Lecture 2
Ciphers

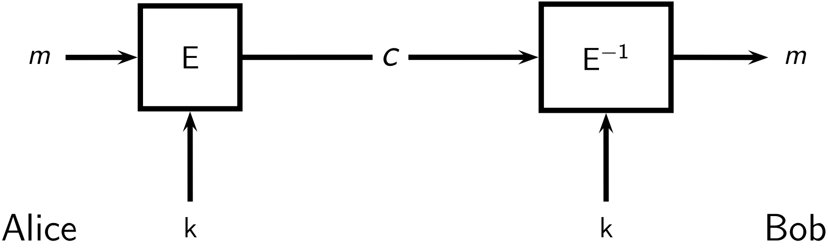
Douglas Wikström
KTH Stockholm
dog@csc.kth.se

January 23, 2015

Introduction to Ciphers

Cipher (Symmetric Cryptosystem)

$c = \mathsf{E}_\mathsf{k}(m)$ $m = \mathsf{E}_\mathsf{k}^{-1}(c)$



Ceasar Cipher (Shift Cipher)

Consider English, with alphabet A-Z_, where _ denotes space, thought of as integers 0-26, i.e., $\mathbb{Z}_{27}$

- **Key.** Random letter $k \in \mathbb{Z}_{27}$.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = m_i + k \bmod 27$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = c_i - k \bmod 27$.

Ceasar Cipher Example

Encoding.

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	_
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Key: $G = 6$

Plaintext.	B	R	I	B	E	_	L	U	L	A	_	T	O	_	B	U	Y	_	J	A	S
Plaintext.	01	17	08	01	04	26	11	20	11	00	26	19	14	26	01	20	24	26	09	00	18
Ciphertext.	07	23	14	07	10	05	17	26	17	06	05	25	20	05	07	26	03	05	15	06	24
Ciphertext.	H	X	O	H	K	F	R	_	R	G	F	Z	U	F	H	_	D	F	P	G	Y

Statistical Attack Against Ceasar (1/3)

Decrypt with all possible keys and see if some English shows up, or more precisely...

Statistical Attack Against Ceasar (2/3)

Written English Letter Frequency Table $F[\cdot]$.

A	0.072	J	0.001	S	0.056
B	0.013	K	0.007	T	0.080
C	0.024	L	0.035	U	0.024
D	0.037	M	0.021	V	0.009
E	**0.112**	N	0.059	W	0.021
F	0.020	O	0.066	X	0.001
G	0.018	P	0.017	Y	0.017
H	0.054	Q	0.001	Z	0.001
I	0.061	R	0.053	**_**	**0.120**

Note that the same frequencies appear in a ciphertext of written English, but in shifted order!

Statistical Attack Against Ceasar (3/3)

- Check that the plaintext of our ciphertext has similar frequencies as written English.
- Find the key k that maximizes the inner product $T(\mathrm{E}_{\mathrm{k}}^{-1}(C)) \cdot F$, where $T(s)$ and F denotes the frequency tables of the string s and English.

This usually gives the correct key k.

Affine Cipher

Affine Cipher.

- **Key.** Random pair k $= (a, b)$, where $a \in \mathbb{Z}_{27}$ is relatively prime to 27, and $b \in \mathbb{Z}_{27}$.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = am_i + b \bmod 27$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = (c_i - b)a^{-1} \bmod 27$.

Substitution Cipher

Ceasar cipher and affine cipher are examples of substitution ciphers.

Substitution Cipher.

- **Key.** Random permutation $\sigma \in S$ of the symbols in the alphabet, for some subset S of all permutations.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = \sigma(m_i)$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = \sigma^{-1}(c_i)$.

Digrams and Trigrams

- A digram is an ordered pair of symbols.
- A trigram is an ordered triple of symbols.
- It is useful to compute frequency tables for the most frequent digrams and trigrams, and not only the frequencies for individual symbols.

Generic Attack Against Substitution Cipher

1. Compute symbol/digram/trigram frequency tables for the candidate language and the ciphertext.
2. Try to match symbols/digrams/trigrams with similar frequencies.
3. Try to recognize words to confirm your guesses (we would use a dictionary (or Google!) here).
4. Backtrack/repeat until the plaintext can be guessed.

