Lecture 3
Substitution-Permutation Networks, Linear Cryptanalysis, AES

Douglas Wikström
KTH Stockholm
dog@csc.kth.se

February 7, 2014

Last Time

- Ceasar and Affine ciphers.
- General (monoalphabetical) substitution.
- Vinénère cipher.
- Permutation map.
- General invertible linear map.

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 $n2^n$ 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 $n2^n$ 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)

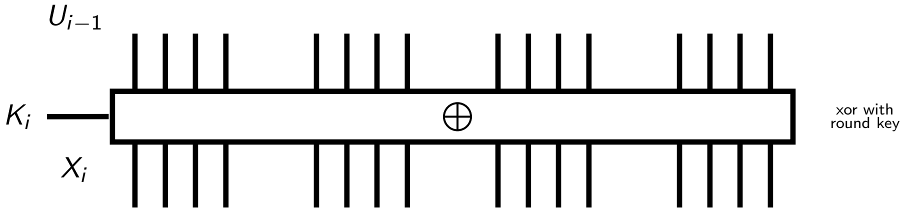
- **Block-size.** We use a block-size of $n = \ell \times m$ bits.
- **Key Schedule.** Each round r uses its own round key K_r derived from the key K using a key schedule.
- **Each Round.** In each round we invoke:
  1. **Round Key.** xor with the current round key.
  2. **Substitution.** ℓ substitution 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)

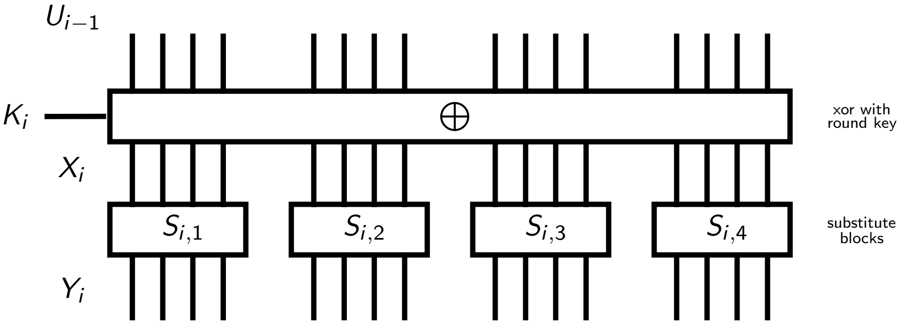
U_{i-1}

K_i

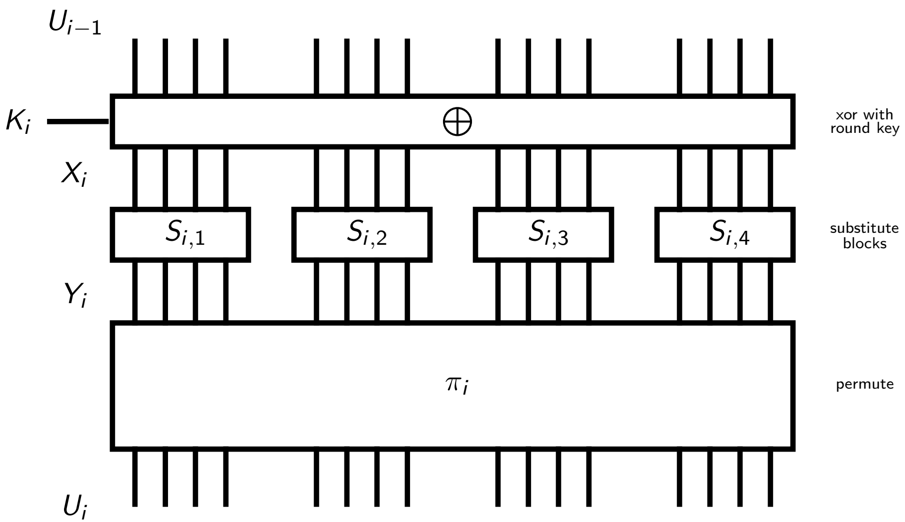
Substitution-Permutation Networks (2/2)



