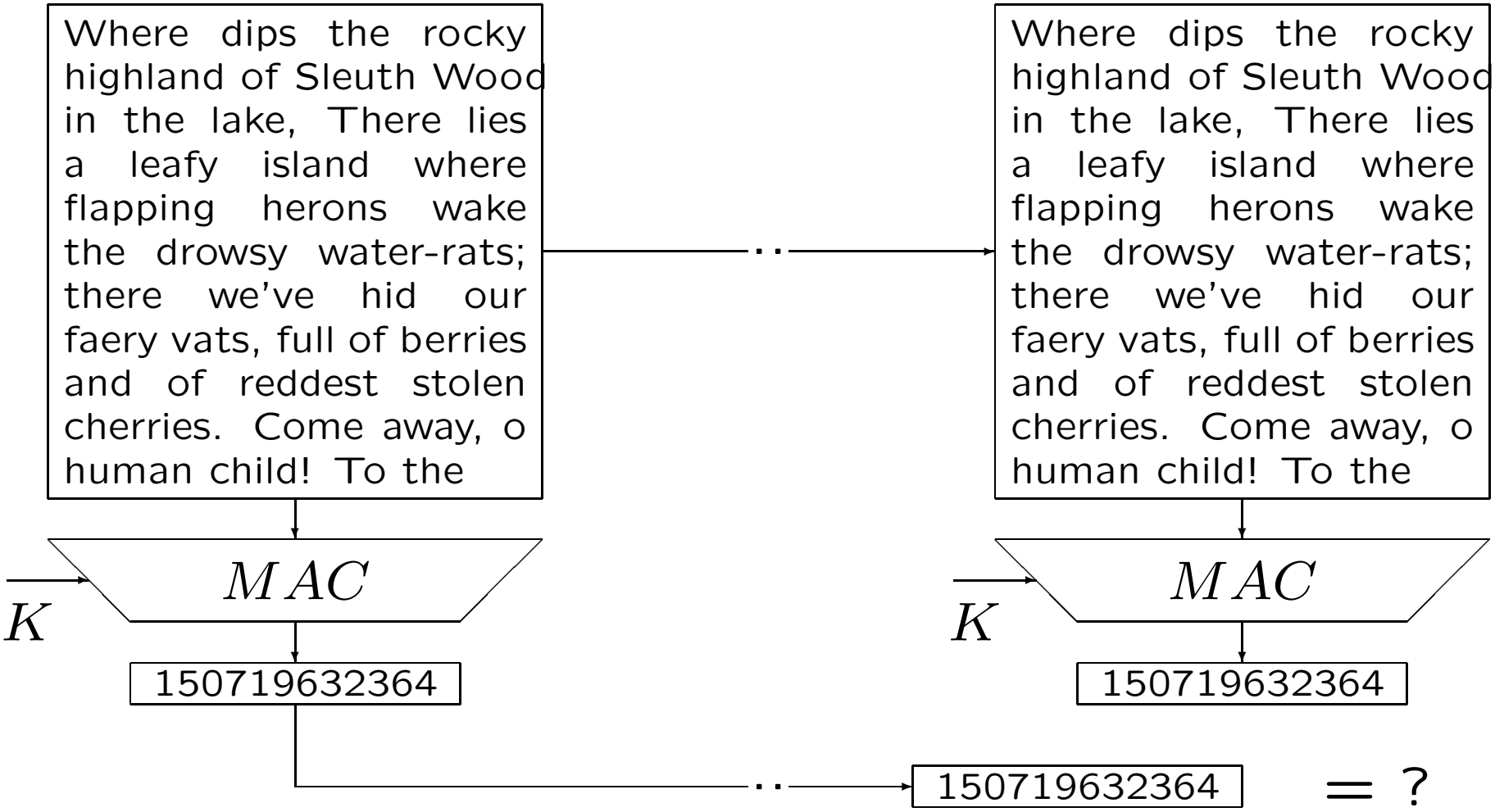
Further details of the competition are available at `http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo`.

Message Authentication Codes

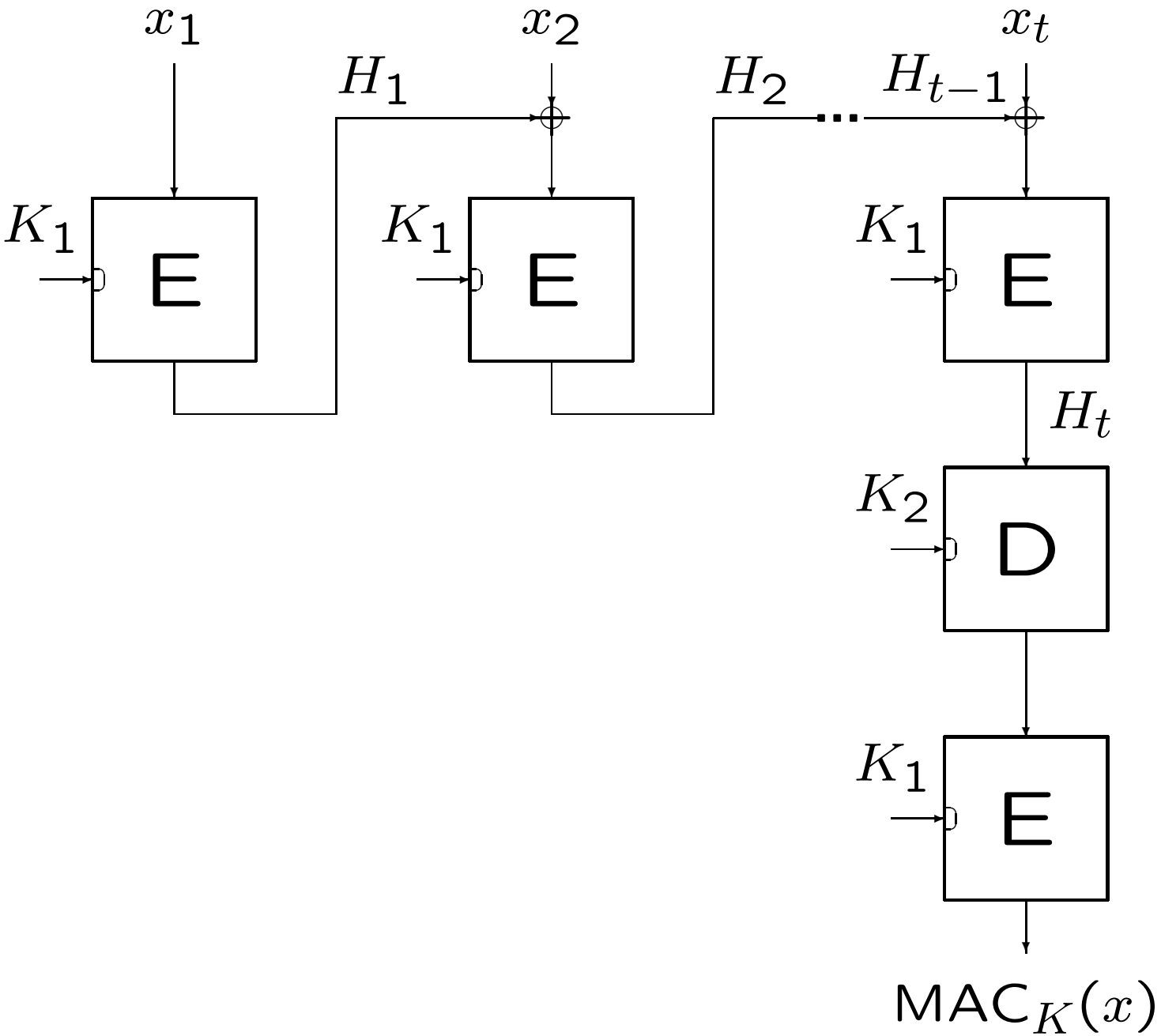
MAC (Message Authentication Code): Hash function with secret key

- hard to produce a forgery
- can only be generated and verified by someone who secret MAC-key
- do not use the same key for MAC and for encryption

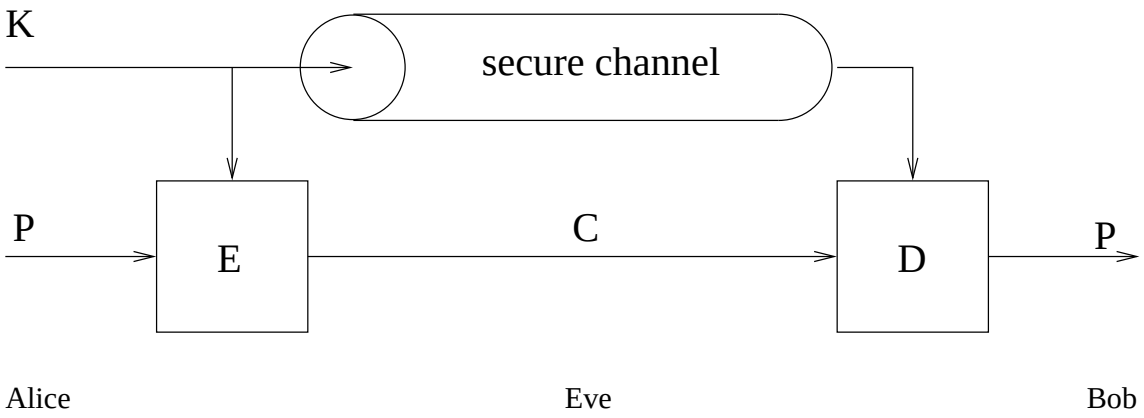
MAC = hash function with secret key



MAC based on block cipher: retail MAC



Symmetric Key Cryptography



Secret Key $\leftrightarrow$ Public Key

- key agreement

  How can 2 people who have never met share a key which is only known to these 2 people

- digital signature

  How can one be sure that a message comes from the sender who claims to have produced that message?

Public Key Cryptosystem

W. Diffie, M. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, Nov 1976, Volume: 22, Issue:6, page(s): 644 - 654.

1. for every $K \in \mathcal{K}$ e_K is the inverse of d_K,

2. for every $K \in \mathcal{K}$, $x \in \mathcal{P}$ and $y \in \mathcal{C}$ $e_K(x) = y$ and $d_K(y) = x$ are easy to compute.

3. for almost every $K \in \mathcal{K}$, each easily computed algorithm equivalent to d_K is computationally infeasible to derive from e_K,

4. for every $K \in \mathcal{K}$, it is feasible to compute inverse pairs e_K and d_K from K.

