Linear and Differential Cryptanalysis of DES

Slava Chernyak and Sourav Sen Gupta

University of Washington

February 2, 2009

$\mathcal{O}$nce upon a time ...

$\mathcal{O}$nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

$\mathcal{O}$nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm

$\mathcal{O}$nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment

$\mathcal{O}$nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment
- 1976: DES is approved as a standard

$\mathcal{O}$nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment
- 1976: DES is approved as a standard
- 1977: DES is published as a FIPS standard FIPS PUB 46

Once upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment
- 1976: DES is approved as a standard
- 1977: DES is published as a FIPS standard FIPS PUB 46
- 1992: Biham and Shamir report the *differential cryptanalysis*

*O*nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment
- 1976: DES is approved as a standard
- 1977: DES is published as a FIPS standard FIPS PUB 46
- 1992: Biham and Shamir report the *differential cryptanalysis*
- 1993: DES is reaffirmed for the third time as FIPS 46-2

$\mathcal{O}$nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment
- 1976: DES is approved as a standard
- 1977: DES is published as a FIPS standard FIPS PUB 46
- 1992: Biham and Shamir report the *differential cryptanalysis*
- 1993: DES is reaffirmed for the third time as FIPS 46-2
- 1994: The first *linear cryptanalysis* of DES is performed by Matsui

$\mathcal{O}$nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment
- 1976: DES is approved as a standard
- 1977: DES is published as a FIPS standard FIPS PUB 46
- 1992: Biham and Shamir report the *differential cryptanalysis*
- 1993: DES is reaffirmed for the third time as FIPS 46-2
- 1994: The first *linear cryptanalysis* of DES is performed by Matsui
- 1997: The DESCHALL Project breaks a DES message in public

Once upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment
- 1976: DES is approved as a standard
- 1977: DES is published as a FIPS standard FIPS PUB 46
- 1992: Biham and Shamir report the *differential cryptanalysis*
- 1993: DES is reaffirmed for the third time as FIPS 46-2
- 1994: The first *linear cryptanalysis* of DES is performed by Matsui
- 1997: The DESCHALL Project breaks a DES message in public
- 1998: The EFF's Deep Crack *brute forces* a DES key in 56 hours

$\mathcal{O}$nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment
- 1976: DES is approved as a standard
- 1977: DES is published as a FIPS standard FIPS PUB 46
- 1992: Biham and Shamir report the *differential cryptanalysis*
- 1993: DES is reaffirmed for the third time as FIPS 46-2
- 1994: The first *linear cryptanalysis* of DES is performed by Matsui
- 1997: The DESCHALL Project breaks a DES message in public
- 1998: The EFF's Deep Crack *brute forces* a DES key in 56 hours
- 1999: Deep Crack and distributed.net break DES in 22 hours

$\mathcal{O}$nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment
- 1976: DES is approved as a standard
- 1977: DES is published as a FIPS standard FIPS PUB 46
- 1992: Biham and Shamir report the *differential cryptanalysis*
- 1993: DES is reaffirmed for the third time as FIPS 46-2
- 1994: The first *linear cryptanalysis* of DES is performed by Matsui
- 1997: The DESCHALL Project breaks a DES message in public
- 1998: The EFF's Deep Crack *brute forces* a DES key in 56 hours
- 1999: Deep Crack and distributed.net break DES in 22 hours
- 1999: FIPS 46-3 reaffirms DES, preferring the use of Triple DES

$\mathcal{O}$nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment
- 1976: DES is approved as a standard
- 1977: DES is published as a FIPS standard FIPS PUB 46
- 1992: Biham and Shamir report the *differential cryptanalysis*
- 1993: DES is reaffirmed for the third time as FIPS 46-2
- 1994: The first *linear cryptanalysis* of DES is performed by Matsui
- 1997: The DESCHALL Project breaks a DES message in public
- 1998: The EFF's Deep Crack *brute forces* a DES key in 56 hours
- 1999: Deep Crack and distributed.net break DES in 22 hours
- 1999: FIPS 46-3 reaffirms DES, preferring the use of Triple DES
- 2001: The Advanced Encryption Standard (AES) is published in FIPS 197

$\mathcal{O}$nce upon a time ... there was a secure standard encryption algorithm called DES (Data Encryption Standard) ...

- 1973: NBS publishes a request for a standard encryption algorithm
- 1975: DES is published in the Federal Register for comment
- 1976: DES is approved as a standard
- 1977: DES is published as a FIPS standard FIPS PUB 46
- 1992: Biham and Shamir report the *differential cryptanalysis*
- 1993: DES is reaffirmed for the third time as FIPS 46-2
- 1994: The first *linear cryptanalysis* of DES is performed by Matsui
- 1997: The DESCHALL Project breaks a DES message in public
- 1998: The EFF's Deep Crack *brute forces* a DES key in 56 hours
- 1999: Deep Crack and distributed.net break DES in 22 hours
- 1999: FIPS 46-3 reaffirms DES, preferring the use of Triple DES
- 2001: The Advanced Encryption Standard (AES) is published in FIPS 197
- 2002: The AES standard becomes effective

1. Introduction
   - Data Encryption Standard
   - Linear Cryptanalysis
   - Differential Cryptanalysis
2. Mathematical Framework
   - Substitution-Permutation Network
   - Linear Attack on SPN
   - Differential Attack on SPN
3. Attacks on DES
   - Linear Attack: 4-round DES
   - Differential Attack: 4-round DES
4. Conclusion
   - A few important points
   - Take Home
   - Fun facts

A quick review of DES

DES Function

DES : (Plaintext (P), Key (K)) $\mapsto$ Cipher (C)

DES Unit Blocks

$$
\begin{aligned}
\sigma_{i+1} &: (0,1)^{64} \to (0,1)^{64} \\
&(L_i, R_i) \mapsto (L_i \oplus f(R_i, K_{i+1}), R_i) \\
\tau &: (0,1)^{64} \to (0,1)^{64} \\
&(L_{i+1}, R_{i+1}) \mapsto (R_{i+1}, L_{i+1})
\end{aligned}
$$

- Input for round $i+1$: (L_i, R_i)
- Subkey for round $i+1$: K_{i+1}
- Feistel function (f)
- 16-rounds for $i = 0, 1, ..., 15$

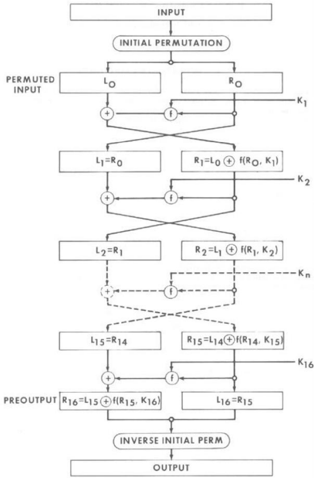


Figure: DES Algorithm

Designers: H Feistel, W Tuchman, D Coppersmith, A Konheim, E Grossman, B Notz, L Smith and B Tuckerman from IBM

Feistel Function

$$f : (0,1)^{32} \times (0,1)^{48} \rightarrow (0,1)^{32}$$
$$(R, K) \mapsto P(S_{Box}(E(R) \oplus K))$$

- Right half of the plaintext (R)
- Expansion function
  $E : (0,1)^{32} \rightarrow (0,1)^{48}$
- Key for the round (K)
- Confusion function (S-Boxes)
  $S_i : (0,1)^{6} \rightarrow (0,1)^{4}$
- Diffusion function
  $P : (0,1)^{32} \rightarrow (0,1)^{32}$

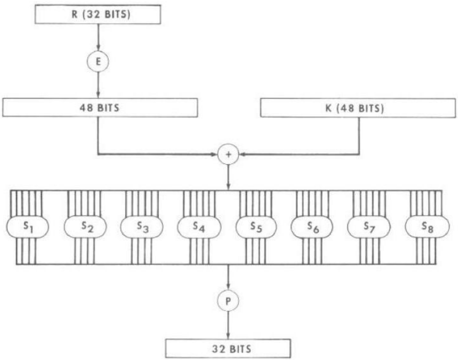


Figure: Feistel Round for DES

Mathematically, the different pieces of DES are:

- Initial Permutation
  $IP : (0,1)^{64} \longrightarrow (0,1)^{64}$
- DES Unit Block functions
  $\sigma_{i+1}(L_i, R_i)$ for $i = 0, 1, ..., 15$
- Transposition
  $\tau : (L_{i+1}, R_{i+1}) \longmapsto (R_{i+1}, L_{i+1})$ for $i = 0, 1, ..., 14$
- Inverse Initial Permutation
  $IP^{-1} : (0,1)^{64} \longrightarrow (0,1)^{64}$

Mathematically, the different pieces of DES are:

- Initial Permutation
  $IP : (0,1)^{64} \longrightarrow (0,1)^{64}$
- DES Unit Block functions
  $\sigma_{i+1}(L_i, R_i)$ for $i = 0, 1, ..., 15$
- Transposition
  $\tau : (L_{i+1}, R_{i+1}) \mapsto (R_{i+1}, L_{i+1})$ for $i = 0, 1, ..., 14$
- Inverse Initial Permutation
  $IP^{-1} : (0,1)^{64} \longrightarrow (0,1)^{64}$

Algebraic Representation of DES

$$C = DES_K(P) = IP^{-1}\,\sigma_{16}\,\tau\,\cdots\,\tau\,\sigma_1\,IP\,(\,P\,)$$
$$P = DES_K^{-1}(C) = IP^{-1}\,\sigma_1\,\tau\,\cdots\,\tau\,\sigma_{16}\,IP\,(\,C\,)$$

Mathematically, the different pieces of DES are:

- Initial Permutation
  $IP : (0,1)^{64} \longrightarrow (0,1)^{64}$
- DES Unit Block functions
  $\sigma_{i+1}(L_i, R_i)$ for $i = 0, 1, ..., 15$
- Transposition
  $\tau : (L_{i+1}, R_{i+1}) \mapsto (R_{i+1}, L_{i+1})$ for $i = 0, 1, ..., 14$
- Inverse Initial Permutation
  $IP^{-1} : (0,1)^{64} \longrightarrow (0,1)^{64}$

Algebraic Representation of DES

$$
\begin{aligned}
C &= DES_K(P) = IP^{-1}\,\sigma_{16}\,\tau \cdots \tau\,\sigma_1\,IP\,(\,P\,) \\
P &= DES_K^{-1}(C) = IP^{-1}\,\sigma_1\,\tau \cdots \tau\,\sigma_{16}\,IP\,(\,C\,)
\end{aligned}
$$

Note: The inverse DES comes from the fact that $\tau^2 = \sigma_i^2 = 1$

Basic Idea of Linear Cryptanalysis

Linear Attack on DES idea:

- S-Boxes depend on a relatively small number of bits (6) which allows us to write down linear (or affine) expressions that approximate S-Boxes

Linear Attack on DES idea:

- S-Boxes depend on a relatively small number of bits (6) which allows us to write down linear (or affine) expressions that approximate S-Boxes
- The effects of one round do not diffuse quickly over following rounds. Thus linear or affine expressions (as above) that hold per-round can be combined across rounds

Specifically, if P_i are plaintext bits, C_i are ciphertext bits, and K_i are subkey bits, then we wish to find an expression of the form

$$P_{i_1} \oplus P_{i_2} \cdots P_{i_j} \oplus C_{i_1} \oplus C_{i_2} \cdots C_{i_k} = K_{i_1} \oplus K_{i_2} \oplus K_{i_m}$$

Specifically, if P_i are plaintext bits, C_i are ciphertext bits, and K_i are subkey bits, then we wish to find an expression of the form

$$P_{i_1} \oplus P_{i_2} \cdots P_{i_j} \oplus C_{i_1} \oplus C_{i_2} \cdots C_{i_k} = K_{i_1} \oplus K_{i_2} \oplus K_{i_m}$$

such that this expression has a high *or* low probability of occurrence.

Specifically, if P_i are plaintext bits, C_i are ciphertext bits, and K_i are subkey bits, then we wish to find an expression of the form

$$P_{i_1} \oplus P_{i_2} \cdots P_{i_j} \oplus C_{i_1} \oplus C_{i_2} \cdots C_{i_k} = K_{i_1} \oplus K_{i_2} \oplus K_{i_m}$$

such that this expression has a high *or* low probability of occurrence.

Consider: No such obvious expression should exist, otherwise the cipher is trivially weak. If we were to randomly select bits for the above expression, it would hold exactly 1/2 the time.

Specifically, if P_i are plaintext bits, C_i are ciphertext bits, and K_i are subkey bits, then we wish to find an expression of the form

$$P_{i_1} \oplus P_{i_2} \cdots P_{i_j} \oplus C_{i_1} \oplus C_{i_2} \cdots C_{i_k} = K_{i_1} \oplus K_{i_2} \oplus K_{i_m}$$

such that this expression has a high *or* low probability of occurrence.

Consider: No such obvious expression should exist, otherwise the cipher is trivially weak. If we were to randomly select bits for the above expression, it would hold exactly 1/2 the time.

If we find an expression such as above that displays a high *bias*, that is, it holds much more or less frequently than 1/2 the time, we can exploit this.

Basic Idea of Differential Cryptanalysis

Definition (Differential)

Suppose two plaintext inputs to the system be X and X' with corresponding output ciphertexts Y and Y' respectively. Then the pair of input difference $(\Delta X = X \oplus X')$ and the output difference $(\Delta Y = Y \oplus Y')$ is called a *differential* for the system.