This is hard when several symbols have similar frequencies. A large amount of ciphertext is needed. How can we ensure this?

Vigénère

Vigénère Cipher.

- **Key.** $\mathsf{k} = (\mathsf{k}_1, \ldots, \mathsf{k}_l)$, where $\mathsf{k}_i \in \mathbb{Z}_{27}$ is random.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = m_i + \mathsf{k}_{i \bmod l} \bmod 27$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = c_i - \mathsf{k}_{i \bmod l} \bmod 27$.

Vigénère

Vigénère Cipher.

- **Key.** $\mathsf{k} = (\mathsf{k}_1, \ldots, \mathsf{k}_l)$, where $\mathsf{k}_i \in \mathbb{Z}_{27}$ is random.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = m_i + \mathsf{k}_{i \bmod l} \bmod 27$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = c_i - \mathsf{k}_{i \bmod l} \bmod 27$.

We could even make a variant of Vigénère based on the affine cipher, **but is Vigénère really any better than Ceasar?**

Attack Vigénère (1/2)

Index of Coincidence.

- Each probability distribution $p_1, \ldots, p_n$ on n symbols may be viewed as a point $p = (p_1, \ldots, p_n)$ on a $n-1$ dimensional hyperplane in $\mathbb{R}^n$ orthogonal to the vector $\overline{1}$
- Such a point $p = (p_1, \ldots, p_n)$ is at distance $\sqrt{F(p)}$ from the origin, where $F(p) = \sum_{i=1}^{n} p_i^2$.
- It is clear that p is closest to the origin, when p is the uniform distribution, i.e., when $F(p)$ is minimized.
- $F(p)$ is invariant under permutation of the underlying symbols $\longrightarrow$ tool to check if a set of symbols is the result of *some* substitution cipher.

Attack Vigénère (2/2)

1. For $l = 1, 2, 3, \ldots$, we form

$$\begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_l \end{pmatrix} = \begin{pmatrix} c_1 & c_{l+1} & c_{2l+1} & \cdots \\ c_2 & c_{l+2} & c_{2l+2} & \cdots \\ \vdots & \vdots & \vdots & \ddots \\ c_l & c_{2l} & c_{3l} & \cdots \end{pmatrix}$$

   and compute $f_l = \frac{1}{l}\sum_{i=1}^{l} F(F_i)$, where F_i is the frequency table of C_i.

2. A local maximum with smallest l is probably the right length.
3. Then attack each C_i separately to recover k_i, using the attack against the Ceasar cipher.

Hill Cipher

Hill Cipher.

- **Key.** $\mathrm{k} = A$, where A is an invertible $l \times l$-matrix over $\mathbb{Z}_{27}$.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^n$ gives ciphertext $c = (c_1, \ldots, c_n)$, where (computed modulo 27):

$$(c_{i+0}, \ldots, c_{i+l-1}) = (m_{i+0}, \ldots, m_{i+l-1})A \ .$$

- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^n$ gives plaintext $m = (m_1, \ldots, m_n)$, where (computed modulo 27):

$$(m_{i+0}, \ldots, m_{i+l-1}) = (c_{i+0}, \ldots, c_{i+l-1})A^{-1} \ .$$

  for $i = 1, l+1, 2l+1, \ldots$

Permutation Cipher (Transposition Cipher)

The permutation cipher is a special case of the Hill cipher.

Permutation Cipher.

- **Key.** Random permutation $\pi \in S$ for some subset S of the set of permutations of $\{1, 2, \ldots, l\}$.
- **Encrypt.** Plaintext $m = (m_1, \ldots, m_n) \in \mathbb{Z}_{27}^l$ gives ciphertext $c = (c_1, \ldots, c_n)$, where $c_i = m_{\pi(i \bmod l)}$.
- **Decrypt.** Ciphertext $c = (c_1, \ldots, c_n) \in \mathbb{Z}_{27}^l$ gives plaintext $m = (m_1, \ldots, m_n)$, where $m_i = c_{\pi^{-1}(i \bmod l)}$.