Because of the third property, a user's enciphering function e_K can be made public without compromising the security of his secret deciphering function d_K. The cryptographic system is therefore split into two parts, a family of enciphering transformations and a family of deciphering transformations in such a way that, given a member of one family, it is infeasible to find the corresponding member of the other.

Problem 1: Key-Agreement (1/3)

Diffie-Hellman Key Agreement Protocol

($f(X, Z)$: commutative one way function)

Alice		*Bob*
$Y_A = f(X_A, Z)$		
	$\overset{Y_A}{\longrightarrow}$	
		$Y_B = f(X_B, Z)$
	$\overset{Y_B}{\longleftarrow}$	
$K_{AB} = f(X_A, Y_B) = f(X_A, f(X_B, Z))$		$K_{BA} = f(X_B, f(X_A, Z))$

Key-Agreement (2/3)

Modular Exponentiation

- given α and a prime p with $\alpha \in [1, p-1]$
- $w = \alpha^x \bmod p$ can be computed efficiently (square and multiply)

Inverse operation (*discrete logarithm*)

- given α, p and w, find x such that

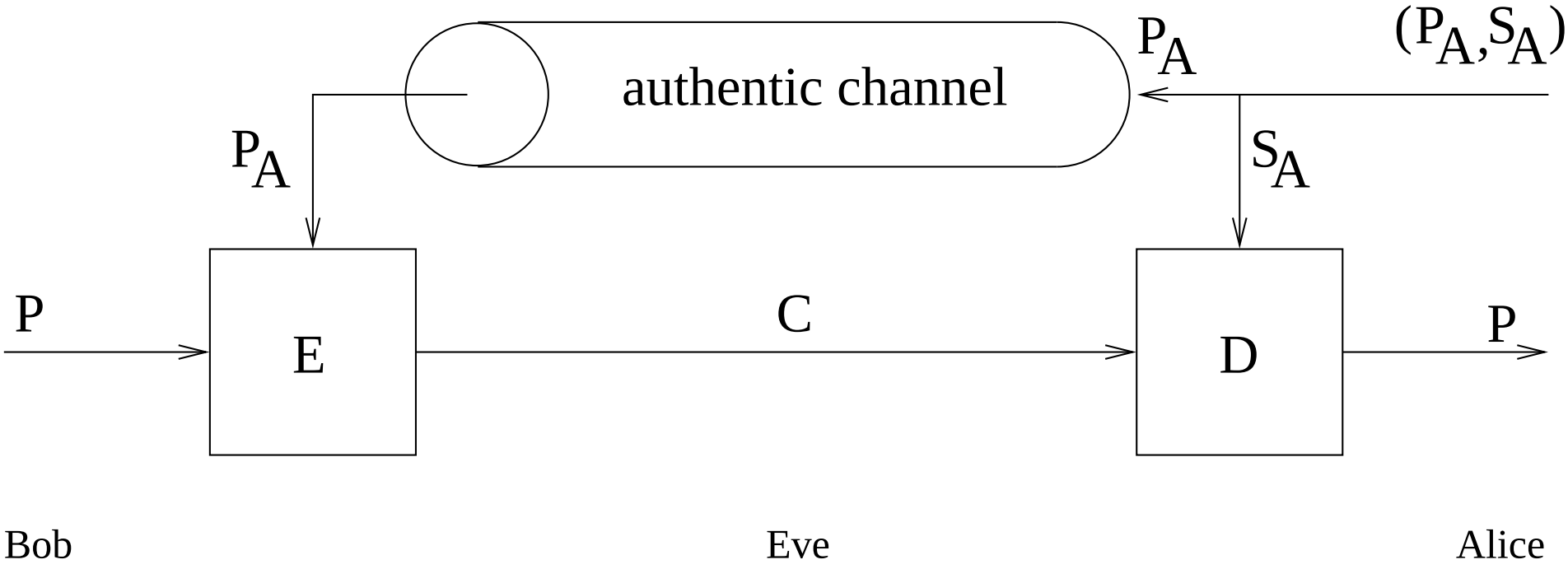
$$\alpha^x \bmod p \equiv w$$

Key-Agreement (3/3)

- $p = 37$: the integers from 0 to 36 form a field with $+$ and $\times$ mod 37
- $\alpha = 2$ is a generator of the non-zero elements: powers of 2 generate all non-zero elements: $2^0 = 1$, $2^1 = 2$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 27$, $2^7 = 17$, ..., $2^{36} = 1$
- $X_A = 10 \Rightarrow Y_A = 2^{10} \bmod 37 = 25$
- $X_B = 13 \Rightarrow Y_B = 2^{13} \bmod 37 = 15$
- $K_{AB} = (Y_B)^{X_A} = 15^{10} \bmod 37 = 15^{8+2} \bmod 37 = 7 \times 3 \bmod 37 = 21$
- $K_{BA} = (Y_A)^{X_B} = 25^{13} \bmod 37 = 25^{8+4+1} \bmod 37 = 34 \times 16 \times 25 \bmod 37 = 21$
- $K_{AB} = K_{BA} = 21$

Problem 2: Public-key cryptography (1/3)

(trapdoor one-way functions)



Public-key cryptography (2/3)

RSA public-key algorithm

trapdoor one-way function:

- given x: "easy" to compute $f(x)$
- given $f(x)$: "hard" to compute x
- given $f(x)$ and the trapdoor information: finding x is "easy"

given two large primes p and q and a public key (e, n)

$n = p \times q$ (factoring n is hard)

$f(x) = x^e \bmod n$ is a trapdoor one-way function

trapdoor information (p, q) allows to find a private key (d, n) such that

$(x^e)^d = (x^e)^{1/e} = x \bmod n$

Public-key cryptography (3/3)

RSA public-key algorithm (2): detail

key generation:

choose two primes p and q

$n = p \times q$, $\phi(n) = (p-1)(q-1)$

choose e prime w.r.t. $\phi(n)$

compute $d = e^{-1} \bmod \phi(n)$

public key $= (e, n)$

private key $= (d, n)$ or (p, q)

encrytion: $c = m^e \bmod n$

decrytion: $m = c^d \bmod n$

Modular Exponantiation

Algorithm 3 Square and Multiply - left to right

Require: $N = (n_{k-1}, \ldots, n_1, n_0)_2$, $E = (e_{k-1}, \ldots, e_1, e_0)_2$, $M = (m_{k-1}, \ldots, m_1, m_0)_2$

Ensure: $C = M^E \bmod N$

1: $C = 1$
2: **for** i from $k-1$ downto 0 **do**
3: $\quad C = C^2 \bmod N$
4: $\quad$ **if** $E_i = 1$ **then**
5: $\quad\quad C = CM \bmod N$
6: $\quad$ **end if**
7: **end for**

Attacks on the RSA Cryptosystem

Although 35 years of research have led to a number of fascinating attacks, none of them is devastating. They mostly illustrate the dangers of improper use of RSA. Indeed, securely implementing RSA is a nontrivial task.

Factoring Large Integers

We refer to factoring the modulus as a brute-force attack on RSA.

Factoring algorithms running time

Pollard's Rho algorithm	$O\left(\sqrt{p}\right)$
Pollard's $p-1$ algorithm	$O\left(p'\right)$ where p' is the largest prime factor of $p-1$
Pollard' s $p+1$ algorithm	$O\left(p'\right)$ where p' is the largest prime factor of $p+1$
Elliptic Curve method (ECM)	$O\left(e^{(1+o(1))(2\ln p \ln\ln p)^{1/2}}\right)$
Quadratic Sieve (Q.S.)	$O\left(e^{(1+o(1))(\ln N \ln\ln N)^{1/2}}\right)$
Number Filed Sieve (NFS)	$O\left(e^{(1.92+o(1))(\ln N)^{1/3}(\ln\ln N)^{2/3}}\right)$

Our objective is to survey attacks on RSA that decrypt messages without directly factoring the RSA modulus N.

Is breaking RSA as hard as factoring?

Chinese Remainder Theorem

The following problem was posed by Sunzi (4th century AD) in the book Sunzi Suanjing:

when a number is

repeatedly divided by 3, the remainder is 2;
by 5 the remainder is 3;
and by 7 the remainder is 2.

What will be the number?

Oystein Ore mentions another puzzle with a dramatic element from Brahma-Sphuta-Siddhanta (Brahma's Correct System) by Brahmagupta (born 598 AD):

An old woman goes to market and a horse steps on her basket and crashes the eggs.

The rider offers to pay for the damages and asks her how many eggs she had brought.

She does not remember the exact number, but when she had taken them out two at a time, there was one egg left. The same happened when she picked them out three, four, five, and six at a time, but when she took them seven at a time they came out even. What is the smallest number of eggs she could have had?

Involves a situation like the following: we are asked to find an integer x which gives a remainder of 4 when divided by 5, a remainder of 7 when divided by 8, and a remainder of 3 when divided by 9.

In other words, we want x to satisfy the following congruences.

$x \equiv 4 \bmod 5$, $x \equiv 7 \bmod 8$, $x \equiv 3 \bmod 9$

There can be any number of moduluses, but no two of them should have any factor in common. Otherwise the existence of a solution cannot be guaranteed.

The method for solving this set of three simultaneous congruences is to reduce it to three separate problems whose answers may be added together to get a solution to the original problem.

To understand this, think about why

144 + 135 + 120 will be a solution to the simultaneous congruences.

144 gives a reminder of 4 when divided by 5. On the other hand, 135 and 120 are multiples of 5, so adding them doesn't change this reminder.

$144 + 135 + 120 \equiv 144 \bmod 5 \equiv 4 \bmod 5$

135 gives a reminder of 7 when divided by 8. On the other hand, 144 and 120 are multiples of 8, so adding them on doesn't change this reminder.

$144 + 135 + 120 \equiv 135 \bmod 8 \equiv 7 \bmod 8$

120 gives a reminder of 3 when divided by 9. But 144 and 135 are multiples of 9, so adding them in doesn't affect this remainder.

$144 + 135 + 120 \equiv 120 \bmod 9 \equiv 3 \bmod 9$

Therefore 399, which is the sum of 144, 135, and 120, satisfies all three of the congruences.

Having now seen why 399 is a valid solution, we can also partly see the process by which it was created. We found it as the sum of three numbers.

The first number, 144, gives the right remainder when divided by 5 and is also a multiple of 8 and of 9.