Definition (Differential)

Suppose two plaintext inputs to the system be X and X' with corresponding output ciphertexts Y and Y' respectively. Then the pair of input difference ($\Delta X = X \oplus X'$) and the output difference ($\Delta Y = Y \oplus Y'$) is called a *differential* for the system.

Vulnerability of DES:

- S-Boxes show a strong tendency to produce certain differential pairs with high probability, rather than being random

Definition (Differential)

Suppose two plaintext inputs to the system be X and X' with corresponding output ciphertexts Y and Y' respectively. Then the pair of input difference $(\Delta X = X \oplus X')$ and the output difference $(\Delta Y = Y \oplus Y')$ is called a *differential* for the system.

Vulnerability of DES:

- S-Boxes show a strong tendency to produce certain differential pairs with high probability, rather than being random
- The differences do not get diffused fast enough through the permutations

Definition (Differential)

Suppose two plaintext inputs to the system be X and X' with corresponding output ciphertexts Y and Y' respectively. Then the pair of input difference ($\Delta X = X \oplus X'$) and the output difference ($\Delta Y = Y \oplus Y'$) is called a *differential* for the system.

Vulnerability of DES:

- S-Boxes show a strong tendency to produce certain differential pairs with high probability, rather than being random
- The differences do not get diffused fast enough through the permutations
- Differentials are not affected by the round keys as they get XOR-ed out

In an k-round DES, we can trace the path of a certain input difference to get an output difference with a high probability.

In an k-round DES, we can trace the path of a certain input difference to get an output difference with a high probability. This tracing path is known as a *differential characteristic*.

$$\Delta P \to \Delta C^{(1)} \to \Delta C^{(2)} \to \cdots \to \Delta C^{(k-1)} \to \Delta C$$

In an k-round DES, we can trace the path of a certain input difference to get an output difference with a high probability. This tracing path is known as a *differential characteristic*.

$$\Delta P \rightarrow \Delta C^{(1)} \rightarrow \Delta C^{(2)} \rightarrow \cdots \rightarrow \Delta C^{(k-1)} \rightarrow \Delta C$$

In an ideal situation, one will expect the probability

$$P\left(\Delta C | \Delta P\right) = \frac{1}{2^n}$$

for an n-bit cipher system.

In an k-round DES, we can trace the path of a certain input difference to get an output difference with a high probability. This tracing path is known as a *differential characteristic*.

$$\Delta P \rightarrow \Delta C^{(1)} \rightarrow \Delta C^{(2)} \rightarrow \cdots \rightarrow \Delta C^{(k-1)} \rightarrow \Delta C$$

In an ideal situation, one will expect the probability

$$P\left(\Delta C | \Delta P\right) = \frac{1}{2^n}$$

for an n-bit cipher system. Differential attacks seek to exploit a scenario where

$$P\left(\Delta C | \Delta P\right) = p_D \gg \frac{1}{2^n}$$

In an k-round DES, we can trace the path of a certain input difference to get an output difference with a high probability. This tracing path is known as a *differential characteristic*.

$$\Delta P \to \Delta C^{(1)} \to \Delta C^{(2)} \to \cdots \to \Delta C^{(k-1)} \to \Delta C$$

In an ideal situation, one will expect the probability

$$P\left(\Delta C | \Delta P\right) = \frac{1}{2^n}$$

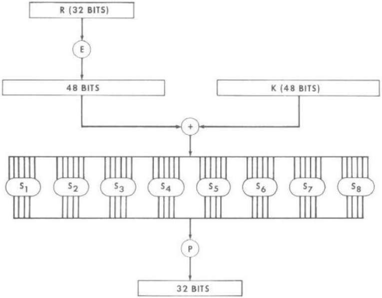
for an n-bit cipher system. Differential attacks seek to exploit a scenario where

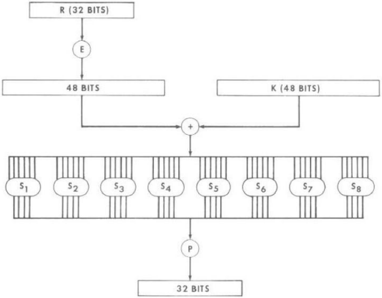
$$P\left(\Delta C | \Delta P\right) = p_D \gg \frac{1}{2^n}$$

Note: This is essentially a chosen-plaintext attack as we want the specific input difference to occur for every pair.

Dive into the Mathematics

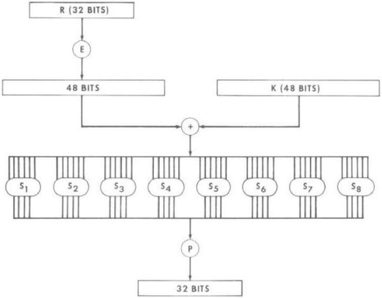
Substitution-Permutation Network





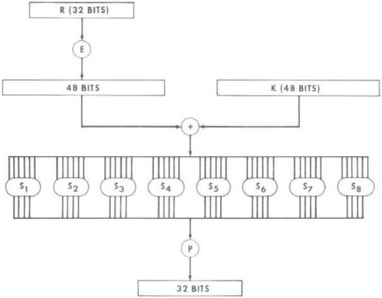
Let us simplify DES by

- Reducing the code length from 64 to 16



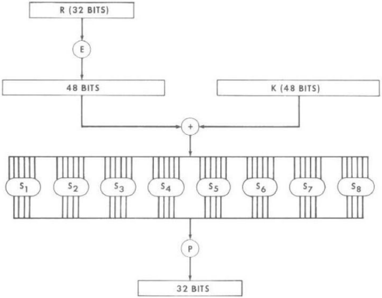
Let us simplify DES by

- Reducing the code length from 64 to 16
- Removing expansion function $E(R)$



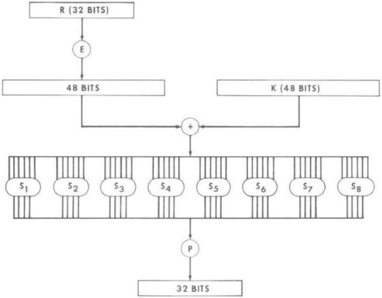
Let us simplify DES by

- Reducing the code length from 64 to 16
- Removing expansion function $E(R)$
- Taking identical S-Boxes of size 4 bits



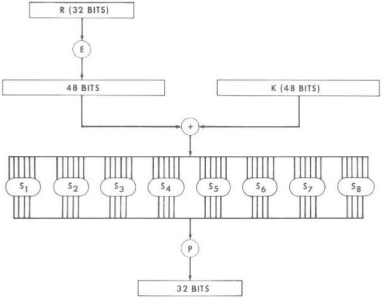
Let us simplify DES by

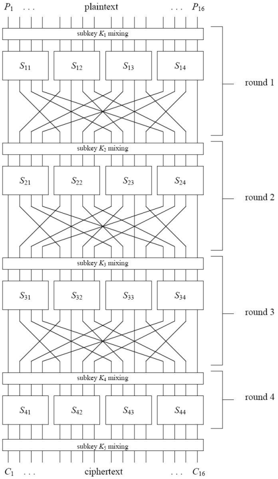
- Reducing the code length from 64 to 16
- Removing expansion function $E(R)$
- Taking identical S-Boxes of size 4 bits
- Repeating over less rounds



Let us simplify DES by

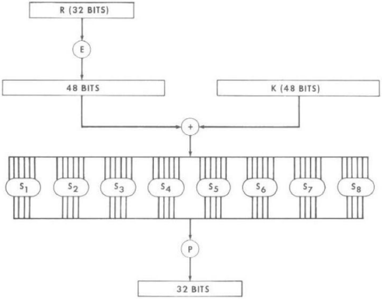
- Reducing the code length from 64 to 16
- Removing expansion function $E(R)$
- Taking identical S-Boxes of size 4 bits
- Repeating over less rounds

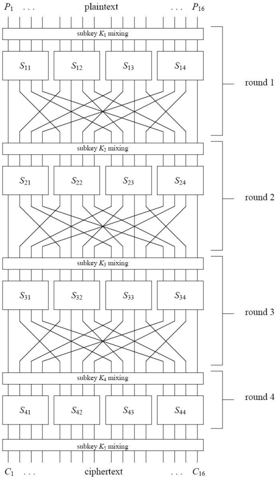




Let us simplify DES by

- Reducing the code length from 64 to 16
- Removing expansion function $E(R)$
- Taking identical S-Boxes of size 4 bits
- Repeating over less rounds





Let us simplify DES by

- Reducing the code length from 64 to 16
- Removing expansion function $E(R)$
- Taking identical S-Boxes of size 4 bits
- Repeating over less rounds

Disclaimer: This simplification does not affect the discussion of the techniques of Linear/Differential Cryptanalysis.

Simple Substitution-Permutation Network

We construct a simple Substitution-Permutation Network Cipher (SPN) which has structural similarity to DES.

- SPN is a 16-bit block cipher (16-bit plaintext to 16-bit ciphertext) with 16-bit round keys

We construct a simple Substitution-Permutation Network Cipher (SPN) which has structural similarity to DES.

- SPN is a 16-bit block cipher (16-bit plaintext to 16-bit ciphertext) with 16-bit round keys
- Each round consists of a substitution and a permutation (much like in DES)

We construct a simple Substitution-Permutation Network Cipher (SPN) which has structural similarity to DES.

- SPN is a 16-bit block cipher (16-bit plaintext to 16-bit ciphertext) with 16-bit round keys
- Each round consists of a substitution and a permutation (much like in DES)
- The substitution is the result of splitting the 16-bit input block to a round in to 4×4-bit sub-blocks. Each 4-bit sub-block is then mapped across a 4-bit to 4-bit S-Box. We use the same S-Box throughout

We construct a simple Substitution-Permutation Network Cipher (SPN) which has structural similarity to DES.

- SPN is a 16-bit block cipher (16-bit plaintext to 16-bit ciphertext) with 16-bit round keys
- Each round consists of a substitution and a permutation (much like in DES)
- The substitution is the result of splitting the 16-bit input block to a round in to 4×4-bit sub-blocks. Each 4-bit sub-block is then mapped across a 4-bit to 4-bit S-Box. We use the same S-Box throughout
- The permutation is applied to all 16-bits of the round following the substitution

We construct a simple Substitution-Permutation Network Cipher (SPN) which has structural similarity to DES.

- SPN is a 16-bit block cipher (16-bit plaintext to 16-bit ciphertext) with 16-bit round keys
- Each round consists of a substitution and a permutation (much like in DES)
- The substitution is the result of splitting the 16-bit input block to a round in to 4×4-bit sub-blocks. Each 4-bit sub-block is then mapped across a 4-bit to 4-bit S-Box. We use the same S-Box throughout
- The permutation is applied to all 16-bits of the round following the substitution
- Finally, key-mixing is achieved by XOR-ing the round key with every input block to a round as well as at the end of the last round (so that we can't simply ignore the last round)

We construct a simple Substitution-Permutation Network Cipher (SPN) which has structural similarity to DES.

- SPN is a 16-bit block cipher (16-bit plaintext to 16-bit ciphertext) with 16-bit round keys
- Each round consists of a substitution and a permutation (much like in DES)
- The substitution is the result of splitting the 16-bit input block to a round in to 4×4-bit sub-blocks. Each 4-bit sub-block is then mapped across a 4-bit to 4-bit S-Box. We use the same S-Box throughout
- The permutation is applied to all 16-bits of the round following the substitution
- Finally, key-mixing is achieved by XOR-ing the round key with every input block to a round as well as at the end of the last round (so that we can't simply ignore the last round)
- The SPN cipher we will look at will consist of 4 rounds

SPN S-Box and Permutation

SPN uses a single 4-bit S-Box that has the following structure:

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

And the following 16-bit permutation:

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

SPN S-Box and Permutation

SPN uses a single 4-bit S-Box that has the following structure:

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

And the following 16-bit permutation:

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

The S-Box provides the *confusion* function and the permutation applies the *diffusion* operation in SPN, thus making it cryptographically similar to DES.

Mathematics of Linear Cryptanalysis

Basic Definitions for the Linear attack

Basic Definitions for the Linear attack

Linear cryptanalysis tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, ciphertext bits and subkey bits. It is a *known plaintext attack*.

Basic Definitions for the Linear attack

Linear cryptanalysis tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, ciphertext bits and subkey bits. It is a *known plaintext attack*.

Define

- P_i where $i = 1, 2, \ldots 16$ as the i-th plaintext bit

Basic Definitions for the Linear attack

Linear cryptanalysis tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, ciphertext bits and subkey bits. It is a *known plaintext attack*.

Define

- P_i where $i = 1, 2, \ldots 16$ as the i-th plaintext bit
- C_i where $i = 1, 2, \ldots 16$ as the i-th ciphertext bit

Basic Definitions for the Linear attack

Linear cryptanalysis tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, ciphertext bits and subkey bits. It is a *known plaintext attack*.

Define

- P_i where $i = 1, 2, \ldots 16$ as the i-th plaintext bit
- C_i where $i = 1, 2, \ldots 16$ as the i-th ciphertext bit
- $U_{j,i}$ where $j = 1, 2, 3, 4$ and $i = 1, 2, \ldots 16$ be the i-th input bit to the j-th round of SPN

Basic Definitions for the Linear attack

Linear cryptanalysis tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, ciphertext bits and subkey bits. It is a *known plaintext attack*.