Substitution-Permutation Networks

Ideal Block Cipher

- For every key a block-cipher with plaintext/ciphertext space $\{0,1\}^n$ gives a permutation of $\{0,1\}^n$.

  What would be an ideal cipher?

Ideal Block Cipher

- For every key a block-cipher with plaintext/ciphertext space $\{0,1\}^n$ gives a permutation of $\{0,1\}^n$.

  What would be an ideal cipher?

- The ideal cipher is one where each key gives a **randomly chosen permutation** of $\{0,1\}^n$.

  Why is this not possible?

Ideal Block Cipher

- For every key a block-cipher with plaintext/ciphertext space $\{0,1\}^n$ gives a permutation of $\{0,1\}^n$.

  What would be an ideal cipher?

- The ideal cipher is one where each key gives a **randomly chosen permutation** of $\{0,1\}^n$.

  Why is this not possible?

- The representation of a single typical function $\{0,1\}^n \to \{0,1\}^n$ requires roughly $n2^n$ bits (130 million TB for $n = 64$)

Ideal Block Cipher

- For every key a block-cipher with plaintext/ciphertext space $\{0,1\}^n$ gives a permutation of $\{0,1\}^n$.

  What would be an ideal cipher?

- The ideal cipher is one where each key gives a **randomly chosen permutation** of $\{0,1\}^n$.

  Why is this not possible?

- The representation of a single typical function $\{0,1\}^n \to \{0,1\}^n$ requires roughly $n2^n$ bits (130 million TB for $n = 64$)

- What should we look for instead?

Something Smaller

Idea. Compose smaller permutations into a large one. Mix the components "thoroughly".

Something Smaller

Idea. Compose smaller permutations into a large one. Mix the components "thoroughly".

Shannon (1948) calls this:

- **Diffusion.** "In the method of diffusion the statistical structure of M which leads to its redundancy is dissipated into long range statistics..."
- **Confusion.** "The method of confusion is to make the relation between the simple statistics of E and the simple description of K a very complex and involved one."

Substitution-Permutation Networks (1/2)

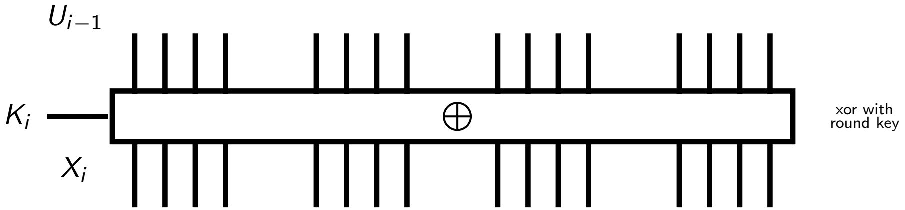
- **Block-size.** We use a block-size of $n = \ell \times m$ bits.
- **Key Schedule.** Each round r uses its own round key K_r derived from the key K using a key schedule.
- **Each Round.** In each round we invoke:
  1. **Round Key.** xor with the current round key.
  2. **Substitution.** ℓ substitution boxes each acting on one m-bit block (m-bit S-Boxes).
  3. **Permutation.** A permutation π_i acting on $\{1, \ldots, n\}$ to reorder the n bits.

Substitution-Permutation Networks (2/2)