The second number, 135, is a multiple of 5 and of 9 and gives the correct remainder when divided by 8.

The third number, 120, is congruent to 3 module 9 and is a multiple of both 5 and 8.

So where did we get these three numbers?

To start with, taking the last two of the three moduli 5, 8, and 9, compute $8 \times 9 = 72$. We look for a multiple of 72 which satisfies the first congruence.

$72 \times 2 = 144$, and $144 \equiv 4 \bmod 5$.

$5 \times 9 = 45$. We look for a multiple of 45 which satisfies the second congruence. We find (by trial and error) that

$1 \times 45 = 45 \equiv 5 \bmod 8$, $2 \times 45 = 90 \equiv 2 \bmod 8$, $3 \times 45 = 135 \equiv 7 \bmod 8$

$5 \times 8 = 40$. We look for a multiple of 40 which is congruent to 3 module 9.

$40 \equiv 4 \bmod 9$, $80 \equiv 8 \bmod 8$, $120 \equiv 3 \bmod 9$

Now the required answer is the sum $144 + 135 + 120$, namely 399.

Consider another example. Look for a number x satisfying the following congruences.

$x \equiv 1 \bmod 2$, $x \equiv 2 \bmod 3$, $x \equiv 3 \bmod 5$, $x \equiv 1 \bmod 7$

$3 \times 5 \times 7 = 105$. We look for a multiple of 105 which is congruent to 1 modulo 2. We can choose 105 itself, since it is odd.

$2 \times 5 \times 7 = 70$. We look for a multiple of 70 which is congruent to 2 modulo 3. $70 \equiv 1 \bmod 3 \Rightarrow 2 \times 70 \equiv 2 \times 1 \equiv 2 \bmod 3$. We can choose 140.

$2 \times 3 \times 7 = 42$. We look for a multiple of 42 which is congruent to 3 modulo 5. $42 \equiv 2 \bmod 5$. $4 \times 42 \equiv 4 \times 2 = 8 \equiv 3 \bmod 5$. We use 168.

$2 \times 3 \times 5 = 30$. We want a multiple of 30 which is congruent to 1 modulo 7. $30 \equiv 2 \bmod 7$. $4 \times 30 \equiv 4 \times 2 = 8 \equiv 1 \bmod 7$. We use 120.

Adding the four numbers we've found together, we get a solution of $105 + 140 + 168 + 120 = 533$

One solution to a system of congruences

$$\begin{aligned} x &\equiv a_1 \bmod m_1 \\ x &\equiv a_2 \bmod m_2 \\ &\dots \\ x &\equiv a_n \bmod m_n \end{aligned}$$

with the m_i mutually prime to each other can be found by adding together n numbers. The i th of these numbers should be congruent to a_i modulo m_i and it should be a multiple of all the other moduli m_k.

If a and m are relatively prime, then the congruence $az \equiv b \bmod m$ is always solvable for z, no matter what b is.

To see why this has to be true, consider, for instance, the first 5 multiples of 42 and reduce modulo 5.

$$\begin{aligned} 42 &\equiv 2 \bmod 5 \\ 0 \times 42 &\equiv 0 \bmod 5 \\ 1 &\equiv 2 \bmod 5 \\ 2 \times 42 &\equiv 4 \bmod 5 \\ 3 \times 42 &\equiv 1 \bmod 5 \\ 4 \times 42 &\equiv 3 \bmod 5 \end{aligned}$$

Notice that one the right hand side, every number from 0 to 4 occurs, showing that a congruence $42z \equiv b \bmod 5$ can always be solved, no matter what b is.

This is not a coincidence, but is a consequence of the fact that 42 has no factor in common with 5. If any of the five numbers from 0 to 4 had been missing on the right-hand side of the five congruences listed, then at least one right-hand side would have to be repeated. But, given the fact that 42 has no factors in common with 5, this would not be possible.

Broadcast Attack

Think that Alice wants to send the same message, x to Bob, Bill and Bart, who have all the same public key, e, but different modulus, n_1, n_2, n_3. Can Eve find x without knowing the private keys?

Yes, she can by using CRT!

$$x^e \equiv a_1 \bmod n_1$$
$$x^e \equiv a_2 \bmod n_2$$
$$x^e \equiv a_3 \bmod n_3$$

Common Modulus

To avoid generating a different modulus $N = pq$ for each user, one may wish to fix N once and for all. The same N is used by all users. A trusted central authority could provide user i with a unique pair (e_i, d_i) from which user i forms a public key $< N, e_i >$ and a secret key $< N, d_i >$.

Fact 1: Let $\langle N, e \rangle$ be an RSA public key. Given the private key d, one can efficiently factor the modulus $N = pq$. Conversely, given the factorization of N, one can efficiently recover d.

By Fact 1 Bob can use his own exponents e_b, d_b to factor the modulus N. Once N is factored Bob can recover Alices private key d_a from her public key e_a. This observation, due to Simmons, shows that an RSA modulus should never be used by more than one entity.

Exposing the private key d and factoring N are equivalent. Hence there is no point in hiding the factorization of N from any party who knows d.

Blinding

Let $< N, d >$ be Bobs private key and $< N, e >$ his corresponding public key. Suppose Marvin wants Bobs signature on a message $M \in Z_N^*$. Being no fool, Bob refuses to sign M.

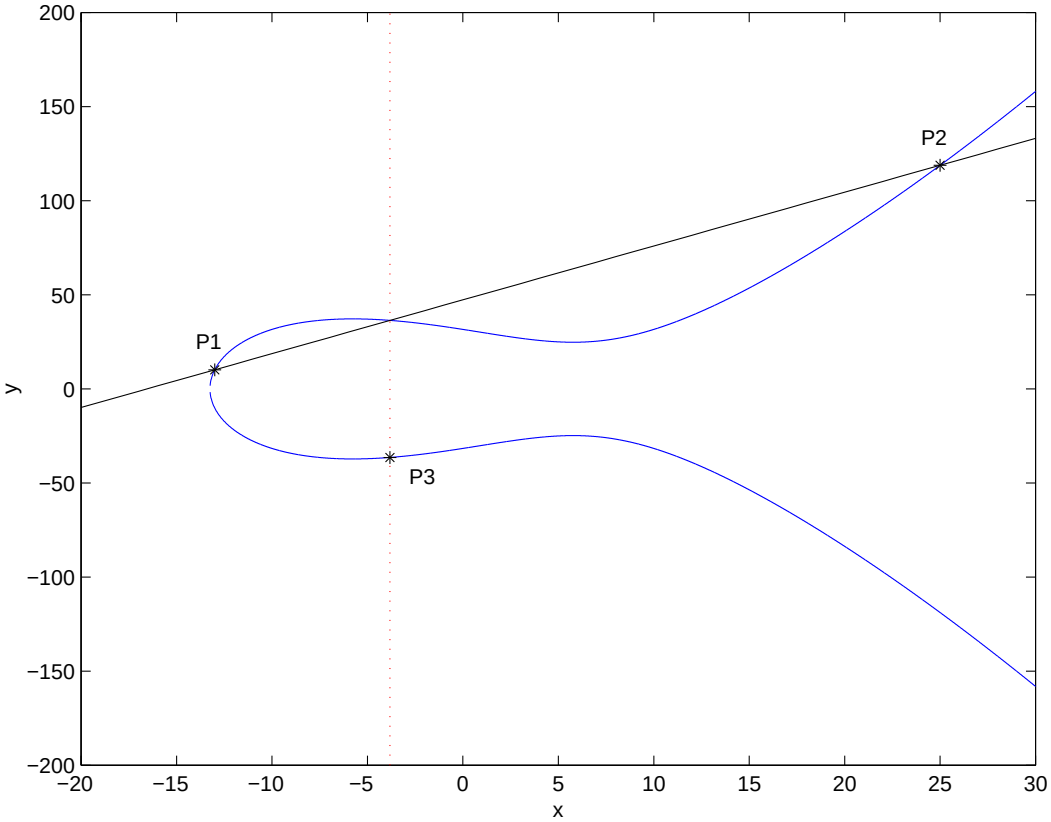
Marvin can try the following: he picks a random $r \in Z_N^*$ and sets $M' = r^e M \bmod N$. He then asks Bob to sign the random message M'. Bob may be willing to provide his signature S' on the innocent-looking M'. Marvin now simply computes $S = S'/r \bmod N$ and obtains Bobs signature S on the original M. Indeed,

$$S = \frac{S'}{r} = \frac{M'^d}{r} = \frac{r^{ed} M^d}{r} = \frac{r M^d}{r} = M^d$$

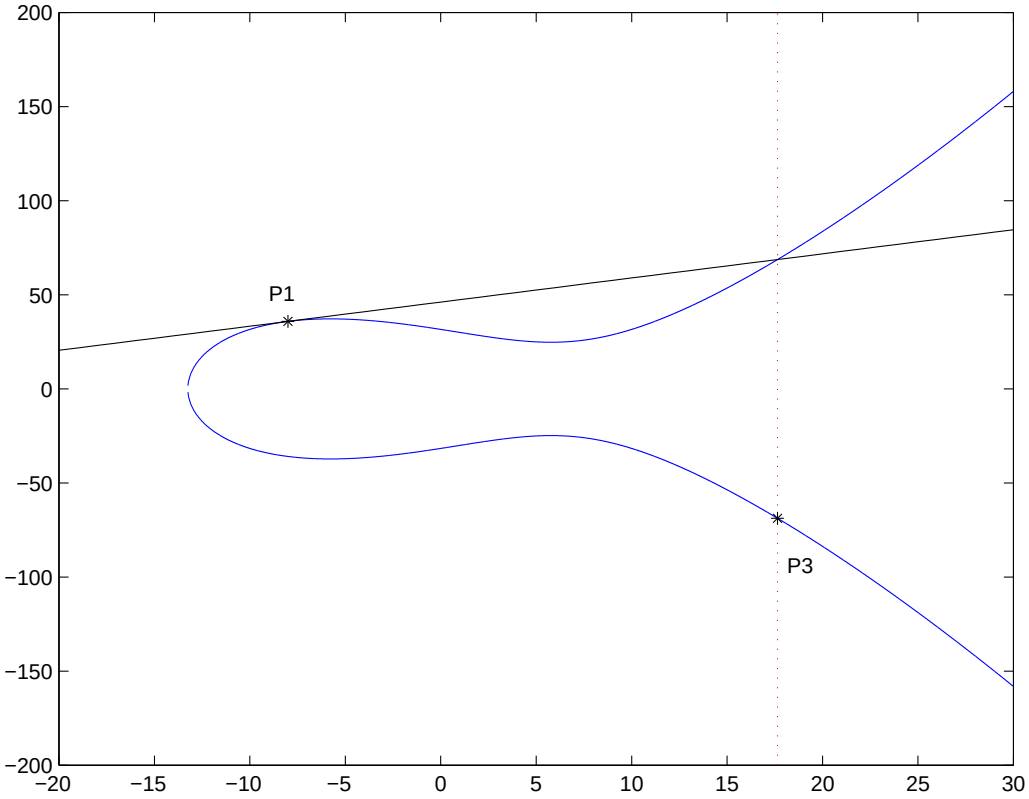
This technique, called blinding, enables Marvin to obtain a valid signature on a message of his choice by asking Bob to sign a random "blinded" message. Bob has no information as to what message he is actually signing.