Define

- P_i where $i = 1, 2, \ldots 16$ as the i-th plaintext bit
- C_i where $i = 1, 2, \ldots 16$ as the i-th ciphertext bit
- $U_{j,i}$ where $j = 1, 2, 3, 4$ and $i = 1, 2, \ldots 16$ be the i-th input bit to the j-th round of SPN
- $V_{j,i}$ where $j = 1, 2, 3, 4$ and $i = 1, 2, \ldots 16$ be the i-th output bit of the j-th round of SPN

Basic Definitions for the Linear attack

Linear cryptanalysis tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, ciphertext bits and subkey bits. It is a *known plaintext attack*.

Define

- P_i where $i = 1, 2, \dots 16$ as the i-th plaintext bit
- C_i where $i = 1, 2, \dots 16$ as the i-th ciphertext bit
- $U_{j,i}$ where $j = 1, 2, 3, 4$ and $i = 1, 2, \dots 16$ be the i-th input bit to the j-th round of SPN
- $V_{j,i}$ where $j = 1, 2, 3, 4$ and $i = 1, 2, \dots 16$ be the i-th output bit of the j-th round of SPN
- $K_{j,i}$ where $j = 1, 2, 3, 4$ and $i = 1, 2, \dots 16$ be the i-th bit of the round key for the j-th round of SPN

Linear and Affine approximation of S-Box

- Question: How do we come up with the desired expression for the entire cipher?

Linear and Affine approximation of S-Box

- Question: How do we come up with the desired expression for the entire cipher?
- Hint: We start by looking at the only non-linear component, the S-Box.

Linear and Affine approximation of S-Box

- Question: How do we come up with the desired expression for the entire cipher?
- Hint: We start by looking at the only non-linear component, the S-Box.

Linear and Affine approximation of S-Box

- Question: How do we come up with the desired expression for the entire cipher?
- Hint: We start by looking at the only non-linear component, the S-Box.

To find the linear/affine approximation of the S-Box we simply consider every possible expression of the input bits X_i and output bits Y_j.

Linear and Affine approximation of S-Box

- Question: How do we come up with the desired expression for the entire cipher?
- Hint: We start by looking at the only non-linear component, the S-Box.

To find the linear/affine approximation of the S-Box we simply consider every possible expression of the input bits X_i and output bits Y_j. Thus the expression has the form

$$\bigoplus_{i \in U} X_i = \bigoplus_{j \in V} Y_j$$

where U and V range over all possible subsets of $\{1, 2, 3, 4\}$.

Linear and Affine approximation of S-Box

- Question: How do we come up with the desired expression for the entire cipher?
- Hint: We start by looking at the only non-linear component, the S-Box.

To find the linear/affine approximation of the S-Box we simply consider every possible expression of the input bits X_i and output bits Y_j. Thus the expression has the form

$$\bigoplus_{i \in U} X_i = \bigoplus_{j \in V} Y_j$$

where U and V range over all possible subsets of $\{1, 2, 3, 4\}$. We then compare how often this expression coincides with the S-Box.

Linear and Affine approximation of S-Box

- Question: How do we come up with the desired expression for the entire cipher?
- Hint: We start by looking at the only non-linear component, the S-Box.

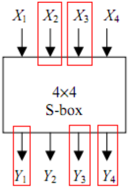
To find the linear/affine approximation of the S-Box we simply consider every possible expression of the input bits X_i and output bits Y_j. Thus the expression has the form

$$\bigoplus_{i \in U} X_i = \bigoplus_{j \in V} Y_j$$

where U and V range over all possible subsets of $\{1, 2, 3, 4\}$. We then compare how often this expression coincides with the S-Box.

Note that there are 16 possibilities for U and V, hence 256 total possible expressions (for a 4-bit S-box).

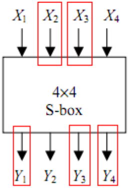
S-Box Approximation Example



We can take the following expression as an example:

$$X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4$$

S-Box Approximation Example

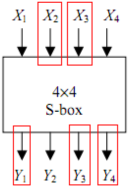


We can take the following expression as an example:

$$X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4$$

Applying all possible values for the input X bits it turns out that the expression holds in 12 out of the 16 cases.

S-Box Approximation Example

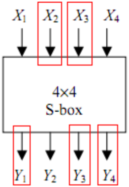


We can take the following expression as an example:

$$X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4$$

Applying all possible values for the input X bits it turns out that the expression holds in 12 out of the 16 cases. Hence, this expression has a bias of $12/16 - 1/2 = 1/4$.

S-Box Approximation Example



We can take the following expression as an example:

$$X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4$$

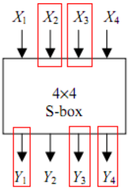
Applying all possible values for the input X bits it turns out that the expression holds in 12 out of the 16 cases. Hence, this expression has a bias of $12/16 - 1/2 = 1/4$.

Similarly the following expression

$$X_3 \oplus X_4 = Y_1 \oplus Y_4$$

Has a bias of $-3/8$, which is higher in magnitude than the previous one.

S-Box Approximation Example



We can take the following expression as an example:

$$X_2 \oplus X_3 = Y_1 \oplus Y_3 \oplus Y_4$$

Applying all possible values for the input X bits it turns out that the expression holds in 12 out of the 16 cases. Hence, this expression has a bias of $12/16 - 1/2 = 1/4$.

Similarly the following expression

$$X_3 \oplus X_4 = Y_1 \oplus Y_4$$

Has a bias of $-3/8$, which is higher in magnitude than the previous one. It is the magnitude of the bias that is important.

The number of agreements (minus 8) between the S-Box and every possible expression is summarized in the table below. Thus to get the bias, one must only divide by 16.

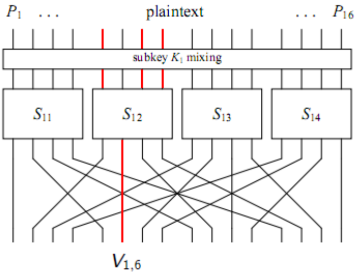
Input Sum \ Output Sum	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	+8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	−2	−2	0	0	−2	+6	+2	+2	0	0	+2	+2	0	0
2	0	0	−2	−2	0	0	−2	−2	0	0	+2	+2	0	0	−6	+2
3	0	0	0	0	0	0	0	0	+2	−6	−2	−2	+2	+2	−2	−2
4	0	+2	0	−2	−2	−4	−2	0	0	−2	0	+2	+2	−4	+2	0
5	0	−2	−2	0	−2	0	+4	+2	−2	0	−4	+2	0	−2	−2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

What this means for 1 Round:

Note, from the previous table, that the expression

$$X_1 \oplus X_3 \oplus X_4 = Y_2$$

has a bias of $+1/4$.



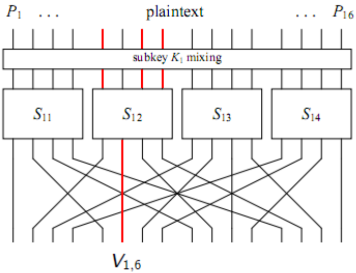
What this means for 1 Round:

Note, from the previous table, that the expression

$$X_1 \oplus X_3 \oplus X_4 = Y_2$$

has a bias of $+1/4$.

Note also that $U_1 = P \oplus K_1$.



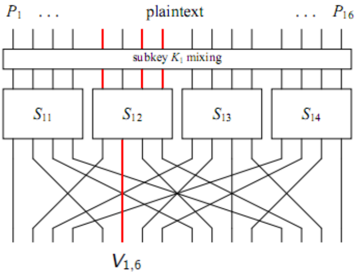
What this means for 1 Round:

Note, from the previous table, that the expression

$$X_1 \oplus X_3 \oplus X_4 = Y_2$$

has a bias of $+1/4$.

Note also that $U_1 = P \oplus K_1$.



We can now write down the following linear approximation across the 1st round of SPN:

$$\begin{aligned} V_{1,6} &= U_{1,5} \oplus U_{1,7} \oplus U_{1,8} \quad \leftarrow \text{S-Box } S_{12} \text{ approximation above} \\ &= (P_5 \oplus K_{1,5}) \oplus (P_7 \oplus K_{1,7}) \oplus (P_8 \oplus K_{1,8}) \end{aligned}$$

This expression holds with probability of $3/4$ (bias of $+1/4$)

We can get expressions that hold with some non-1/2 probability for every round. But we must somehow combine them to write an expression relating the plaintext and ciphertext bits.

We can get expressions that hold with some non-1/2 probability for every round. But we must somehow combine them to write an expression relating the plaintext and ciphertext bits. We use the Piling Up Principle for this purpose.

We can get expressions that hold with some non-1/2 probability for every round. But we must somehow combine them to write an expression relating the plaintext and ciphertext bits. We use the Piling Up Principle for this purpose.

Note:

$X_1 \oplus X_2 = 0$ is a linear expression equivalent to $X_1 = X_2$

$X_1 \oplus X_2 = 1$ is an affine expression equivalent to $X_1 \neq X_2$

We can get expressions that hold with some non-1/2 probability for every round. But we must somehow combine them to write an expression relating the plaintext and ciphertext bits. We use the Piling Up Principle for this purpose.

Note:

$X_1 \oplus X_2 = 0$ is a linear expression equivalent to $X_1 = X_2$
$X_1 \oplus X_2 = 1$ is an affine expression equivalent to $X_1 \neq X_2$

Assume the following probability distribution

$$Pr(X_1 = i) = \begin{cases} p_1 & i = 0 \\ (1 - p_1) & i = 1. \end{cases}$$

$$Pr(X_2 = i) = \begin{cases} p_2 & i = 0; \\ (1 - p_2) & i = 1. \end{cases}$$

Piling Up Principle (continued):

Assuming that X_1 and X_2 are independent, we get

$$Pr(X_1 = i, X_2 = j) = \begin{cases} p_1 p_2 & i = 0, j = 0; \\ (1 - p_1)p_2 & i = 1, j = 0 \\ p_1(1 - p_2) & i = 0, j = 1 \\ (1 - p_1)(1 - p_2) & i = 1, j = 1. \end{cases}$$

Piling Up Principle (continued):

Assuming that X_1 and X_2 are independent, we get

$$Pr(X_1 = i, X_2 = j) = \begin{cases} p_1 p_2 & i = 0, j = 0; \\ (1 - p_1)p_2 & i = 1, j = 0 \\ p_1(1 - p_2) & i = 0, j = 1 \\ (1 - p_1)(1 - p_2) & i = 1, j = 1. \end{cases}$$

Now, note that $X_1 \oplus X_2 = 0 \Rightarrow X_1 = 1, X_2 = 1$ OR $X_1 = 0, X_2 = 0$.

Piling Up Principle (continued):

Assuming that X_1 and X_2 are independent, we get

$$Pr(X_1 = i, X_2 = j) = \begin{cases} p_1 p_2 & i = 0, j = 0; \\ (1 - p_1) p_2 & i = 1, j = 0 \\ p_1 (1 - p_2) & i = 0, j = 1 \\ (1 - p_1)(1 - p_2) & i = 1, j = 1. \end{cases}$$

Now, note that $X_1 \oplus X_2 = 0 \Rightarrow X_1 = 1, X_2 = 1$ OR $X_1 = 0, X_2 = 0$. Hence,

$$\begin{aligned} Pr(X_1 \oplus X_2 = 0) &= Pr(X_1 = 0, X_2 = 0) + Pr(X_1 = 1, X_2 = 1) \\ &= p_1 p_2 + (1 - p_1)(1 - p_2) \end{aligned}$$

Piling Up Principle (continued):

Assuming that X_1 and X_2 are independent, we get

$$Pr(X_1 = i, X_2 = j) = \begin{cases} p_1 p_2 & i = 0, j = 0; \\ (1 - p_1)p_2 & i = 1, j = 0 \\ p_1(1 - p_2) & i = 0, j = 1 \\ (1 - p_1)(1 - p_2) & i = 1, j = 1. \end{cases}$$

Now, note that $X_1 \oplus X_2 = 0 \Rightarrow X_1 = 1, X_2 = 1$ OR $X_1 = 0, X_2 = 0$. Hence,

$$\begin{aligned} Pr(X_1 \oplus X_2 = 0) &= Pr(X_1 = 0, X_2 = 0) + Pr(X_1 = 1, X_2 = 1) \\ &= p_1 p_2 + (1 - p_1)(1 - p_2) \end{aligned}$$

If we write $p_k = 1/2 + \epsilon_k$ we have

$$Pr(X_1 \oplus X_2 = 0) = 1/2 + 2\epsilon_1\epsilon_2$$

Piling Up Principle (continued):

Assuming that X_1 and X_2 are independent, we get

$$Pr(X_1 = i, X_2 = j) = \begin{cases} p_1 p_2 & i = 0, j = 0; \\ (1 - p_1)p_2 & i = 1, j = 0 \\ p_1(1 - p_2) & i = 0, j = 1 \\ (1 - p_1)(1 - p_2) & i = 1, j = 1. \end{cases}$$

Now, note that $X_1 \oplus X_2 = 0 \Rightarrow X_1 = 1, X_2 = 1$ OR $X_1 = 0, X_2 = 0$. Hence,

$$\begin{aligned} Pr(X_1 \oplus X_2 = 0) &= Pr(X_1 = 0, X_2 = 0) + Pr(X_1 = 1, X_2 = 1) \\ &= p_1 p_2 + (1 - p_1)(1 - p_2) \end{aligned}$$

If we write $p_k = 1/2 + \epsilon_k$ we have

$$Pr(X_1 \oplus X_2 = 0) = 1/2 + 2\epsilon_1\epsilon_2$$

Hence the bias of $X_1 \oplus X_2 = 0$ is

$$2\epsilon_1\epsilon_2$$

That is, twice the product of the bias of the original expressions.