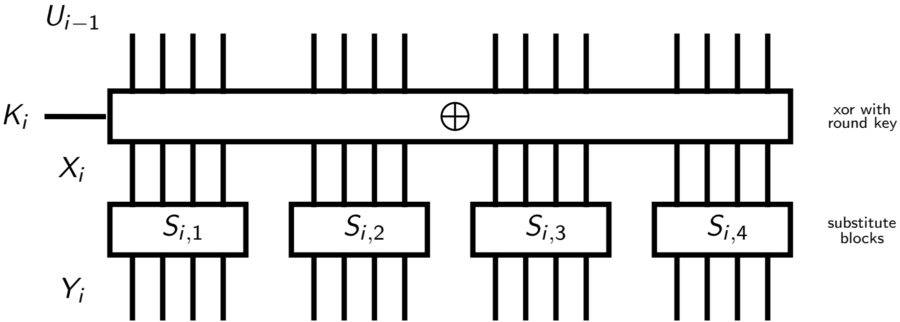
U_{i-1}

K_i

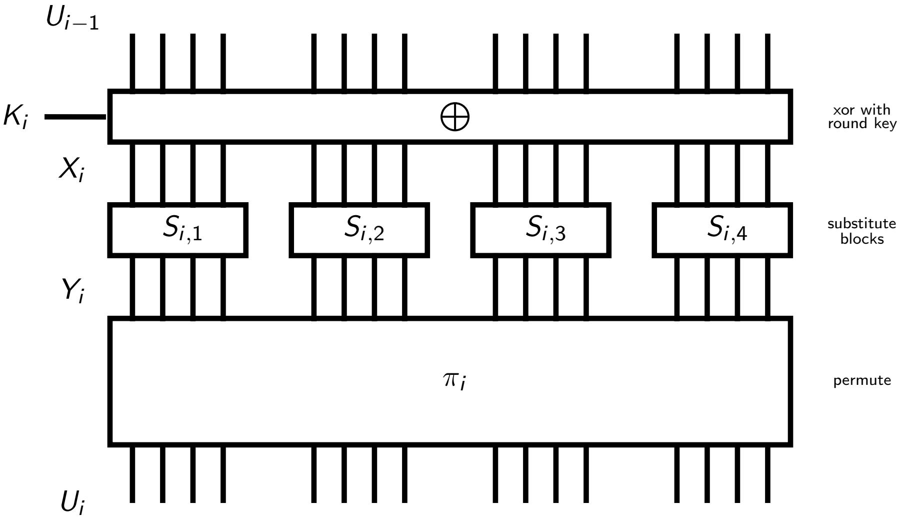
Substitution-Permutation Networks (2/2)



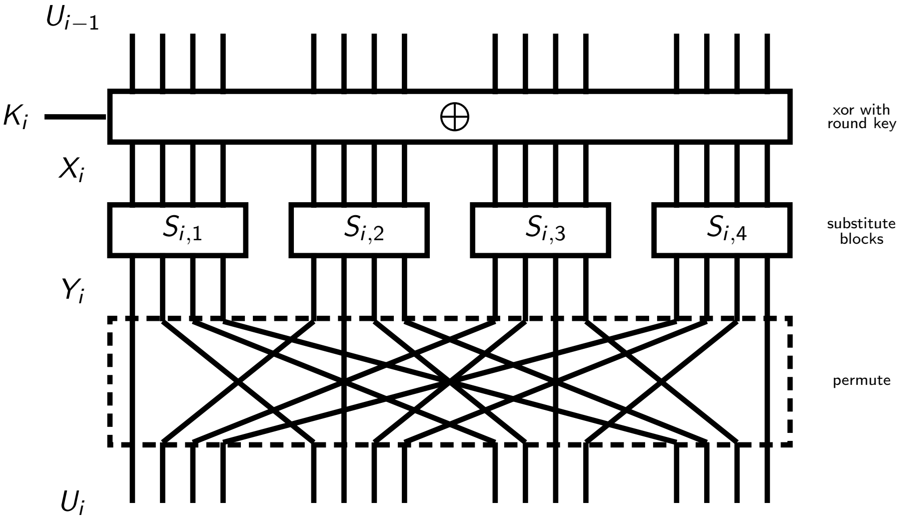
Substitution-Permutation Networks (2/2)