Elliptic Curve Group over $\mathbb{R}$

Definition: set of the solutions of Weierstrass equation $E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ over a field and the point at infinity $\mathcal{O}$.



Adding two points



Doubling a point

Elliptic Curve Point Addition and Doubling over $GF(p)$ $p > 3$

$$E : y^2 = x^3 + ax + b$$

$$P_1 = (x_1, y_1),\ P_2 = (x_2, y_2) \text{ and } P_3 = (x_3, y_3) = P_1 + P_2$$

$$x_3 = \lambda^2 - x_1 - x_2$$
$$y_3 = \lambda(x_1 - x_3) - y_1$$

$$\lambda = \begin{cases} (y_2 - y_1)\,(x_2 - x_1)^{-1} & \text{if } P_1 \neq P_2 \\ \left(3x_1^2 + a\right)(2y_1)^{-1} & \text{if } P_1 = P_2 \end{cases}$$

projective coordinates are used to get rid of modular multiplicative inversion

Elliptic Curve Point Multiplication

$$[k]P = \underbrace{P + P + \cdots + P}_{k}$$

Require: EC point $P = (x, y)$, integer k, $0 < k < M$,
$k = (k_{l-1}, k_{l-2}, \cdots, k_0)_2$, $k_{l-1} = 1$ and M

Ensure: $Q = [k]P = (x', y')$

$Q \leftarrow P$

for i from $l - 2$ downto 0 **do**

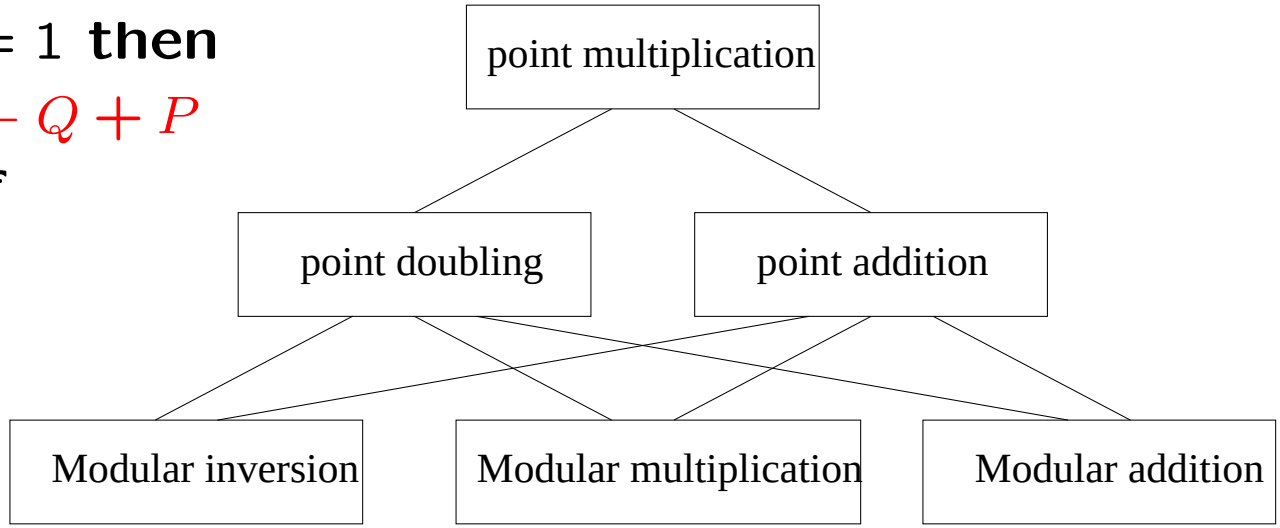
$Q \leftarrow 2Q$

if $k_i = 1$ **then**

$Q \leftarrow Q + P$

end if

end for



Elliptic Curve Point Addition and Doubling

Require: $P_1 = (x, y, 1, a)$, $P_2 = (X_2, Y_2, Z_2, aZ_2^4)$
Ensure: $P_1 + P_2 = P_3 = (X_3, Y_3, Z_3, aZ_3^4)$

1.	$T_1 \leftarrow Z_2^2$	
2.	$T_2 \leftarrow xT_1$	
3.	$T_1 \leftarrow T_1 Z_2$	$T_3 \leftarrow X_2 - T_2$
4.	$T_1 \leftarrow yT_1$	
5.	$T_4 \leftarrow T_3^2$	$T_5 \leftarrow Y_2 - T_1$
6.	$T_2 \leftarrow T_2 T_4$	
7.	$T_4 \leftarrow T_4 T_3$	$T_6 \leftarrow 2T_2$
8.	$Z_3 \leftarrow Z_2 T_3$	$T_6 \leftarrow T_4 + T_6$
9.	$T_3 \leftarrow T_5^2$	
10.	$T_1 \leftarrow T_1 T_4$	$X_3 \leftarrow T_3 - T_6$
11.	$aZ_3^4 \leftarrow Z_3^2$	$T_2 \leftarrow T_2 - X_3$
12.	$T_3 \leftarrow T_5 T_2$	
13.	$aZ_3^4 \leftarrow \left(aZ_3^4\right)^2$	$Y_3 \leftarrow T_3 - T_1$
14.	$aZ_3^4 \leftarrow a\left(aZ_3^4\right)$	

latency $= 14T_{MM}$

Require: $P_1 = (X_1, Y_1, Z_1, aZ_1^4)$
Ensure: $2P_1 = P_3 = (X_3, Y_3, Z_3, aZ_3^4)$

1.	$T_1 \leftarrow Y_1^2$	$T_2 \leftarrow 2X_1$
2.	$T_3 \leftarrow T_1^2$	$T_2 \leftarrow 2T_2$
3.	$T_1 \leftarrow T_2 T_1$	$T_3 \leftarrow 2T_3$
4.	$T_2 \leftarrow X_1^2$	$T_3 \leftarrow 2T_3$
5.	$T_4 \leftarrow Y_1 Z_1$	$T_3 \leftarrow 2T_3$
6.	$T_5 \leftarrow T_3\left(aZ_1^4\right)$	$T_6 \leftarrow 2T_2$
7.		$T_2 \leftarrow T_6 + T_2$
8.		$T_2 \leftarrow T_2 + \left(aZ_1^4\right)$
9.	$T_6 \leftarrow T_2^2$	$Z_3 \leftarrow 2T_4$
10.		$T_4 \leftarrow 2T_1$
11.		$X_3 \leftarrow T_6 - T_4$
12.		$T_1 \leftarrow T_1 - X_3$
13.	$T_2 \leftarrow T_2 T_1$	$aZ_3^4 \leftarrow 2T_5$
14.		$Y_3 \leftarrow T_2 - T_3$

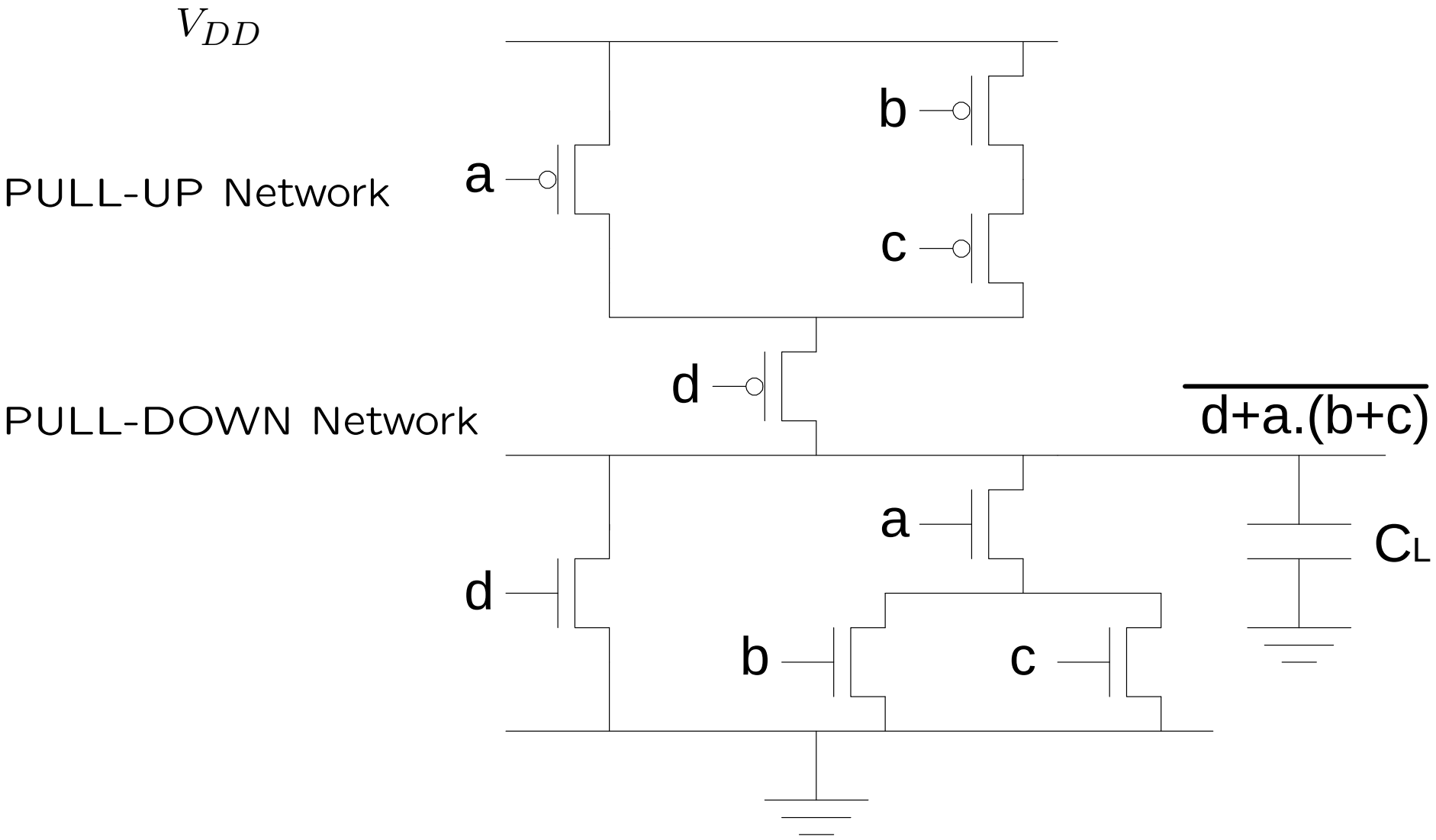
latency $= 8T_{MM} + 6T_{MAS}$

Modular Addition, Subtraction Circuit over $GF(p)$

Require: M, $0 \leq A < M$,
$0 \leq B < M$
Ensure: $C = A + B \bmod M$
$C' = A + B$
$C'' = C' - M$
if $C'' < 0$ **then**
$C = C'$
else
$C = C''$
end if