Piling Up Principle (continued) - The Piling Up Lemma:

We extend the idea on the previous slide to the general case

Piling Up Principle (continued) - The Piling Up Lemma:

We extend the idea on the previous slide to the general case : For n independent random binary variables, $X_1, X_2, \ldots, X_n$,

$$Pr(X_1 \oplus X_2 \cdots \oplus X_n = 0) = 1/2 + 2^{n-1} \prod_{i=1}^{n} \epsilon_i$$

Piling Up Principle (continued) - The Piling Up Lemma:

We extend the idea on the previous slide to the general case : For n independent random binary variables, $X_1, X_2, \ldots, X_n$,

$$Pr(X_1 \oplus X_2 \cdots \oplus X_n = 0) = 1/2 + 2^{n-1} \prod_{i=1}^{n} \epsilon_i$$

In terms of bias,

$$\epsilon_{1,2,\ldots,n} = 2^{n-1} \prod_{i=1}^{n} \epsilon_i$$

Where $\epsilon_{1,2,\ldots,n}$ is the total bias of $X_1 \oplus X_2 \cdots \oplus X_n = 0$.

Piling Up Principle (continued) - The Piling Up Lemma:

We extend the idea on the previous slide to the general case : For n independent random binary variables, $X_1, X_2, \ldots, X_n$,

$$Pr(X_1 \oplus X_2 \cdots \oplus X_n = 0) = 1/2 + 2^{n-1} \prod_{i=1}^{n} \epsilon_i$$

In terms of bias,

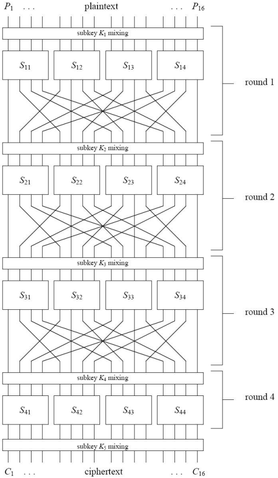
$$\epsilon_{1,2,\ldots,n} = 2^{n-1} \prod_{i=1}^{n} \epsilon_i$$

Where $\epsilon_{1,2,\ldots,n}$ is the total bias of $X_1 \oplus X_2 \cdots \oplus X_n = 0$.

Note that if some $\epsilon_k = 0$ (that is, some X_k has no bias) then total bias will be 0. More about the probabilistic analysis later.

Now, come back to per-round approximations. We can write down 4 approximations (S_{ij} represents the j-th S-Box in the i-th round):

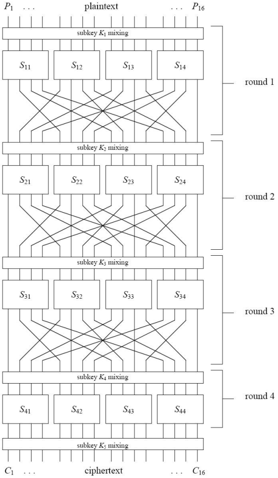
$$S_{12} : X_1 \oplus X_3 \oplus X_4 = Y_2$$
$$S_{22} : X_2 = Y_2 \oplus Y_4$$
$$S_{32} : X_2 = Y_2 \oplus Y_4$$
$$S_{34} : X_2 = Y_2 \oplus Y_4$$



Now, come back to per-round approximations. We can write down 4 approximations (S_{ij} represents the j-th S-Box in the i-th round):

$$S_{12} : X_1 \oplus X_3 \oplus X_4 = Y_2$$
$$S_{22} : X_2 = Y_2 \oplus Y_4$$
$$S_{32} : X_2 = Y_2 \oplus Y_4$$
$$S_{34} : X_2 = Y_2 \oplus Y_4$$

Each of these has a probability bias magnitude of 1/4. We can use the Piling Up Principle to combine them into a single expression relating plaintext bits to ciphertext bits.

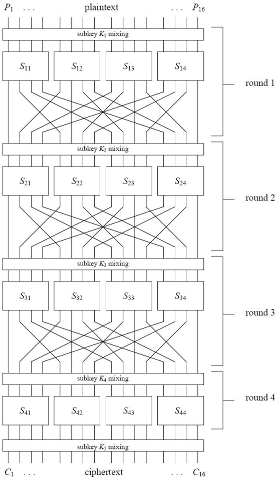


Consider the first 2 rounds:

$$X_1 \oplus X_3 \oplus X_4 = Y_2$$

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

$$X_2 = Y_2 \oplus Y_4 \Rightarrow (V_{1,6} \oplus K_{2,6}) = V_{2,6} \oplus V_{2,8}$$



Consider the first 2 rounds:

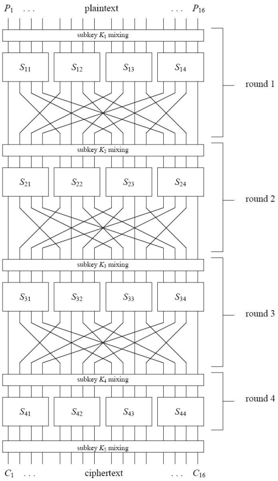
$$X_1 \oplus X_3 \oplus X_4 = Y_2$$

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

$$X_2 = Y_2 \oplus Y_4 \;\Rightarrow\; (V_{1,6} \oplus K_{2,6}) = V_{2,6} \oplus V_{2,8}$$

Each of these has a bias of magnitude 1/4 and we can combine to obtain:

$$V_{2,6} \oplus V_{2,8} \oplus P_5 \oplus P_7 \oplus P_8 \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} = 0$$



Consider the first 2 rounds:

$$X_1 \oplus X_3 \oplus X_4 = Y_2$$

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

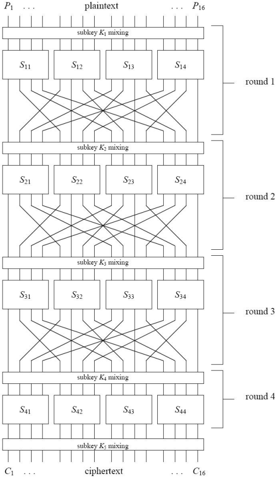
$$X_2 = Y_2 \oplus Y_4 \;\Rightarrow\; (V_{1,6} \oplus K_{2,6}) = V_{2,6} \oplus V_{2,8}$$

Each of these has a bias of magnitude $1/4$ and we can combine to obtain:

$$V_{2,6} \oplus V_{2,8} \oplus P_5 \oplus P_7 \oplus P_8 \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} = 0$$

By the Piling Up Principle this holds with bias

$$2 \times 1/4 \times 1/4 = 1/8$$



Consider the first 2 rounds:

$$X_1 \oplus X_3 \oplus X_4 = Y_2$$
$$\Rightarrow (P_5 \oplus K_{1,5}) \oplus (P_7 \oplus K_{1,7}) \oplus (P_8 \oplus K_{1,8}) = V_{1,6}$$

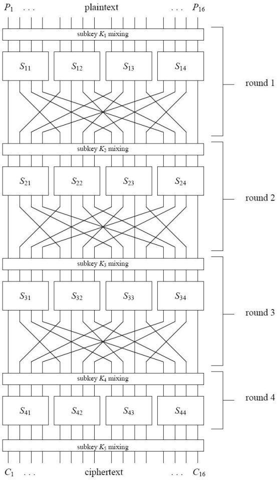
$$X_2 = Y_2 \oplus Y_4 \;\Rightarrow\; (V_{1,6} \oplus K_{2,6}) = V_{2,6} \oplus V_{2,8}$$

Each of these has a bias of magnitude $1/4$ and we can combine to obtain:

$$V_{2,6} \oplus V_{2,8} \oplus P_5 \oplus P_7 \oplus P_8 \oplus K_{1,5} \oplus K_{1,7} \oplus K_{1,8} \oplus K_{2,6} = 0$$

By the Piling Up Principle this holds with bias

$$2 \times 1/4 \times 1/4 = 1/8$$



Note the implicit assumption that S-Boxes are independent. More on this later.

Using this principle we can write the following equation over 3 rounds of SPN:

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

Using this principle we can write the following equation over 3 rounds of SPN:

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

Where ΣK is the sum over some key bits. Note that since the key is fixed, $\Sigma K = 0$ or 1 and thus we can ignore it since we only care about the bias.

Using this principle we can write the following equation over 3 rounds of SPN:

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

Where ΣK is the sum over some key bits. Note that since the key is fixed, $\Sigma K = 0$ or 1 and thus we can ignore it since we only care about the bias.

The magnitude of the bias of the above expression, by the Piling Up Principle, is $1/32$.

Using this principle we can write the following equation over 3 rounds of SPN:

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

Where ΣK is the sum over some key bits. Note that since the key is fixed, $\Sigma K = 0$ or 1 and thus we can ignore it since we only care about the bias.

The magnitude of the bias of the above expression, by the Piling Up Principle, is $1/32$.

Next we show how we can extract key bits using this information.

Attack Idea:

We have an expression that links plaintext bits to input bits to the 4th round of SPN that holds with high bias (1/32).

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

Attack Idea:

We have an expression that links plaintext bits to input bits to the 4th round of SPN that holds with high bias (1/32).

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

We can partially undo the last round by guessing the last key.

Attack Idea:

We have an expression that links plaintext bits to input bits to the 4th round of SPN that holds with high bias (1/32).

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

We can partially undo the last round by guessing the last key.

If we guess correctly, the equation will hold with high bias. If we guess wrong, the equation will probably hold with probability close to 1/2, that is, a bias close to 0.

Attack Idea:

We have an expression that links plaintext bits to input bits to the 4th round of SPN that holds with high bias (1/32).

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

We can partially undo the last round by guessing the last key.

If we guess correctly, the equation will hold with high bias. If we guess wrong, the equation will probably hold with probability close to 1/2, that is, a bias close to 0.

BUT To do this, we don't need to guess the entire key for the last round!

Attack Idea:

We have an expression that links plaintext bits to input bits to the 4th round of SPN that holds with high bias (1/32).

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

We can partially undo the last round by guessing the last key.

If we guess correctly, the equation will hold with high bias. If we guess wrong, the equation will probably hold with probability close to 1/2, that is, a bias close to 0.

BUT To do this, we don't need to guess the entire key for the last round! Our expression only involves 4 fourth round input (U_4) bits, output from 2 S-Boxes of the third round. Thus we only need to guess $2^8 = 256$ values.

Attack Idea:

We have an expression that links plaintext bits to input bits to the 4th round of SPN that holds with high bias (1/32).

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

We can partially undo the last round by guessing the last key.

If we guess correctly, the equation will hold with high bias. If we guess wrong, the equation will probably hold with probability close to 1/2, that is, a bias close to 0.

BUT To do this, we don't need to guess the entire key for the last round! Our expression only involves 4 fourth round input (U_4) bits, output from 2 S-Boxes of the third round. Thus we only need to guess $2^8 = 256$ values.

For each value of the guessed *target partial subkey* we can undo the last round and determine the bias of the equation. Highest bias indicates likely correct guess.

SPN Linear Approximation

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

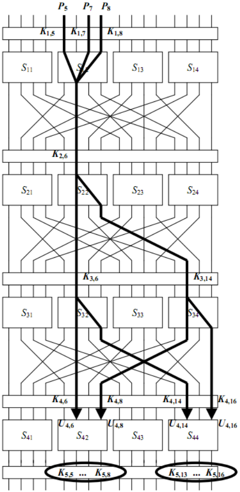


Figure: Linear Attack on SPN

Probabilistic justification for number of pairs:

Probabilistic justification for number of pairs:

In general, the number of plaintext-ciphertext pairs needed is related inversely-quadratically to the bias. That is

$$N_{pairs} \approx 1/\epsilon^2$$

Probabilistic justification for number of pairs:

In general, the number of plaintext-ciphertext pairs needed is related inversely-quadratically to the bias. That is

$$N_{pairs} \approx 1/\epsilon^2$$

For the SPN cipher approximation with bias $1/32$ we need about 1000 pairs to perform the attack with near full probability of success.

DEMO: Attack on 4-round SPN

Mathematics of Differential Cryptanalysis

Differential: $(\Delta P, \Delta C) = (P \oplus P', C \oplus C')$

Differential: $(\Delta P, \Delta C) = (P \oplus P', C \oplus C')$

Differential Cryptanalysis tries to exploit the high probability of certain occurrences of differential characteristics $(\Delta P \rightarrow \Delta C)$ in the cipher.

Differential: $(\Delta P, \Delta C) = (P \oplus P', C \oplus C')$

Differential Cryptanalysis tries to exploit the high probability of certain occurrences of differential characteristics $(\Delta P \rightarrow \Delta C)$ in the cipher.

Note: This is a chosen plaintext attack.

Differential: $(\Delta P, \Delta C) = (P \oplus P', C \oplus C')$

Differential Cryptanalysis tries to exploit the high probability of certain occurrences of differential characteristics ($\Delta P \rightarrow \Delta C$) in the cipher.