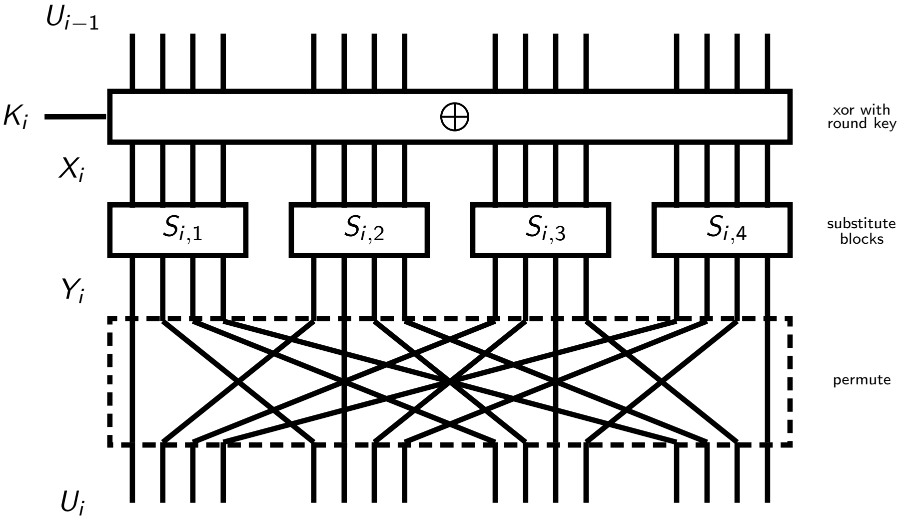
Substitution-Permutation Networks (2/2)



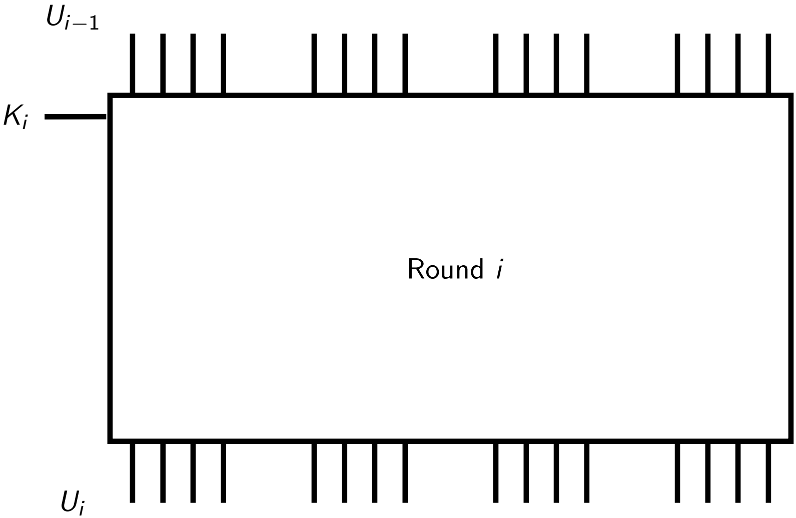
Substitution-Permutation Networks (2/2)



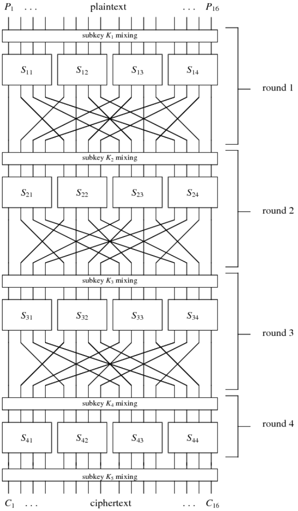
Substitution-Permutation Networks (2/2)



Substitution-Permutation Networks (2/2)



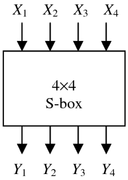
A Simple Block Cipher (1/2)



- $|P| = |C| = 16$
- 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 ($S \neq S^{-1}$)

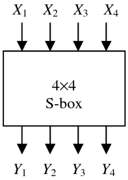


- $Y = S(X)$
- Can be described using 4 boolean functions

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

Basic Idea

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

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

Bias

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

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

$\approx 1/\epsilon^2(X)$ 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)

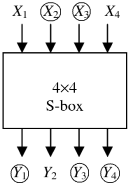
Let X and Y be the input and output of an S-box, i.e.