Require: M, $0 \leq A < M$,
$0 \leq B < M$
Ensure: $C = A - B \bmod M$
$C' = A - B$
$C'' = C' + M$
if $C' < 0$ **then**
$C = C''$
else
$C = C'$
end if

Power-Analysis Attacks: Why do they work?



Dynamic power consumption is mainly due to the charge and discharge of the load capacitance C_L.

Types of Power-Analysis Attacks

Simple Power Analysis (SPA) Attacks:

- every instruction $\Longrightarrow$ unique power-consumption trace
- one measurement

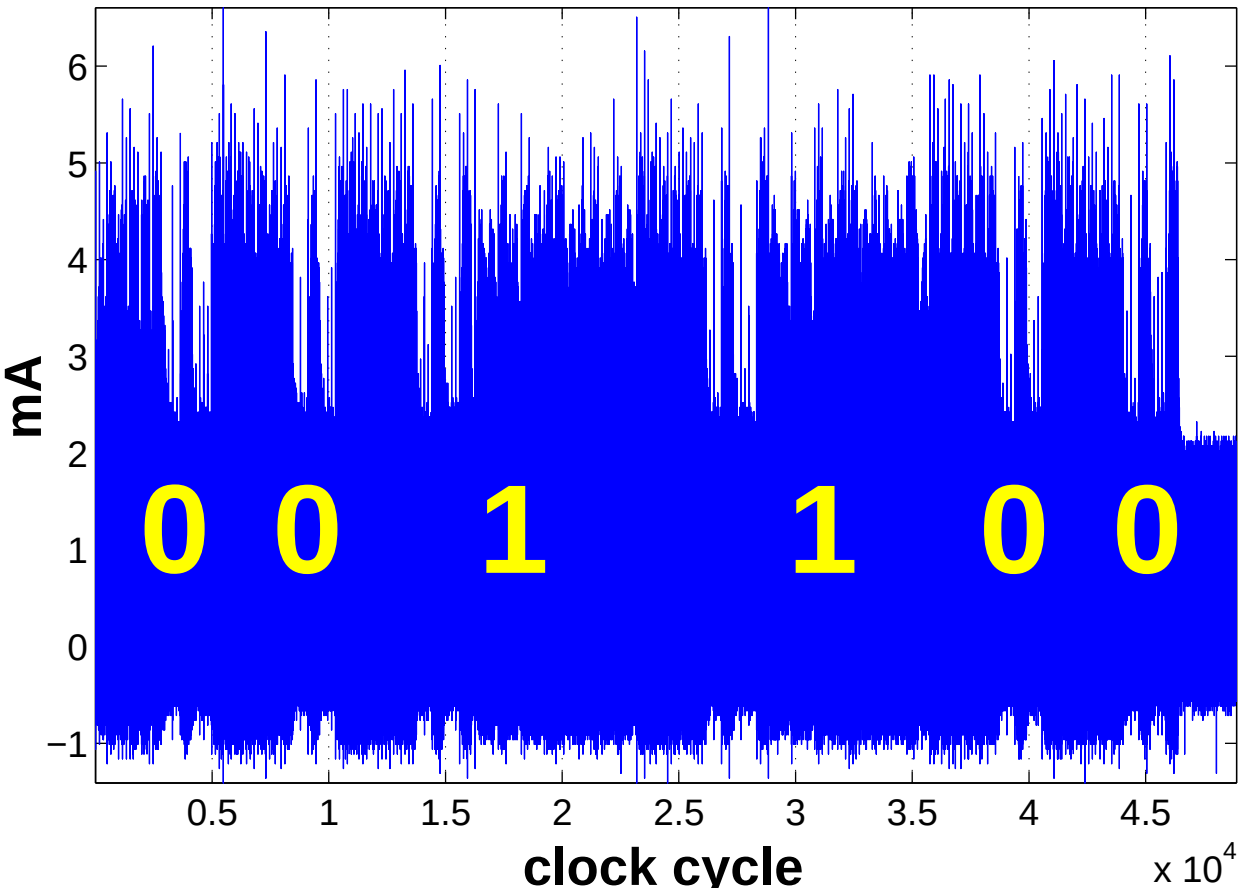
Differential Power Analysis (DPA) Attacks:

- many measurements
- statistical analysis used

SPA Attack on Elliptic Curve Point Multiplication

Require: EC point $P = (x, y)$, integer k, $k = (k_{l-1}, k_{l-2}, \cdots, k_0)_2$, $k_{l-1} = 1$
Ensure: $Q = (x', y')$

$Q \leftarrow P$
for i from $l - 2$ downto 0 **do**
  $Q \leftarrow 2Q$
  if $k_i = 1$ **then**
    $Q \leftarrow Q + P$
  end if
end for



The key used during this measurement is 1001100.

Countermeasure for SPA Attack

Require: EC point $P = (x, y)$, integer k, $k = (k_{l-1}, k_{l-2}, \cdots, k_0)_2$, $k_{l-1} = 1$