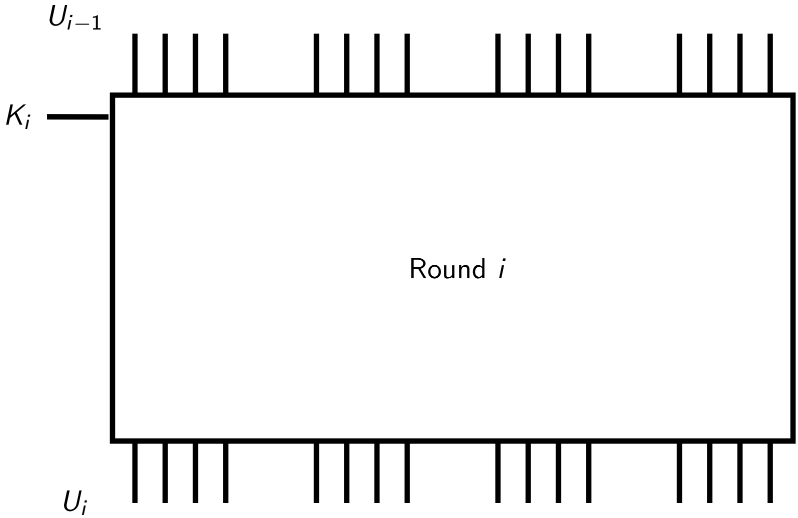
Substitution-Permutation Networks (2/2)



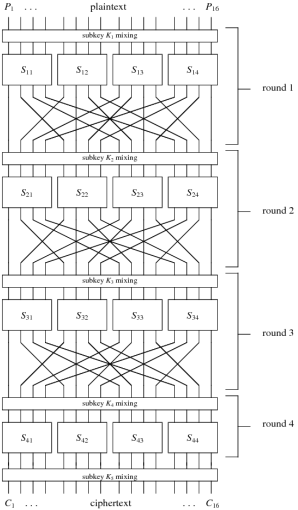
Substitution-Permutation Networks (2/2)



Substitution-Permutation Networks (2/2)



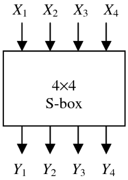
A Simple Block Cipher (1/2)



- $|P| = |C| = 16$
- 4 rounds
- $|K| = 32$
- rth round key K_r consists of the $4r$th to the $(4r + 16)$th bits of key K.
- 4-bit S-Boxes

A Simple Block Cipher (2/2)

S-Boxes the same ($S \neq S^{-1}$)

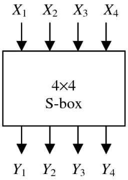


- $Y = S(X)$
- Can be described using 4 boolean functions

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

A Simple Block Cipher (2/2)

S-Boxes the same ($S \neq S^{-1}$)



- $Y = S(X)$
- Can be described using 4 boolean functions

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

16-bit permutation ($\pi = \pi^{-1}$)

Input	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Output	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

Linear Cryptanalysis

Basic Idea – Linearize

Find an expression of the following form with a high probability of occurrence.

$$P_{i_1} \oplus \cdots \oplus P_{i_p} \oplus C_{j_1} \oplus \cdots \oplus C_{j_c} = K_{\ell_1, s_1} \oplus \cdots \oplus K_{\ell_k, s_k}$$

Each random plaintext/ciphertext pair gives an estimate of

$$K_{\ell_1, s_1} \oplus \cdots \oplus K_{\ell_k, s_k}$$

Collect many pairs and make a better estimate based on the majority vote.

How do we come up with the desired expression?

How do we compute the required number of samples?

Bias

Definition. The bias $\epsilon(X)$ of a binary random variable X is defined by

$$\epsilon(X) = \Pr[X = 0] - \frac{1}{2} \ .$$

Bias

Definition. The bias $\epsilon(X)$ of a binary random variable X is defined by

$$\epsilon(X) = \Pr\left[X = 0\right] - \frac{1}{2} \ .$$

$\approx 1/\epsilon^2(X)$ samples are required to estimate X (Matsui)

Linear Approximation of S-Box (1/3)

Let X and Y be the input and output of an S-box, i.e.

$$Y = S(X) \ .$$

We consider the bias of linear combinations of the form

$$a \cdot X \oplus b \cdot Y = \left(\bigoplus_i a_i X \right) \oplus \left(\bigoplus_i b_i Y \right) \ .$$

Linear Approximation of S-Box (1/3)

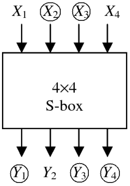
Let X and Y be the input and output of an S-box, i.e.

$$Y = S(X) \ .$$

We consider the bias of linear combinations of the form

$$a \cdot X \oplus b \cdot Y = \left(\bigoplus_i a_i X \right) \oplus \left(\bigoplus_i b_i Y \right) \ .$$

Example: $X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4$
The expression holds in 12 out of the 16 cases. Hence, it has a bias of $(12 - 8)/16 = 4/16 = 1/4$.