Note: This is a chosen plaintext attack.

We have to analyze the cipher in pieces.

Differential: $(\Delta P, \Delta C) = (P \oplus P', C \oplus C')$

Differential Cryptanalysis tries to exploit the high probability of certain occurrences of differential characteristics $(\Delta P \rightarrow \Delta C)$ in the cipher.

Note: This is a chosen plaintext attack.

We have to analyze the cipher in pieces. The basic idea is

- Analyze the difference pattern and their probabilities of the S-Box

Differential: $(\Delta P, \Delta C) = (P \oplus P', C \oplus C')$

Differential Cryptanalysis tries to exploit the high probability of certain occurrences of differential characteristics $(\Delta P \rightarrow \Delta C)$ in the cipher.

Note: This is a chosen plaintext attack.

We have to analyze the cipher in pieces. The basic idea is

- Analyze the difference pattern and their probabilities of the S-Box
- Combine the S-Boxes over the rounds to build the differential characteristic of the cipher

Differential: $(\Delta P, \Delta C) = (P \oplus P', C \oplus C')$

Differential Cryptanalysis tries to exploit the high probability of certain occurrences of differential characteristics $(\Delta P \rightarrow \Delta C)$ in the cipher.

Note: This is a chosen plaintext attack.

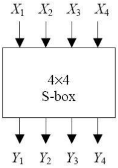
We have to analyze the cipher in pieces. The basic idea is

- Analyze the difference pattern and their probabilities of the S-Box
- Combine the S-Boxes over the rounds to build the differential characteristic of the cipher
- Choose the characteristic with maximum probability and use that to retrieve subkey bits of the cipher

Analyzing the S-Box

Let us consider the S-Box of the SPN cipher as we constructed it:

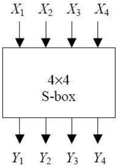
- 4×4 S-Box
- Input: $X = [X_1\ X_2\ X_3\ X_4]$
- Output: $Y = [Y_1\ Y_2\ Y_3\ Y_4]$
- Difference pair: $(\Delta X, \Delta Y)$



Analyzing the S-Box

Let us consider the S-Box of the SPN cipher as we constructed it:

- 4×4 S-Box
- Input: $X = [X_1\ X_2\ X_3\ X_4]$
- Output: $Y = [Y_1\ Y_2\ Y_3\ Y_4]$
- Difference pair: $(\Delta X, \Delta Y)$



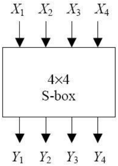
Algorithm

1. Fix a certain input difference $\Delta X = 0100$, say

Analyzing the S-Box

Let us consider the S-Box of the SPN cipher as we constructed it:

- 4×4 S-Box
- Input: $X = [X_1\ X_2\ X_3\ X_4]$
- Output: $Y = [Y_1\ Y_2\ Y_3\ Y_4]$
- Difference pair: $(\Delta X, \Delta Y)$



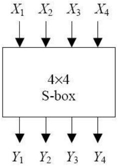
Algorithm

1. Fix a certain input difference $\Delta X = 0100$, say
2. Iterate through the first input $X = \{0000, ..., 1111\}$

Analyzing the S-Box

Let us consider the S-Box of the SPN cipher as we constructed it:

- 4×4 S-Box
- Input: $X = [X_1\ X_2\ X_3\ X_4]$
- Output: $Y = [Y_1\ Y_2\ Y_3\ Y_4]$
- Difference pair: $(\Delta X, \Delta Y)$



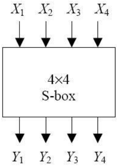
Algorithm

1. Fix a certain input difference $\Delta X = 0100$, say
2. Iterate through the first input $X = \{0000, ..., 1111\}$
3. This fixes the second input $X' = X \oplus \Delta X = \{0100, ..., 1011\}$ (ordering does not matter)

Analyzing the S-Box

Let us consider the S-Box of the SPN cipher as we constructed it:

- 4×4 S-Box
- Input: $X = [X_1\ X_2\ X_3\ X_4]$
- Output: $Y = [Y_1\ Y_2\ Y_3\ Y_4]$
- Difference pair: $(\Delta X, \Delta Y)$



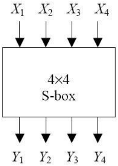
Algorithm

1. Fix a certain input difference $\Delta X = 0100$, say
2. Iterate through the first input $X = \{0000, ..., 1111\}$
3. This fixes the second input $X' = X \oplus \Delta X = \{0100, ..., 1011\}$ (ordering does not matter)
4. Obtain corresponding output difference $\Delta Y = Y \oplus Y'$

Analyzing the S-Box

Let us consider the S-Box of the SPN cipher as we constructed it:

- 4×4 S-Box
- Input: $X = [X_1 \ X_2 \ X_3 \ X_4]$
- Output: $Y = [Y_1 \ Y_2 \ Y_3 \ Y_4]$
- Difference pair: $(\Delta X, \Delta Y)$



Algorithm

1. Fix a certain input difference $\Delta X = 0100$, say
2. Iterate through the first input $X = \{0000, ..., 1111\}$
3. This fixes the second input $X' = X \oplus \Delta X = \{0100, ..., 1011\}$ (ordering does not matter)
4. Obtain corresponding output difference $\Delta Y = Y \oplus Y'$
5. Iterate through steps 2 to 4 for $\Delta X = \{0000, ..., 1111\}$

Sample Difference Pairs for the S-Box

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

Sample Difference Pairs for the S-Box

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

Note that, for $\Delta X = 1011$, $\Delta Y = 0010$ occurs 8 times, out of the possible 16 times. So, the pair $(1011, 0010)$ has a probability of occurrence $8/16 = 1/2$.

Sample Difference Pairs for the S-Box

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

Note that, for $\Delta X = 1011$, $\Delta Y = 0010$ occurs 8 times, out of the possible 16 times. So, the pair $(1011, 0010)$ has a probability of occurrence $8/16 = 1/2$.

We can tabulate the complete data for the S-Box to check this probabilities.

Difference Distribution Table

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

Difference Distribution Table

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	$\boxed{8}$	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

Note that

- In an ideal S-Box, we would like all entries to be 1

Difference Distribution Table

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

Note that

- In an ideal S-Box, we would like all entries to be 1
- But this table clearly shows bias towards certain pairs

Difference Distribution Table

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

Note that

- In an ideal S-Box, we would like all entries to be 1
- But this table clearly shows bias towards certain pairs
- Just divide the entries by $2^4 = 16$ to get the probabilities, and exploit the scenario of highest probability of occurrence ($Pr(2|B) = 8/16 = 1/2$ here)

A few properties of the S-Box Distribution Table

A few properties of the S-Box Distribution Table

Notice that in the difference table

- The $(0, 0)$ entry is 16, just because identical inputs ($\Delta X = 0 \Rightarrow X = X'$) should produce identical outputs ($Y = Y' \Rightarrow \Delta Y = 0$)

A few properties of the S-Box Distribution Table

Notice that in the difference table

- The $(0, 0)$ entry is 16, just because identical inputs ($\Delta X = 0 \Rightarrow X = X'$) should produce identical outputs ($Y = Y' \Rightarrow \Delta Y = 0$)
- All the entries are even, because ΔX is the same for both the input pairs (X, X') and (X', X), producing same ΔY

A few properties of the S-Box Distribution Table

Notice that in the difference table

- The $(0, 0)$ entry is 16, just because identical inputs ($\Delta X = 0 \Rightarrow X = X'$) should produce identical outputs ($Y = Y' \Rightarrow \Delta Y = 0$)
- All the entries are even, because ΔX is the same for both the input pairs (X, X') and (X', X), producing same ΔY
- Sum of all entries in a row is $2^4 = 16$

A few properties of the S-Box Distribution Table

Notice that in the difference table

- The $(0, 0)$ entry is 16, just because identical inputs $(\Delta X = 0 \Rightarrow X = X')$ should produce identical outputs $(Y = Y' \Rightarrow \Delta Y = 0)$
- All the entries are even, because ΔX is the same for both the input pairs (X, X') and (X', X), producing same ΔY
- Sum of all entries in a row is $2^4 = 16$

A few properties of the S-Box Distribution Table

Notice that in the difference table

- The $(0, 0)$ entry is 16, just because identical inputs ($\Delta X = 0 \Rightarrow X = X'$) should produce identical outputs ($Y = Y' \Rightarrow \Delta Y = 0$)
- All the entries are even, because ΔX is the same for both the input pairs (X, X') and (X', X), producing same ΔY
- Sum of all entries in a row is $2^4 = 16$

As an ideal S-Box expects to give out no information of ΔY given ΔX, we wish it could have all entries 1, i.e,

$$Pr(\Delta Y|\Delta X) = \frac{1}{16} \quad \forall \, \Delta X$$

A few properties of the S-Box Distribution Table

Notice that in the difference table

- The $(0, 0)$ entry is 16, just because identical inputs ($\Delta X = 0 \Rightarrow X = X'$) should produce identical outputs ($Y = Y' \Rightarrow \Delta Y = 0$)
- All the entries are even, because ΔX is the same for both the input pairs (X, X') and (X', X), producing same ΔY
- Sum of all entries in a row is $2^4 = 16$

As an ideal S-Box expects to give out no information of ΔY given ΔX, we wish it could have all entries 1, i.e,

$$Pr(\Delta Y | \Delta X) = \frac{1}{16} \quad \forall \, \Delta X$$

But, from our discussion above, this is infeasible, and we hope to exploit this.

What happens for a Keyed S-Box?

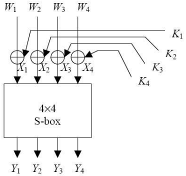
What happens for a Keyed S-Box?

Let us consider the S-Box of the SPN cipher, with the round keys this time.

What happens for a Keyed S-Box?

Let us consider the S-Box of the SPN cipher, with the round keys this time.

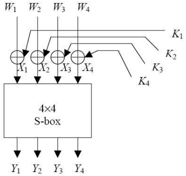
- 4×4-bit S-Box
- Input: $W = [W_1\ W_2\ W_3\ W_4]$
- Round Key: $K = [K_1\ K_2\ K_3\ K_4]$
- Output: $Y = [Y_1\ Y_2\ Y_3\ Y_4]$
- Difference pair: $(\Delta W, \Delta Y)$



What happens for a Keyed S-Box?

Let us consider the S-Box of the SPN cipher, with the round keys this time.

- 4×4-bit S-Box
- Input: $W = [W_1\ W_2\ W_3\ W_4]$
- Round Key: $K = [K_1\ K_2\ K_3\ K_4]$
- Output: $Y = [Y_1\ Y_2\ Y_3\ Y_4]$
- Difference pair: $(\Delta W, \Delta Y)$



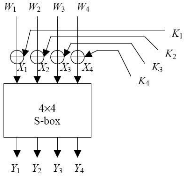
Note that for each pair of inputs (W, W'), the actual inputs to the S-Box are $(X, X') = (W \oplus K, W' \oplus K)$. Hence, the input difference for the keyed S-Box is

$$\Delta W = W \oplus W' = (X \oplus K) \oplus (X' \oplus K) = X \oplus X' = \Delta X$$

What happens for a Keyed S-Box?

Let us consider the S-Box of the SPN cipher, with the round keys this time.

- 4×4-bit S-Box
- Input: $W = [W_1\ W_2\ W_3\ W_4]$
- Round Key: $K = [K_1\ K_2\ K_3\ K_4]$
- Output: $Y = [Y_1\ Y_2\ Y_3\ Y_4]$
- Difference pair: $(\Delta W, \Delta Y)$



Note that for each pair of inputs (W, W'), the actual inputs to the S-Box are $(X, X') = (W \oplus K, W' \oplus K)$. Hence, the input difference for the keyed S-Box is

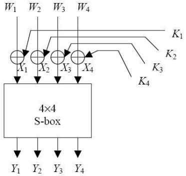
$$\Delta W = W \oplus W' = (X \oplus K) \oplus (X' \oplus K) = X \oplus X' = \Delta X$$

and hence $\Delta W \rightarrow \Delta Y$ corresponds identically to $\Delta X \rightarrow \Delta Y$.

What happens for a Keyed S-Box?

Let us consider the S-Box of the SPN cipher, with the round keys this time.

- 4×4-bit S-Box
- Input: $W = [W_1\ W_2\ W_3\ W_4]$
- Round Key: $K = [K_1\ K_2\ K_3\ K_4]$
- Output: $Y = [Y_1\ Y_2\ Y_3\ Y_4]$
- Difference pair: $(\Delta W, \Delta Y)$



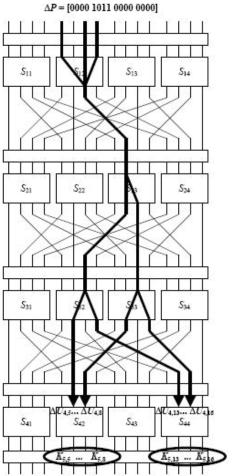
Note that for each pair of inputs (W, W'), the actual inputs to the S-Box are $(X, X') = (W \oplus K, W' \oplus K)$. Hence, the input difference for the keyed S-Box is

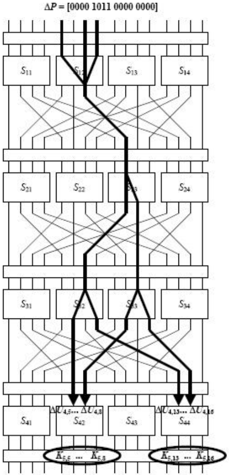
$$\Delta W = W \oplus W' = (X \oplus K) \oplus (X' \oplus K) = X \oplus X' = \Delta X$$

and hence $\Delta W \rightarrow \Delta Y$ corresponds identically to $\Delta X \rightarrow \Delta Y$.