Ensure: $Q = (x', y')$

$Q \leftarrow P$

for i from $l - 2$ downto 0 **do**

$Q_1 \leftarrow 2Q$

$Q_2 \leftarrow Q_1 + P$

if $k_i = 1$ **then**

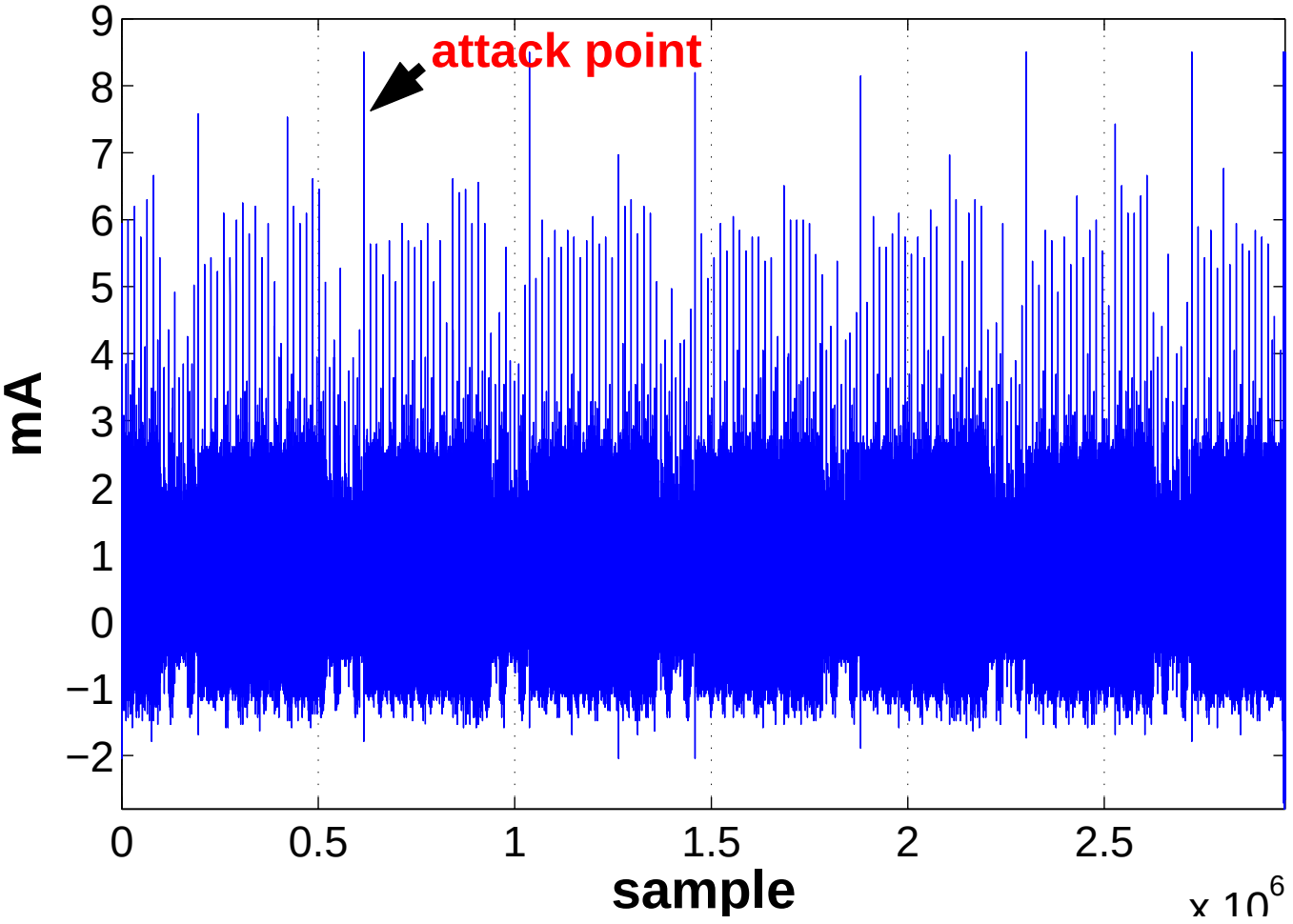
$Q \leftarrow Q_2$

else

$Q \leftarrow Q_1$

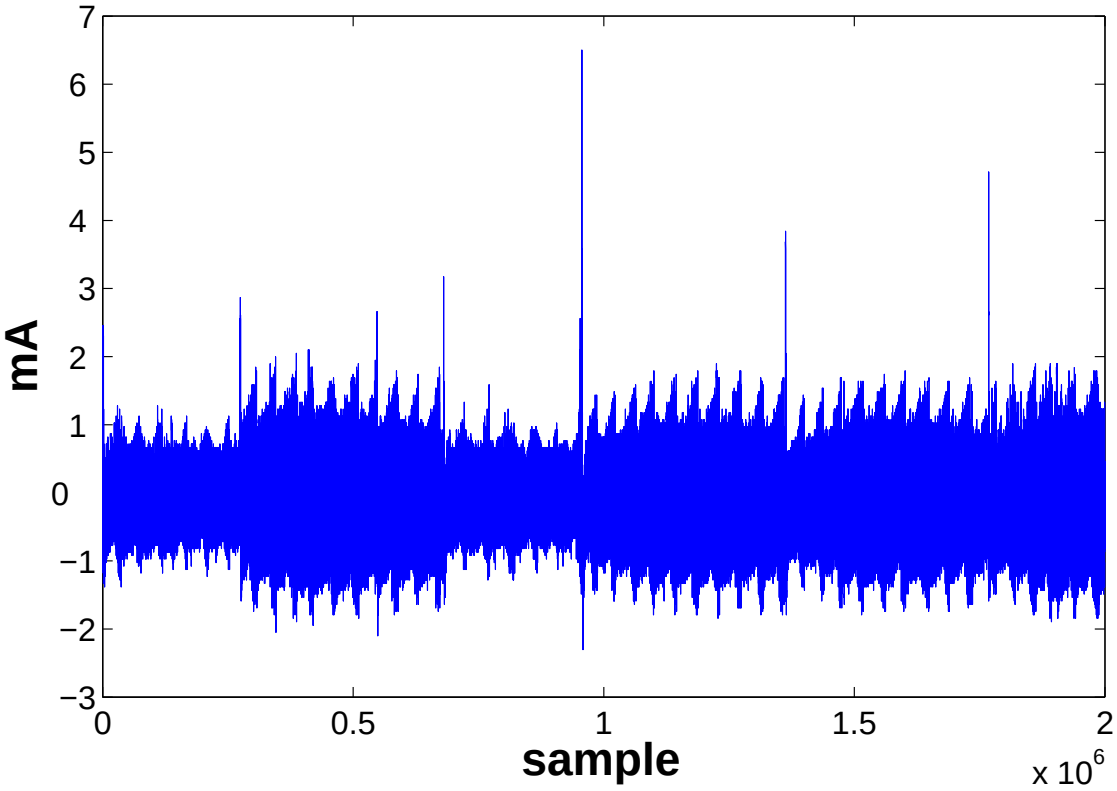
end if

end for

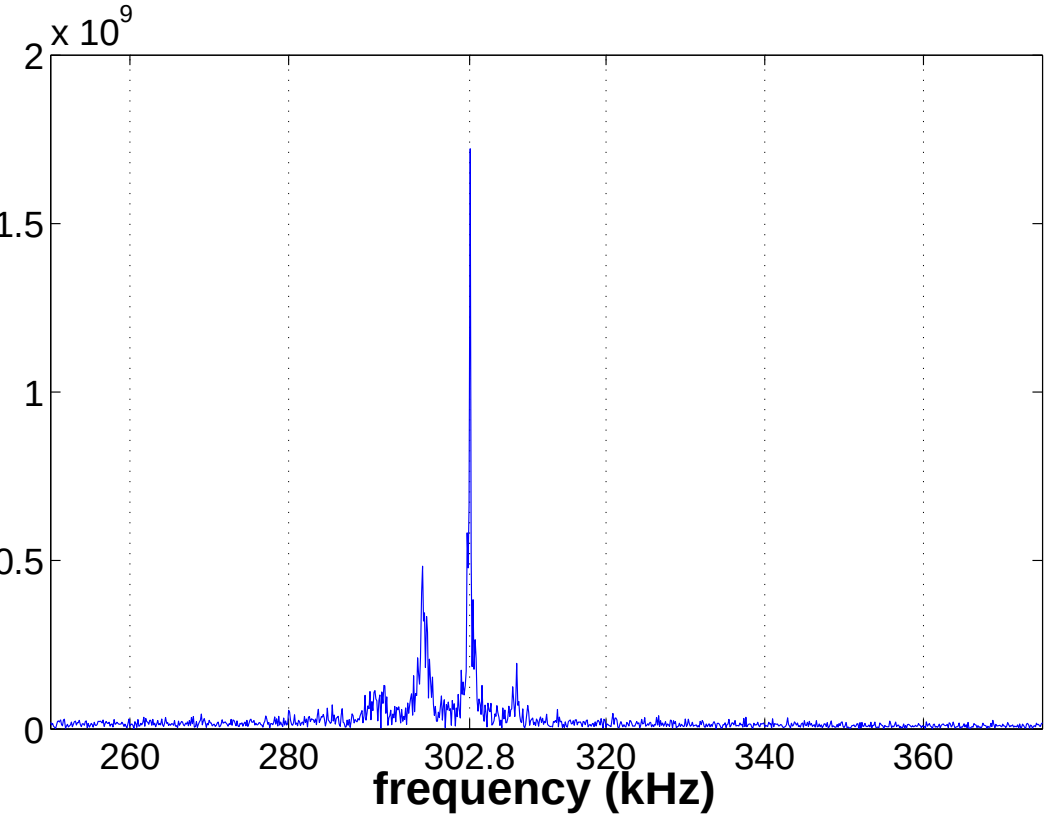


Current Consumption Measurement for DPA Attack

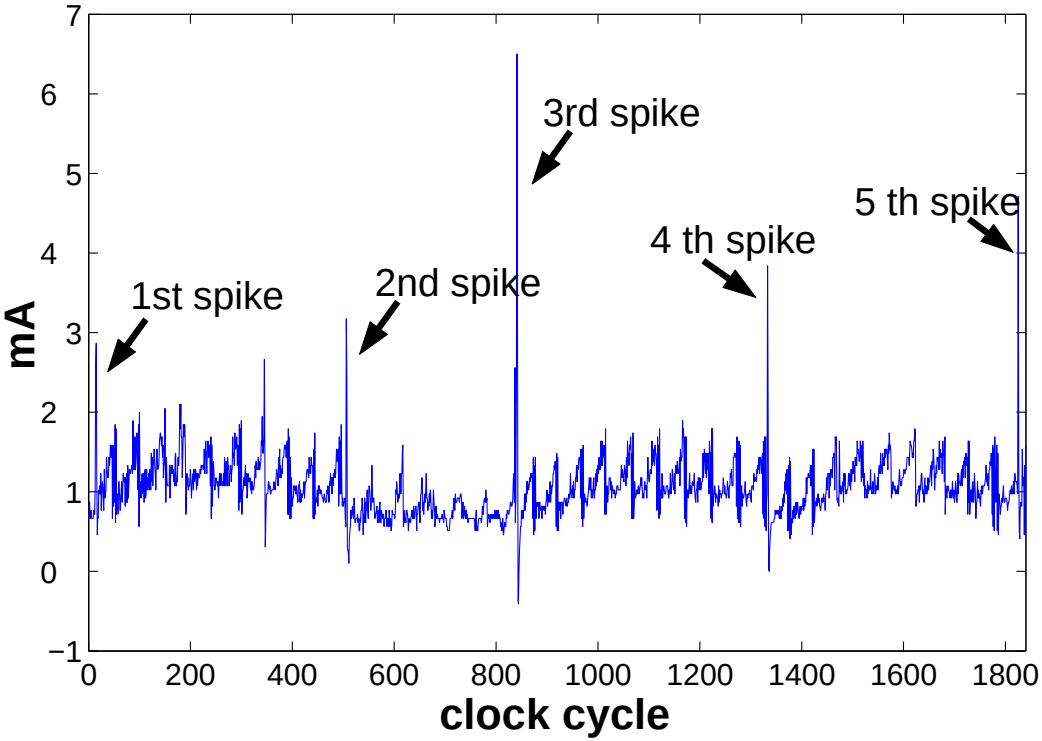
- Data length: 2400 clock cycles around the 2nd update of Q_1.
- Clock frequency: 300 kHz.
- Sampling frequency: 250 MHz.



Pre-Processing



Discrete Fourier transform
between 250 kHz and 375 kHz
Clock frequency : 302.8 kHz



maximum in every clock cycle

Correlation Analysis

1. hypothetical model $\Longrightarrow$ predict side-channel output for N inputs.

   Prediction is the number of bits changed from 0 to 1 from X_i to X_{i+1}

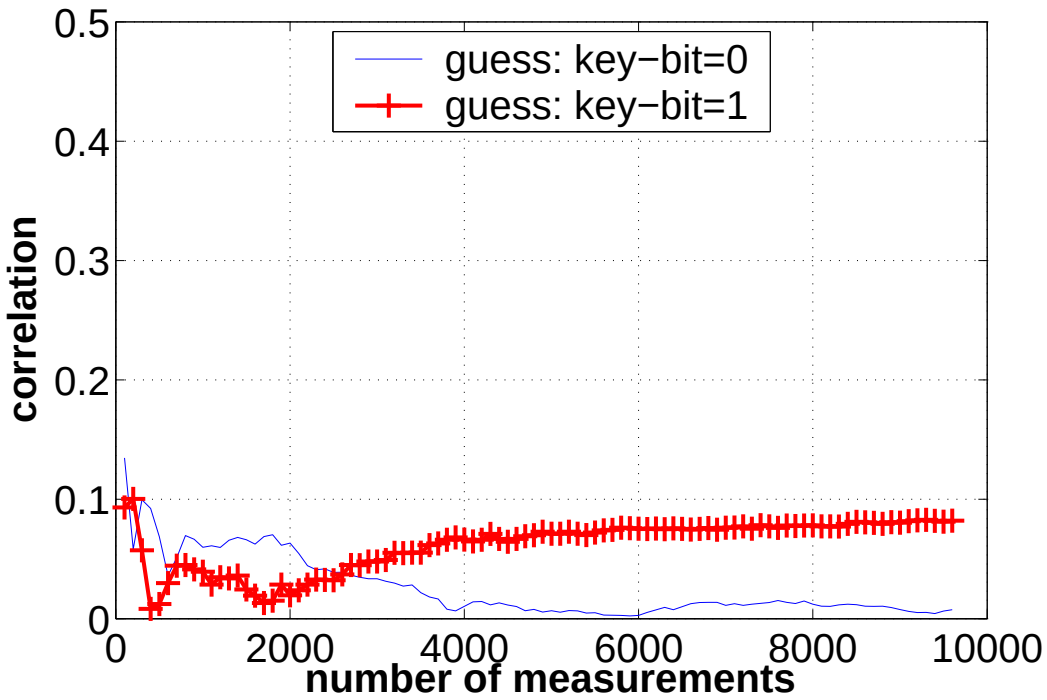
2. Prediction is for:

   - a certain moment of time
   - a certain key guess

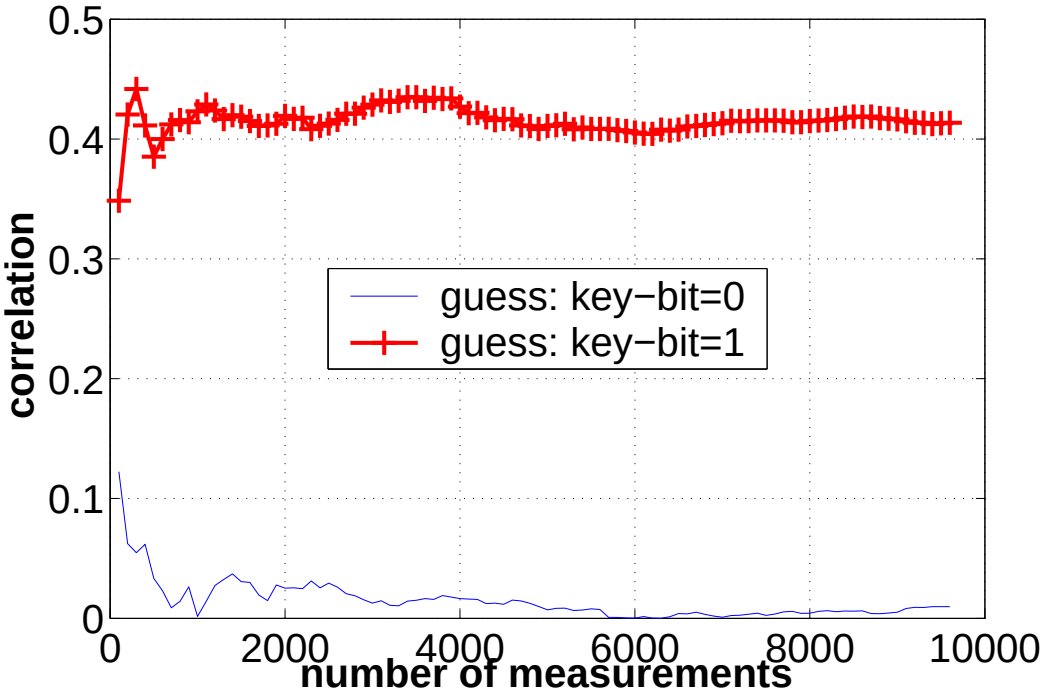
3. Predictions are correlated with the real side-channel output.