Linear Approximation of S-Box (2/3)

- Let $N_L(a, b)$ be the number of zero-outcomes of $a \cdot X \oplus b \cdot Y$.
- The bias is then

$$\epsilon(a \cdot X \oplus b \cdot Y) = \frac{N_L(a, b) - 8}{16} \ ,$$

since there are four bits in X, and Y is determined by X.

Linear Approximation Table (3/3)

$$N_L(a, b) - 8$$

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

This gives linear approximation for one round.

How do we come up with linear approximation for more rounds?

Piling-Up Lemma

Lemma. Let $X_1, \ldots, X_t$ be independent binary random variables and let $\epsilon_i = \epsilon(X_i)$. Then

$$\epsilon\left(\bigoplus_i X_i\right) = 2^{t-1} \prod_i \epsilon_i \ .$$

Proof. Case $t = 2$:

$$\begin{aligned}
\Pr[X_1 \oplus X_2 = 0] &= \Pr[(X_1 = 0 \wedge X_1 = 0) \vee (X_1 = 1 \wedge X_1 = 1)] \\
&= (\frac{1}{2} + \epsilon_1)(\frac{1}{2} + \epsilon_2) + (\frac{1}{2} - \epsilon_1)(\frac{1}{2} - \epsilon_2) \\
&= \frac{1}{2} + 2\epsilon_1\epsilon_2 \ .
\end{aligned}$$

By induction $\Pr[X_1 \oplus \cdots \oplus X_t = 0] = \frac{1}{2} + 2^{t-1} \prod_i \epsilon_i$

Linear Trail

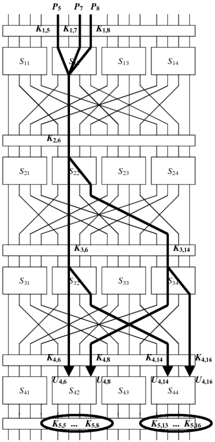
Four linear approximations with $|\epsilon_i| = 1/4$

$$
\begin{aligned}
S_{12}: &\quad X_1 \oplus X_3 \oplus X_4 = Y_2 \\
S_{22}: &\quad X_2 = Y_2 \oplus Y_4 \\
S_{32}: &\quad X_2 = Y_2 \oplus Y_4 \\
S_{34}: &\quad X_2 = Y_2 \oplus Y_4
\end{aligned}
$$

Combine them to get:

$$U_{4,6} \oplus U_{4,8} \oplus U_{4,14} \oplus U_{4,16} \oplus P_5 \oplus P_7 \oplus P_8 = \bigoplus K_{i,j}$$

with bias $|\epsilon| = 2^{4-1}(\frac{1}{4})^4 = 2^{-5}$



Attack Idea

- Our expression (with bias 2^{-5}) links plaintext bits to input bits to the 4th round
- Partially undo the last round by guessing the last key. Only 2 S-Boxes are involved, i.e., $2^8 = 256$ guesses
- For a correct guess, the equation holds with bias 2^{-5}. For a wrong guess, it holds with bias zero (i.e., probability close to $1/2$).

Attack Idea

- Our expression (with bias 2^{-5}) links plaintext bits to input bits to the 4th round
- Partially undo the last round by guessing the last key. Only 2 S-Boxes are involved, i.e., $2^8 = 256$ guesses
- For a correct guess, the equation holds with bias 2^{-5}. For a wrong guess, it holds with bias zero (i.e., probability close to $1/2$).

Required pairs $2^{10} \approx 1000$
Attack complexity $2^{18} \ll 2^{32}$ operations

Linear Cryptanalysis Summary

1. Find linear approximation of S-Boxes.
2. Compute bias of each approximation.
3. Find linear trails.
4. Compute bias of linear trails.
5. Compute data and time complexity.
6. Estimate key bits from many plaintext-ciphertexts pairs.

Linear cryptanalysis is a **known plaintext attack**.