Thus, the keyed S-Box has the same difference distribution table.

Constructing a Differential Characteristic

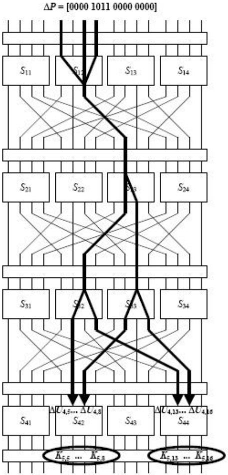




Constructing a Differential Characteristic

The diagram beside illustrates the tracing of the non-zero bits of the input difference through the SPN structure.

Input Difference: $\Delta P = 0B00$



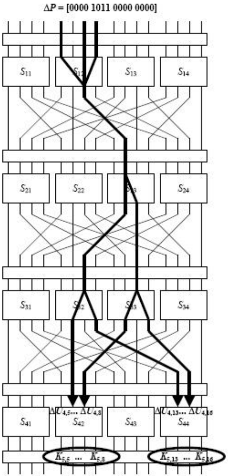
Constructing a Differential Characteristic

The diagram beside illustrates the tracing of the non-zero bits of the input difference through the SPN structure.

Input Difference: $\Delta P = 0B00$

Propagating Difference Pairs

- S_{12}: $1011 \rightarrow 0010$ $[Pr = \frac{8}{16}]$



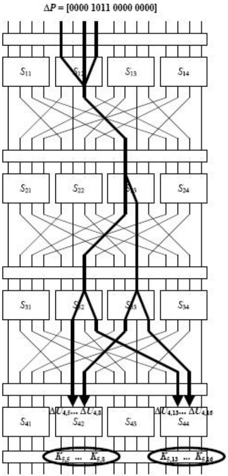
Constructing a Differential Characteristic

The diagram beside illustrates the tracing of the non-zero bits of the input difference through the SPN structure.

Input Difference: $\Delta P = 0B00$

Propagating Difference Pairs

- S_{12}: $1011 \rightarrow 0010$ $[Pr = \frac{8}{16}]$
- S_{23}: $0100 \rightarrow 0110$ $[Pr = \frac{6}{16}]$



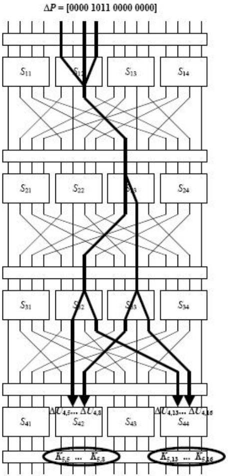
Constructing a Differential Characteristic

The diagram beside illustrates the tracing of the non-zero bits of the input difference through the SPN structure.

Input Difference: $\Delta P = 0B00$

Propagating Difference Pairs

- S_{12}: $1011 \rightarrow 0010$ $[Pr = \frac{8}{16}]$
- S_{23}: $0100 \rightarrow 0110$ $[Pr = \frac{6}{16}]$
- S_{32}: $0010 \rightarrow 0101$ $[Pr = \frac{6}{16}]$



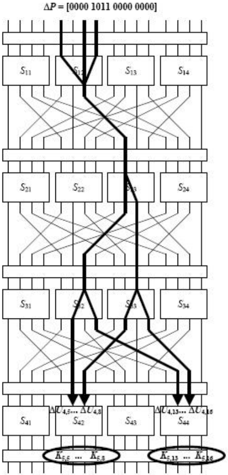
Constructing a Differential Characteristic

The diagram beside illustrates the tracing of the non-zero bits of the input difference through the SPN structure.

Input Difference: $\Delta P = 0B00$

Propagating Difference Pairs

- S_{12}: $1011 \rightarrow 0010$ $[Pr = \frac{8}{16}]$
- S_{23}: $0100 \rightarrow 0110$ $[Pr = \frac{6}{16}]$
- S_{32}: $0010 \rightarrow 0101$ $[Pr = \frac{6}{16}]$
- S_{33}: $0010 \rightarrow 0101$ $[Pr = \frac{6}{16}]$



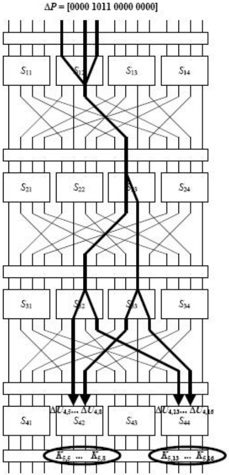
Constructing a Differential Characteristic

The diagram beside illustrates the tracing of the non-zero bits of the input difference through the SPN structure.

Input Difference: $\Delta P = 0B00$

Propagating Difference Pairs

- S_{12}: $1011 \rightarrow 0010$ $[Pr = \frac{8}{16}]$
- S_{23}: $0100 \rightarrow 0110$ $[Pr = \frac{6}{16}]$
- S_{32}: $0010 \rightarrow 0101$ $[Pr = \frac{6}{16}]$
- S_{33}: $0010 \rightarrow 0101$ $[Pr = \frac{6}{16}]$
- S_{ij}: $0000 \rightarrow 0000$ $[Pr = \frac{16}{16}]$



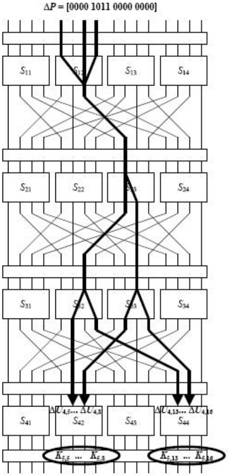
Constructing a Differential Characteristic

The diagram beside illustrates the tracing of the non-zero bits of the input difference through the SPN structure.

Input Difference: $\Delta P = 0B00$

Propagating Difference Pairs

- S_{12}: $1011 \rightarrow 0010$ $[Pr = \frac{8}{16}]$
- S_{23}: $0100 \rightarrow 0110$ $[Pr = \frac{6}{16}]$
- S_{32}: $0010 \rightarrow 0101$ $[Pr = \frac{6}{16}]$
- S_{33}: $0010 \rightarrow 0101$ $[Pr = \frac{6}{16}]$
- S_{ij}: $0000 \rightarrow 0000$ $[Pr = \frac{16}{16}]$



Constructing a Differential Characteristic

The diagram beside illustrates the tracing of the non-zero bits of the input difference through the SPN structure.

Input Difference: $\Delta P = 0B00$

Propagating Difference Pairs

- S_{12}: $1011 \rightarrow 0010$ $[Pr = \frac{8}{16}]$
- S_{23}: $0100 \rightarrow 0110$ $[Pr = \frac{6}{16}]$
- S_{32}: $0010 \rightarrow 0101$ $[Pr = \frac{6}{16}]$
- S_{33}: $0010 \rightarrow 0101$ $[Pr = \frac{6}{16}]$
- S_{ij}: $0000 \rightarrow 0000$ $[Pr = \frac{16}{16}]$

Differential Characteristic

$$0B00 \rightarrow 0040 \rightarrow 0220 \rightarrow 0606$$

Piling up Probabilities

Piling up Probabilities

Notation

- ΔU_i: Input difference to i-th round
- ΔV_i: Output difference from i-th round
- K_i: Round Key for i-th round
- $X_{i,j}$: j-th bit of X_i

Piling up Probabilities

Notation

- ΔU_i: Input difference to i-th round
- ΔV_i: Output difference from i-th round
- K_i: Round Key for i-th round
- $X_{i,j}$: j-th bit of X_i

Here, we have the following in the differential characteristic

- Round 1: $\Delta P = \Delta U_1 = 0B00 \rightarrow 0040 = \Delta V_1$ $[Pr = 8/16]$

Piling up Probabilities

Notation

- ΔU_i: Input difference to i-th round
- ΔV_i: Output difference from i-th round
- K_i: Round Key for i-th round
- $X_{i,j}$: j-th bit of X_i

Here, we have the following in the differential characteristic

- Round 1: $\Delta P = \Delta U_1 = 0B00 \rightarrow 0040 = \Delta V_1$ $[Pr = 8/16]$
- Round 2: $\Delta V_1 = \Delta U_2 = 0040 \rightarrow 0220 = \Delta V_2$ $[Pr = 6/16]$

Piling up Probabilities

Notation

- ΔU_i: Input difference to i-th round
- ΔV_i: Output difference from i-th round
- K_i: Round Key for i-th round
- $X_{i,j}$: j-th bit of X_i

Here, we have the following in the differential characteristic

- Round 1: $\Delta P = \Delta U_1 = 0B00 \rightarrow 0040 = \Delta V_1$ $[Pr = 8/16]$
- Round 2: $\Delta V_1 = \Delta U_2 = 0040 \rightarrow 0220 = \Delta V_2$ $[Pr = 6/16]$
- Round 3: $\Delta V_2 = \Delta U_3 = 0220 \rightarrow 0606 = \Delta V_3$ $[Pr = (6/16)^2]$

Piling up Probabilities

Notation

- ΔU_i: Input difference to i-th round
- ΔV_i: Output difference from i-th round
- K_i: Round Key for i-th round
- $X_{i,j}$: j-th bit of X_i

Here, we have the following in the differential characteristic

- Round 1: $\Delta P = \Delta U_1 = 0B00 \rightarrow 0040 = \Delta V_1$ $[Pr = 8/16]$
- Round 2: $\Delta V_1 = \Delta U_2 = 0040 \rightarrow 0220 = \Delta V_2$ $[Pr = 6/16]$
- Round 3: $\Delta V_2 = \Delta U_3 = 0220 \rightarrow 0606 = \Delta V_3$ $[Pr = (6/16)^2]$
- Round 4: $\Delta V_3 = \Delta U_4 = 0606$

Piling up Probabilities

Notation

- ΔU_i: Input difference to i-th round
- ΔV_i: Output difference from i-th round
- K_i: Round Key for i-th round
- $X_{i,j}$: j-th bit of X_i

Here, we have the following in the differential characteristic

- Round 1: $\Delta P = \Delta U_1 = 0B00 \rightarrow 0040 = \Delta V_1$ $[Pr = 8/16]$
- Round 2: $\Delta V_1 = \Delta U_2 = 0040 \rightarrow 0220 = \Delta V_2$ $[Pr = 6/16]$
- Round 3: $\Delta V_2 = \Delta U_3 = 0220 \rightarrow 0606 = \Delta V_3$ $[Pr = (6/16)^2]$
- Round 4: $\Delta V_3 = \Delta U_4 = 0606$

Piling up Probabilities

Notation

- ΔU_i: Input difference to i-th round
- ΔV_i: Output difference from i-th round
- K_i: Round Key for i-th round
- $X_{i,j}$: j-th bit of X_i

Here, we have the following in the differential characteristic

- Round 1: $\Delta P = \Delta U_1 = 0B00 \rightarrow 0040 = \Delta V_1$ $[Pr = 8/16]$
- Round 2: $\Delta V_1 = \Delta U_2 = 0040 \rightarrow 0220 = \Delta V_2$ $[Pr = 6/16]$
- Round 3: $\Delta V_2 = \Delta U_3 = 0220 \rightarrow 0606 = \Delta V_3$ $[Pr = (6/16)^2]$
- Round 4: $\Delta V_3 = \Delta U_4 = 0606$

Hence, assuming the S-Boxes to be independent over the rounds, we obtain

$$\Delta P = 0B00 \rightarrow 0606 = \Delta U_4$$

with probability $Pr(\Delta U_4 | \Delta P) = 8/16 \times 6/16 \times (6/16)^2 = 27/1024$.

Piling up Probabilities

Notation

- ΔU_i: Input difference to i-th round
- ΔV_i: Output difference from i-th round
- K_i: Round Key for i-th round
- $X_{i,j}$: j-th bit of X_i

Here, we have the following in the differential characteristic

- Round 1: $\Delta P = \Delta U_1 = 0B00 \rightarrow 0040 = \Delta V_1$ $[Pr = 8/16]$
- Round 2: $\Delta V_1 = \Delta U_2 = 0040 \rightarrow 0220 = \Delta V_2$ $[Pr = 6/16]$
- Round 3: $\Delta V_2 = \Delta U_3 = 0220 \rightarrow 0606 = \Delta V_3$ $[Pr = (6/16)^2]$
- Round 4: $\Delta V_3 = \Delta U_4 = 0606$

Hence, assuming the S-Boxes to be independent over the rounds, we obtain

$$\Delta P = 0B00 \rightarrow 0606 = \Delta U_4$$

with probability $Pr(\Delta U_4 | \Delta P) = 8/16 \times 6/16 \times (6/16)^2 = 27/1024$.