   - Correlation is high $\Longrightarrow$ model is correct

Results of the Correlation Analysis

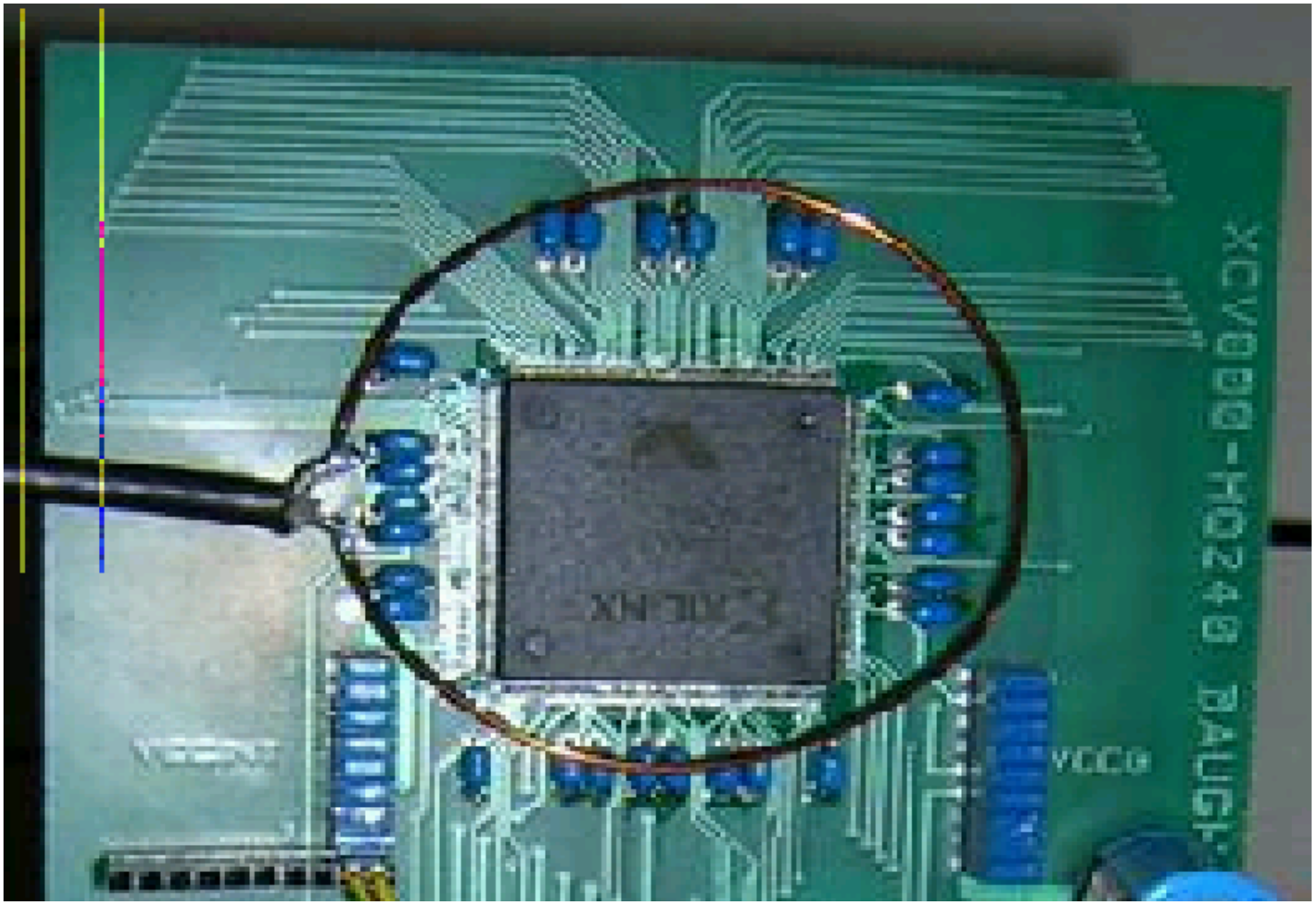


Third spike

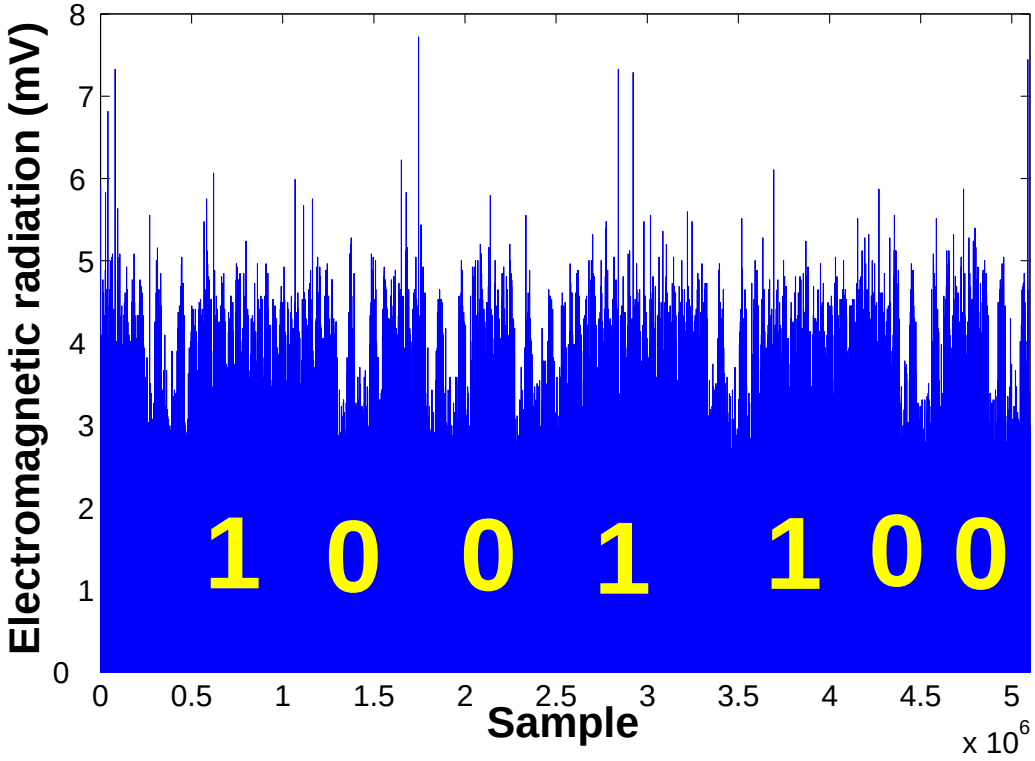


Fifth spike

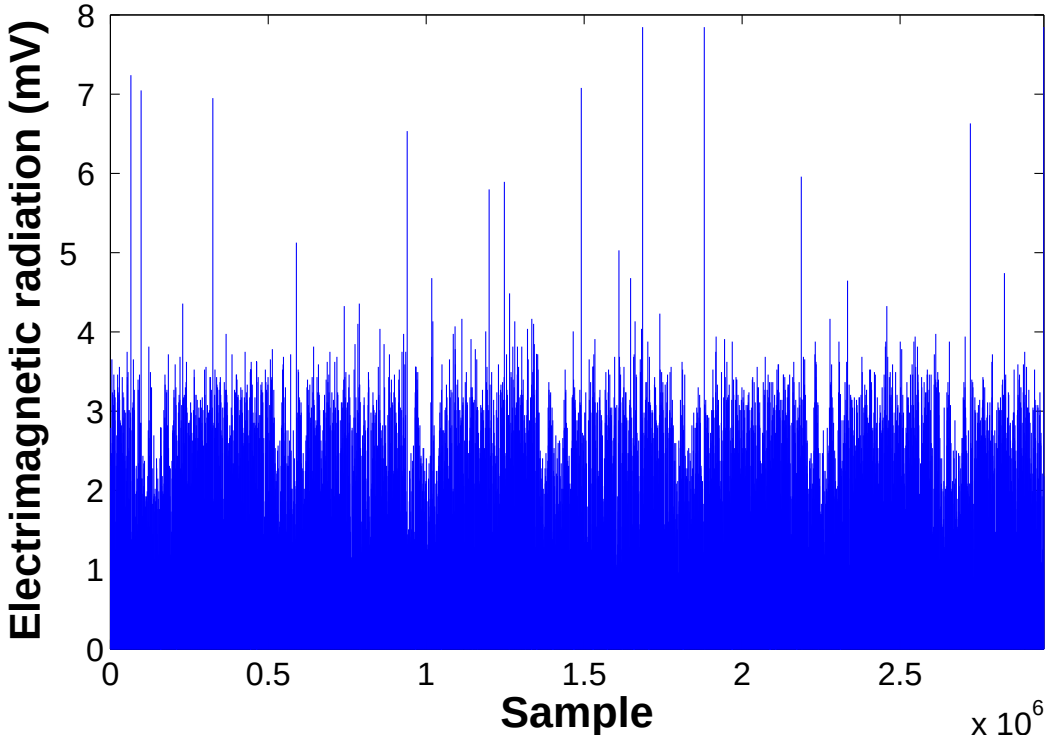
Electromagnetic Analysis of an FPGA Implementation of Elliptic Curve Cryptosystem over $GF(p)$



SEMA Attack

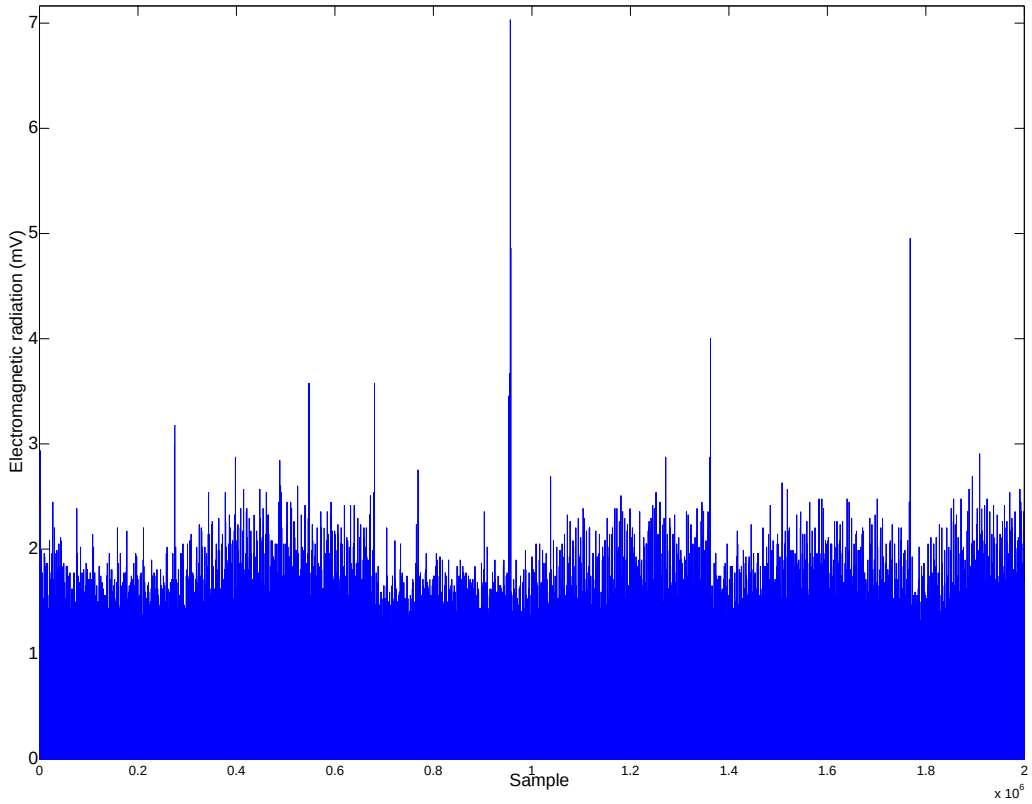


with double and add algorithm



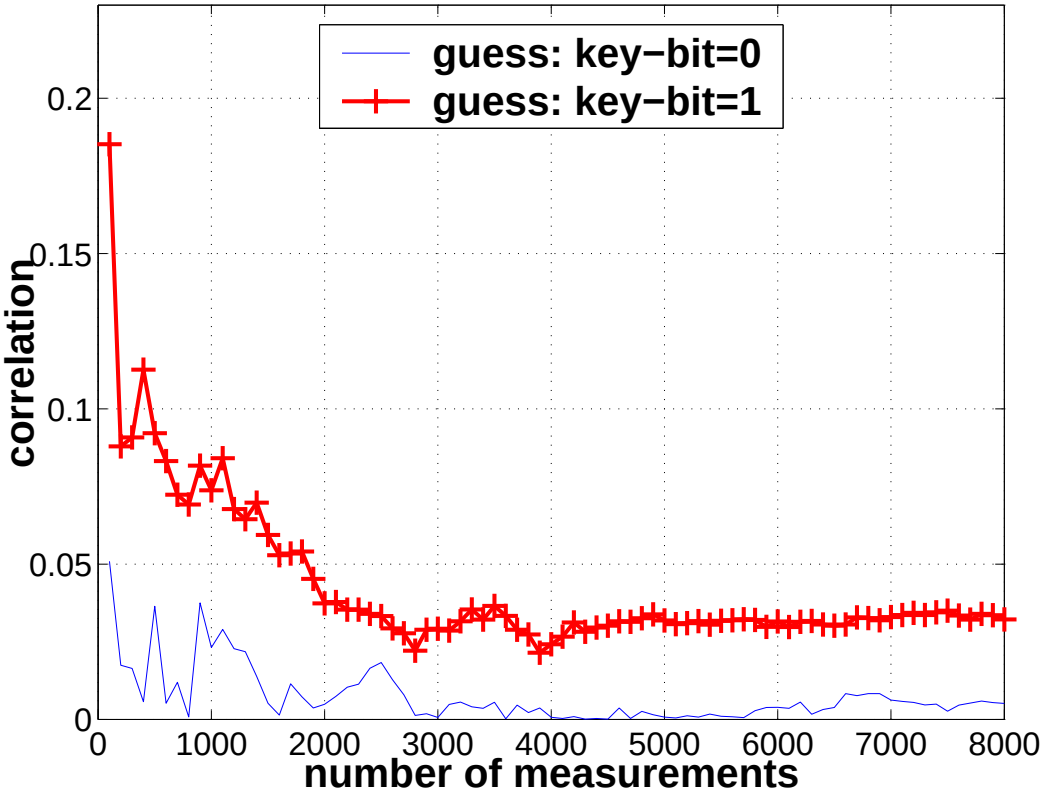
with always double and add algorithm

DEMA Attack

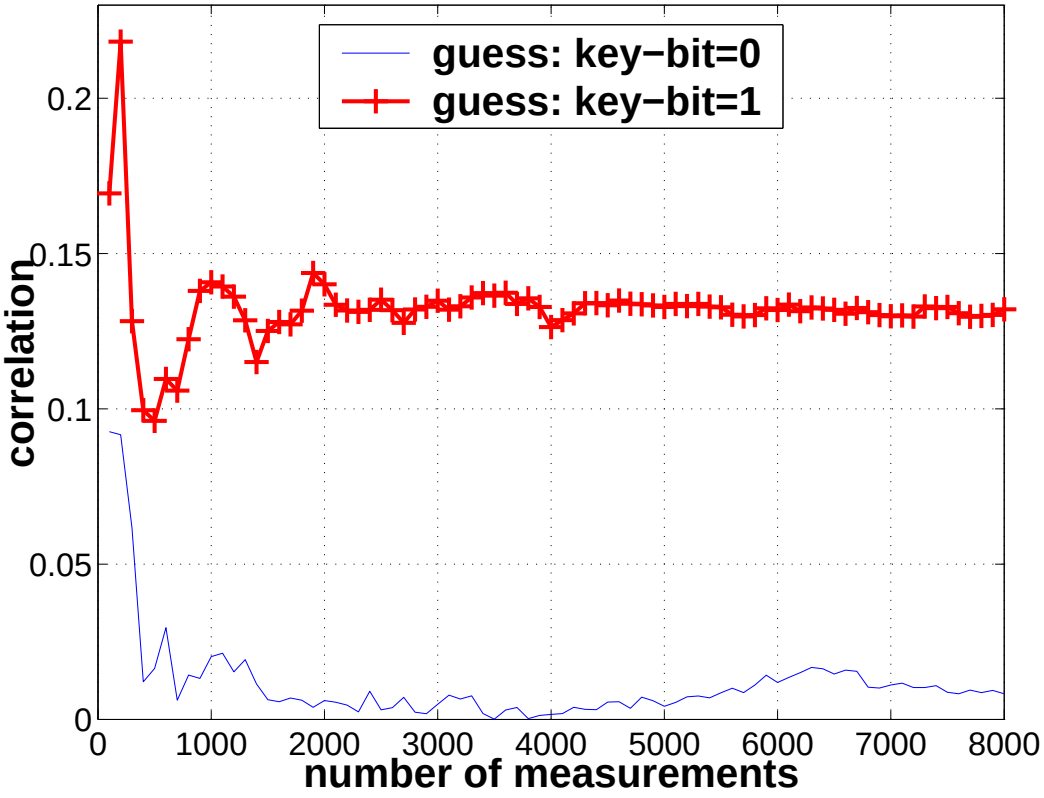


Electromagnetic radiation trace of the FPGA for the attacked point

Correlation Analysis



Third spike



Fifth spike