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

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

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

Proof. Case $t = 2$:

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

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

Linear Trail

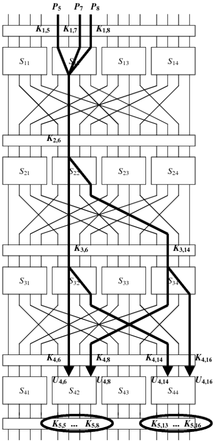
Four linear approximations with $|\epsilon_i| = 1/4$

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

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} operations

Linear Cryptanalysis Summary

1. Approximation of S-Boxes
2. Finding linear trails
3. Computing the bias
4. Identifying relevant key bits
5. Computing data and time complexity
6. Mounting the attack

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

Quote

The news here is not that DES is insecure, that hardware algorithm-crackers can be built, or that a 56-bit key length is too short. ... The news is how long the government has been denying that these machines were possible. As recently as 8 June 98, Robert Litt, principal associate deputy attorney general at the Department of Justice, denied that it was possible for the FBI to crack DES. ... My comment was that the FBI is either incompetent or lying, or both.

– Bruce Schneier, 1998

Advanced Encryption Standard (AES)

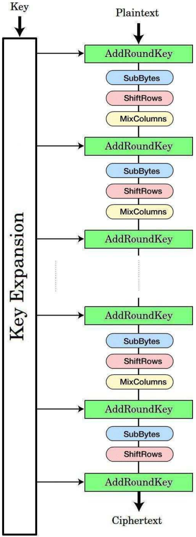
- Chosen in worldwide **public competition** 1998-2000. Probably no back-doors. Increased confidence!
- Winning proposal named "Rijndael", by Rijmen and Daemen
- Family of 128-bit block ciphers:

Key bits	128	192	256
Rounds	10	12	14

- The first key-recovery attacks on full AES due to Bogdanov, Khovratovich, and Rechberger, published **2011**, is faster than brute force by a factor of about **4**.
- ... algebraics of AES make some people uneasy.

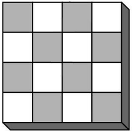
AES

- **AddRoundKey**: XOR With Round Key
- **SubBytes**: Substitution
- **ShiftRows**: Permutation
- **MixColumns**: Linear Map



Similar to SPN

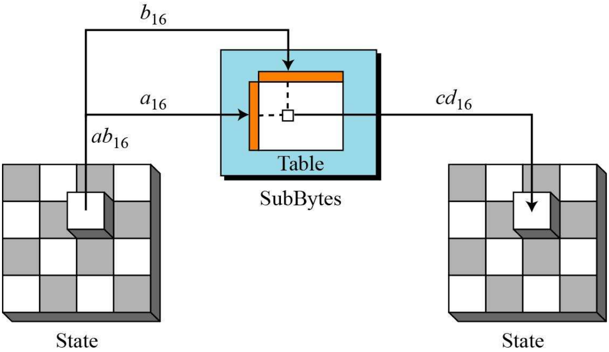
The 128 bit state is interpreted as a 4×4 matrix of bytes.



Something like a mix between substitution, permutation, affine version of Hill cipher. In each round!

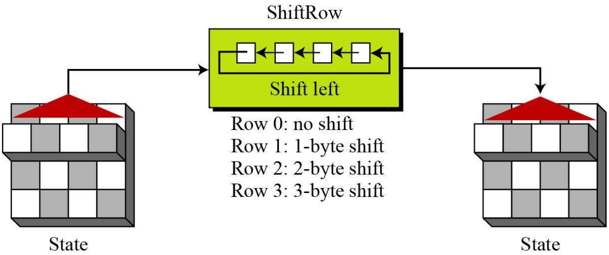
SubBytes

SubBytes is field inversion in $\mathbb{F}_{2^8}$ plus affine map in $\mathbb{F}_2^8$.



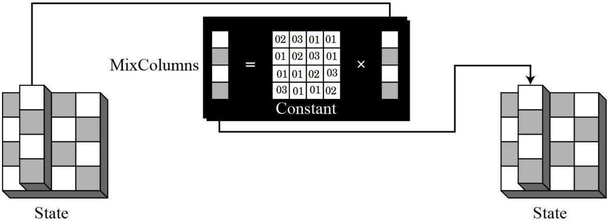
ShiftRows

`ShiftRows` is a cyclic shift of bytes with offsets: 0, 1, 2, and 3.



MixColumns

`MixColumns` is an invertible linear map over $\mathbb{F}_{2^8}$ (with irreducibile polynomial $x^8 + x^4 + x^3 + x + 1$) with good diffusion.



Decryption

Uses the following transformations:

- **AddRoundKey**
- **InvSubBytes**
- **InvShiftRows**
- **InvMixColumns**