Note: The probability of this characteristic $Pr = 27/1024 \gg 1/2^{16}$

Attack Idea

- We have a differential characteristic that links plaintext difference ΔP to input difference ΔU_4 of the 4th round output of SPN and holds with high probability (27/1024)

Attack Idea

- We have a differential characteristic that links plaintext difference ΔP to input difference ΔU_4 of the 4th round output of SPN and holds with high probability (27/1024)
- We can partially undo the last round by guessing the last key K_5

Attack Idea

- We have a differential characteristic that links plaintext difference ΔP to input difference ΔU_4 of the 4th round output of SPN and holds with high probability (27/1024)
- We can partially undo the last round by guessing the last key K_5
- If we guess correctly, the differential characteristic will hold with high probability, close to 27/1024. If we guess wrong, it will hold with low probability

Attack Idea

- We have a differential characteristic that links plaintext difference ΔP to input difference ΔU_4 of the 4th round output of SPN and holds with high probability (27/1024)
- We can partially undo the last round by guessing the last key K_5
- If we guess correctly, the differential characteristic will hold with high probability, close to 27/1024. If we guess wrong, it will hold with low probability
- But in this case, we don't need to guess the entire key K_5 for the last round!

Attack Idea

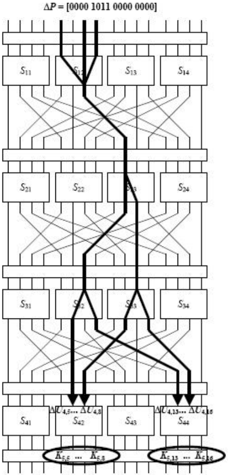
- We have a differential characteristic that links plaintext difference ΔP to input difference ΔU_4 of the 4th round output of SPN and holds with high probability (27/1024)
- We can partially undo the last round by guessing the last key K_5
- If we guess correctly, the differential characteristic will hold with high probability, close to 27/1024. If we guess wrong, it will hold with low probability
- But in this case, we don't need to guess the entire key K_5 for the last round!
- The differential output only involves 2 S-Boxes, S_{42} and S_{44}. Thus we only need to guess $2^8 = 256$ values

Attack Idea

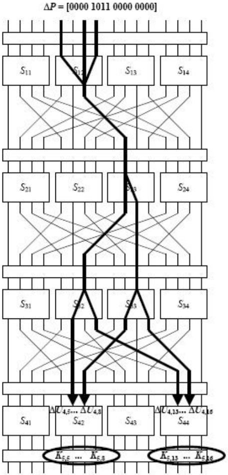
- We have a differential characteristic that links plaintext difference ΔP to input difference ΔU_4 of the 4th round output of SPN and holds with high probability (27/1024)
- We can partially undo the last round by guessing the last key K_5
- If we guess correctly, the differential characteristic will hold with high probability, close to 27/1024. If we guess wrong, it will hold with low probability
- But in this case, we don't need to guess the entire key K_5 for the last round!
- The differential output only involves 2 S-Boxes, S_{42} and S_{44}. Thus we only need to guess $2^8 = 256$ values
- For each value of the guessed *target partial subkey* we can undo the last round and determine the probability of the differential characteristic

Attack Idea

- We have a differential characteristic that links plaintext difference ΔP to input difference ΔU_4 of the 4th round output of SPN and holds with high probability $(27/1024)$
- We can partially undo the last round by guessing the last key K_5
- If we guess correctly, the differential characteristic will hold with high probability, close to $27/1024$. If we guess wrong, it will hold with low probability
- But in this case, we don't need to guess the entire key K_5 for the last round!
- The differential output only involves 2 S-Boxes, S_{42} and S_{44}. Thus we only need to guess $2^8 = 256$ values
- For each value of the guessed *target partial subkey* we can undo the last round and determine the probability of the differential characteristic
- Highest probability indicates likely correct guess

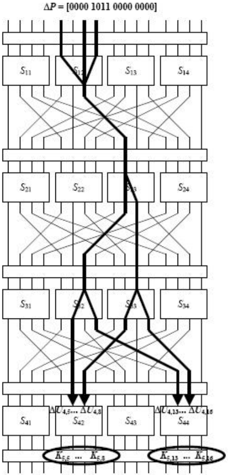


Differential Characteristic flow



Differential Characteristic flow

Input Difference: $\Delta P = 0B00$



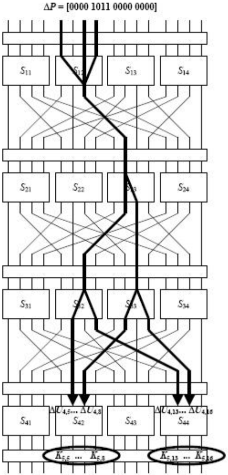
Differential Characteristic flow

Input Difference: $\Delta P = 0B00$

Characteristic

$$\Delta P = 0B00 \to 0606 = \Delta U_4$$

Probability
$Pr(\Delta U_4 = 0606 | \Delta P = 0B00) = 27/1024$



Differential Characteristic flow

Input Difference: $\Delta P = 0B00$

Characteristic

$$\Delta P = 0B00 \rightarrow 0606 = \Delta U_4$$

Probability
$Pr(\Delta U_4 = 0606 | \Delta P = 0B00) = 27/1024$

Partial Subkey guess only for

$$[K_{5,5} \ldots K_{5,8}, K_{5,13} \ldots K_{5,16}]$$

DEMO: Attack on 4-round SPN

It's time for a Break!

Do these work for DES?

Linear Cryptanalysis of 4 Round DES

The plan for the DES Attacks:

- Structurally and cryptographically, DES is similar to SPN.

The plan for the DES Attacks:

- Structurally and cryptographically, DES is similar to SPN.
- We will adapt the linear attack on SPN to a linear attack on a 4-round DES algorithm and mount this attack.

The plan for the DES Attacks:

- Structurally and cryptographically, DES is similar to SPN.
- We will adapt the linear attack on SPN to a linear attack on a 4-round DES algorithm and mount this attack.
- We will also show how to extend this attack to a full 16-round DES algorithm and show it is faster than an exhaustive search.

The first step, as before, is to come up with a linear approximation for the S-Boxes used by DES.

The first step, as before, is to come up with a linear approximation for the S-Boxes used by DES.

Note that the S-Boxes here are 6-bit to 4-bit maps. We can use the same technique as before.

The first step, as before, is to come up with a linear approximation for the S-Boxes used by DES.

Note that the S-Boxes here are 6-bit to 4-bit maps. We can use the same technique as before.

For a given S-Box $a \in \{1, 2, \ldots 8\}$, $1 \leq \alpha \leq 63$, $1 \leq \beta \leq 15$, we define $NS_a(\alpha, \beta)$ as the number of times (out of 64) for S-Box a that for all the input patterns masked by α the output pattern masked by β agrees with the value of S-Box a.

$$NS_a(\alpha, \beta) = |\{X|0 \leq X < 64, (\bigoplus_{i=0}^{5} X[i] \cdot \alpha[i]) = (\bigoplus_{j=0}^{3} S_a(X)[j] \cdot \beta[j])\}|$$

The first step, as before, is to come up with a linear approximation for the S-Boxes used by DES.

Note that the S-Boxes here are 6-bit to 4-bit maps. We can use the same technique as before.

For a given S-Box $a \in \{1, 2, \ldots 8\}$, $1 \leq \alpha \leq 63$, $1 \leq \beta \leq 15$, we define $NS_a(\alpha, \beta)$ as the number of times (out of 64) for S-Box a that for all the input patterns masked by α the output pattern masked by β agrees with the value of S-Box a.

$$NS_a(\alpha, \beta) = |\{X | 0 \leq X < 64, (\bigoplus_{i=0}^{5} X[i] \cdot \alpha[i]) = (\bigoplus_{j=0}^{3} S_a(X)[j] \cdot \beta[j])\}|$$

This looks more complicated, but is in fact the same expression as for SPN, except it takes into account all possible S-Boxes.

Using NS to find a linear approximation for the S-Boxes.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	4	-2	2	-2	2	-4	0	4	0	2	-2	2	-2	0	-4
3	0	-2	6	-2	-2	4	-4	0	0	-2	6	-2	-2	4	-4
4	2	-2	0	0	2	-2	0	0	2	2	4	-4	-2	-2	0
5	2	2	-4	0	10	-6	-4	0	2	-10	0	4	-2	2	4
6	-2	-4	-6	-2	-4	2	0	0	-2	0	-2	-6	-8	2	0
7	2	0	2	-2	8	6	0	-4	6	0	-6	-2	0	-6	-4
8	0	2	6	0	0	-2	-6	-2	2	4	-12	2	6	-4	4
9	-4	6	-2	0	-4	-6	-6	6	-2	0	-4	2	-6	-8	-4
10	4	0	0	-2	-6	2	2	2	2	-2	2	4	-4	-4	0
11	4	4	4	6	2	-2	-2	-2	-2	-2	2	0	-8	-4	0
12	2	0	-2	0	2	4	10	-2	4	-2	-8	-2	4	-6	-4
13	6	0	2	0	-2	4	-10	-2	0	-2	4	-2	8	-6	0
14	-2	-2	0	-2	4	0	2	-2	0	4	2	-4	6	-2	-4
15	-2	-2	8	6	4	0	2	2	4	8	-2	8	-6	2	0
16	2	-2	0	0	-2	-6	-8	0	-2	-2	-4	0	2	10	-20
17	2	-2	0	4	2	-2	-4	4	2	2	0	-8	-6	2	4
18	-2	0	-2	2	-4	-2	-8	4	6	4	6	-2	4	-6	0
19	-6	0	2	-2	4	2	0	4	-6	4	2	-6	4	-2	0
20	4	-4	0	0	0	0	0	-4	-4	4	4	0	4	-4	0
21	4	0	-4	-4	4	-8	-8	0	0	-4	4	8	4	0	4
22	0	6	6	2	-2	4	0	4	0	6	2	2	2	0	0
23	4	-6	-2	6	-2	-4	4	4	-4	-6	2	-2	2	0	4
24	6	0	2	4	-10	-4	2	2	0	-2	0	2	4	-2	-4
25	2	4	-6	0	-2	4	-2	6	8	6	4	10	0	2	-4

Figure: $NS_5(\alpha, \beta) - 32$

It turns out that the highest bias is for S-Box 5 with $\alpha = 16$ and $\beta = 15$

So, we noticed the S-Box approximation with highest bias for S-Box 5.

So, we noticed the S-Box approximation with highest bias for S-Box 5.

Letting X be the input bits to S-Box 5 and Y be the output bits,

$$X_2 = Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4$$

holds with bias of absolute value ~ 0.31

So, we noticed the S-Box approximation with highest bias for S-Box 5.

Letting X be the input bits to S-Box 5 and Y be the output bits,

$$X_2 = Y_1 \oplus Y_2 \oplus Y_3 \oplus Y_4$$

holds with bias of absolute value ~ 0.31

Tracing through the Feistel function to the right input bits R, round key bits K, and Feistel function output bits F we get

$$R_{17} \oplus F_{25} \oplus F_{14} \oplus F_8 \oplus F_3 = K_{26}$$

Let P_L and P_R be the left and right plaintext bits. Let C_L and C_R be the left and right ciphertext bits. Let K_i be the round key for round i and U_{4L} and U_{4R} the left and right halves of the input to round 4.

Let P_L and P_R be the left and right plaintext bits. Let C_L and C_R be the left and right ciphertext bits. Let K_i be the round key for round i and U_{4L} and U_{4R} the left and right halves of the input to round 4.

Using the S-Box 5 approximation and the Piling Up Principle we can combine the approximations to get

$$\begin{aligned} &P_L[25] \oplus P_L[14] \oplus P_L[8] \oplus P_L[3] \oplus \\ &\quad U_{4R}[25] \oplus U_{4R}[14] \oplus U_{4R}[8] \oplus U_{4R}[3] \oplus \\ &\quad P_R[17] \oplus U_{4L}[17] = K_1[26] \oplus K_3[26] \end{aligned}$$

Let P_L and P_R be the left and right plaintext bits. Let C_L and C_R be the left and right ciphertext bits. Let K_i be the round key for round i and U_{4L} and U_{4R} the left and right halves of the input to round 4.

Using the S-Box 5 approximation and the Piling Up Principle we can combine the approximations to get

$$\begin{aligned}&P_L[25] \oplus P_L[14] \oplus P_L[8] \oplus P_L[3] \oplus \\ &\quad U_{4R}[25] \oplus U_{4R}[14] \oplus U_{4R}[8] \oplus U_{4R}[3] \oplus \\ &\quad P_R[17] \oplus U_{4L}[17] = K_1[26] \oplus K_3[26]\end{aligned}$$

The bias magnitude for this expression is

$$(12/64)^2 + (1 - 12/64)^2 - 1/2 \approx 0.30$$

Let P_L and P_R be the left and right plaintext bits. Let C_L and C_R be the left and right ciphertext bits. Let K_i be the round key for round i and U_{4L} and U_{4R} the left and right halves of the input to round 4.

Using the S-Box 5 approximation and the Piling Up Principle we can combine the approximations to get

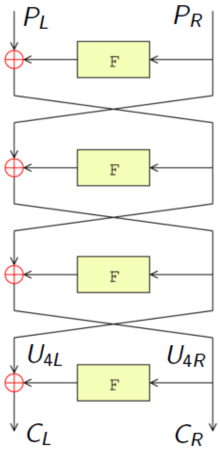
$$\begin{aligned} &P_L[25] \oplus P_L[14] \oplus P_L[8] \oplus P_L[3] \oplus \\ &\quad U_{4R}[25] \oplus U_{4R}[14] \oplus U_{4R}[8] \oplus U_{4R}[3] \oplus \\ &\quad P_R[17] \oplus U_{4L}[17] = K_1[26] \oplus K_3[26] \end{aligned}$$

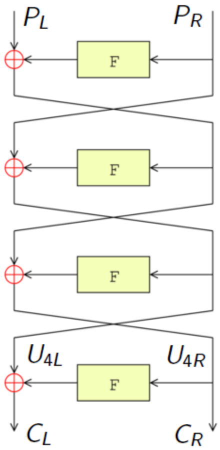
The bias magnitude for this expression is

$$(12/64)^2 + (1 - 12/64)^2 - 1/2 \approx 0.30$$

This is the best expression for the 3-round DES.

We perform the attack on the 4-round DES similarly to how we performed the attack on 4-round SPN, by "undoing" the last round.

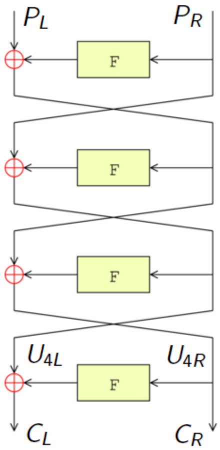




We perform the attack on the 4-round DES similarly to how we performed the attack on 4-round SPN, by "undoing" the last round.

Note that in the last round, U_{4R} bits are not changed, only U_{4L} bits are affected. Note also that there is only one U_{4L} bit the 3-round equation.

$$P_L[25] \oplus P_L[14] \oplus P_L[8] \oplus P_L[3] \oplus$$
$$U_{4R}[25] \oplus U_{4R}[14] \oplus U_{4R}[8] \oplus U_{4R}[3] \oplus$$
$$P_R[17] \oplus U_{4L}[17] = K_1[26] \oplus K_3[26]$$

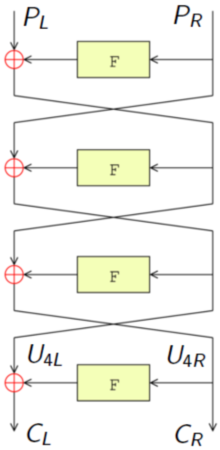


We perform the attack on the 4-round DES similarly to how we performed the attack on 4-round SPN, by "undoing" the last round.

Note that in the last round, U_{4R} bits are not changed, only U_{4L} bits are affected. Note also that there is only one U_{4L} bit the 3-round equation.

$$P_L[25] \oplus P_L[14] \oplus P_L[8] \oplus P_L[3] \oplus$$
$$U_{4R}[25] \oplus U_{4R}[14] \oplus U_{4R}[8] \oplus U_{4R}[3] \oplus$$
$$P_R[17] \oplus U_{4L}[17] = K_1[26] \oplus K_3[26]$$

This bit, $U_{4L}[17]$, is affected by the result of S-Box 1 in round 4. If we guess correctly for the 6 partial subkey bits $K_{4,1}$ to $K_{4,6}$, the 3-round equation will hold with high bias, otherwise it will likely hold with close to 0 bias.



We perform the attack on the 4-round DES similarly to how we performed the attack on 4-round SPN, by "undoing" the last round.

Note that in the last round, U_{4R} bits are not changed, only U_{4L} bits are affected. Note also that there is only one U_{4L} bit the 3-round equation.

$$P_L[25] \oplus P_L[14] \oplus P_L[8] \oplus P_L[3] \oplus$$
$$U_{4R}[25] \oplus U_{4R}[14] \oplus U_{4R}[8] \oplus U_{4R}[3] \oplus$$
$$P_R[17] \oplus U_{4L}[17] = K_1[26] \oplus K_3[26]$$

This bit, $U_{4L}[17]$, is affected by the result of S-Box 1 in round 4. If we guess correctly for the 6 partial subkey bits $K_{4,1}$ to $K_{4,6}$, the 3-round equation will hold with high bias, otherwise it will likely hold with close to 0 bias.

Thus, this attack will give us 6 of the 56 key bits. If it takes a year to find a key through exhaustive search of 2^{56} bits, it only takes 5 days to exhaustively search 2^{50} bits.

How many pairs do we need? (Probabilistic justification)

How many pairs do we need? (Probabilistic justification)

Matsui Lemma 5: Let N be the number of given random plaintexts and ϵ, the bias, be sufficiently small. Let $q^{(i)}$ be the probability that the following equation holds for a subkey candidate $K_n^{(i)}$ and a random variable X:

$$F_n(X, K_n)[l_1, l_2, \ldots, l_d] = F_n(X, K_n^{(i)})[l_1, l_2, \ldots, l_d].$$

How many pairs do we need? (Probabilistic justification)

Matsui Lemma 5: Let N be the number of given random plaintexts and ϵ, the bias, be sufficiently small. Let $q^{(i)}$ be the probability that the following equation holds for a subkey candidate $K_n^{(i)}$ and a random variable X:

$$F_n(X, K_n)[l_1, l_2, \ldots, l_d] = F_n(X, K_n^{(i)})[l_1, l_2, \ldots, l_d].$$

Then if $q^{(i)}$'s are independent, the success rate of the attack is

$$\int_{x=-2\sqrt{N}\epsilon}^{\infty} \left(\prod_{K_n^{(i)} \neq K_n} \int_{-x-4\sqrt{N}\epsilon q^{(i)}}^{x+4\sqrt{N}\epsilon(1-q^{(i)})} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy \right) \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$$

where the product is taken over all subkey candidates except K_n.

How many pairs do we need? (Probabilistic justification)

Matsui Lemma 5: Let N be the number of given random plaintexts and ϵ, the bias, be sufficiently small. Let $q^{(i)}$ be the probability that the following equation holds for a subkey candidate $K_n^{(i)}$ and a random variable X:

$$F_n(X, K_n)[l_1, l_2, \ldots, l_d] = F_n(X, K_n^{(i)})[l_1, l_2, \ldots, l_d].$$

Then if $q^{(i)}$'s are independent, the success rate of the attack is

$$\int_{x=-2\sqrt{N}\epsilon}^{\infty} \left(\prod_{K_n^{(i)} \neq K_n} \int_{-x-4\sqrt{N}\epsilon q^{(i)}}^{x+4\sqrt{N}\epsilon(1-q^{(i)})} \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy \right) \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$$

where the product is taken over all subkey candidates except K_n.

Let $d = 1$ and $l_1 = 17$ in the above expression - this corresponds to our attack (distribution over guessing for one bit, $U_{4R}[17]$ in the equation). The following table summarizes the success rates:

N	$2\epsilon^{-2}$	$4\epsilon^{-2}$	$8\epsilon^{-2}$	$16\epsilon^{-2}$
Success Rate	0.486	0.785	0.967	0.999

How many pairs do we need? (Continued)

Using the calculation from the previous slide, we see that for a 4-round DES, our equation holds with bias of about 1 in 4.

How many pairs do we need? (Continued)

Using the calculation from the previous slide, we see that for a 4-round DES, our equation holds with bias of about 1 in 4. Thus we will need about $16 \times 4^2 = 256$ pairs for the attack to have near perfect chance of success.

How many pairs do we need? (Continued)

Using the calculation from the previous slide, we see that for a 4-round DES, our equation holds with bias of about 1 in 4. Thus we will need about $16 \times 4^2 = 256$ pairs for the attack to have near perfect chance of success.

For a full 16-round DES, the best approximation has bias $\sim 1.19 \times 10^{-22}$, thus we would need on the order of 2^{47} plaintext-ciphertext pairs.

DEMO: Attack on 4-round DES

Differential Cryptanalysis of 4 Round DES

Differential Attack Idea

Similar to the differential attack on SPN, here we care about the differential characteristics of the cipher.

Differential Attack Idea

Similar to the differential attack on SPN, here we care about the differential characteristics of the cipher. So, the basic idea is as follows

- Construct the difference tables for the S-Boxes

Differential Attack Idea

Similar to the differential attack on SPN, here we care about the differential characteristics of the cipher. So, the basic idea is as follows

- Construct the difference tables for the S-Boxes
- Find the most probable difference pairs

Differential Attack Idea

Similar to the differential attack on SPN, here we care about the differential characteristics of the cipher. So, the basic idea is as follows

- Construct the difference tables for the S-Boxes
- Find the most probable difference pairs
- Combine the pairs through all the rounds to get a highly probable differential characteristic until the penultimate round

Differential Attack Idea

Similar to the differential attack on SPN, here we care about the differential characteristics of the cipher. So, the basic idea is as follows

- Construct the difference tables for the S-Boxes
- Find the most probable difference pairs
- Combine the pairs through all the rounds to get a highly probable differential characteristic until the penultimate round
- Exploit this differential characteristic by choosing a number of pairs of plaintexts with the desired difference and undo the last round by guessing the keybits

Differential Attack Idea

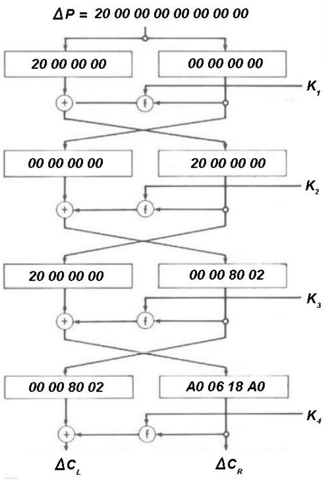
Similar to the differential attack on SPN, here we care about the differential characteristics of the cipher. So, the basic idea is as follows

- Construct the difference tables for the S-Boxes
- Find the most probable difference pairs
- Combine the pairs through all the rounds to get a highly probable differential characteristic until the penultimate round
- Exploit this differential characteristic by choosing a number of pairs of plaintexts with the desired difference and undo the last round by guessing the keybits
- If the initial choice of the input difference was smart, we will only need to guess a few keybits

Differential Attack Idea

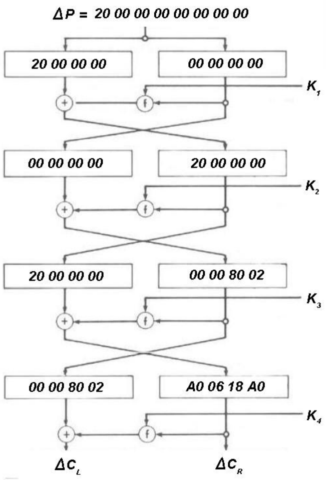
Similar to the differential attack on SPN, here we care about the differential characteristics of the cipher. So, the basic idea is as follows

- Construct the difference tables for the S-Boxes
- Find the most probable difference pairs
- Combine the pairs through all the rounds to get a highly probable differential characteristic until the penultimate round
- Exploit this differential characteristic by choosing a number of pairs of plaintexts with the desired difference and undo the last round by guessing the keybits
- If the initial choice of the input difference was smart, we will only need to guess a few keybits
- The guess approving the differential characteristic with highest probability is correct



Differential Characteristic of DES

Input Difference: $\Delta P = 20\ 00\ 00\ 00\ 00\ 00\ 00\ 00$

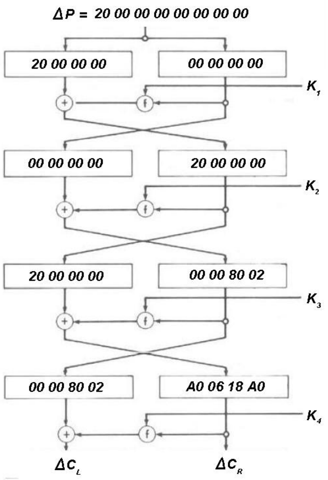


Differential Characteristic of DES

Input Difference: $\Delta P = 20\ 00\ 00\ 00\ 00\ 00\ 00\ 00$

Propagating Difference Pairs

- Round 1: $S_i\ :\ 00 \rightarrow 00$

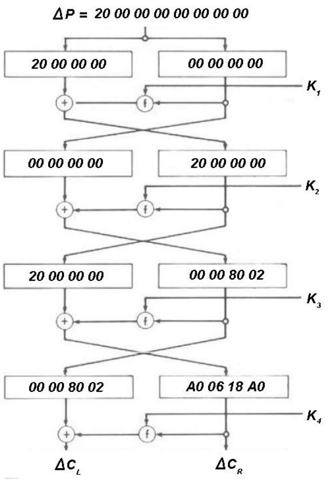


Differential Characteristic of DES

Input Difference: $\Delta P = 20\ 00\ 00\ 00\ 00\ 00\ 00\ 00$

Propagating Difference Pairs

- Round 1: S_i : $00 \rightarrow 00$
- Round 2: S_1 : $04 \rightarrow 05$

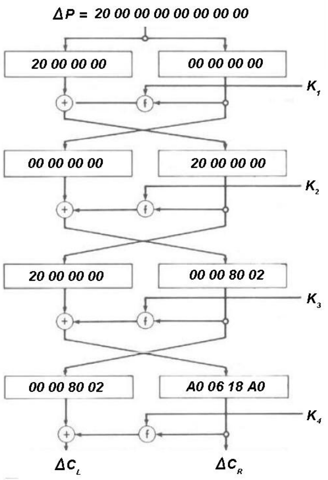


Differential Characteristic of DES

Input Difference: $\Delta P = 20\ 00\ 00\ 00\ 00\ 00\ 00\ 00$

Propagating Difference Pairs

- Round 1: $S_i\ :\ 00 \longrightarrow 00$
- Round 2: $S_1\ :\ 04 \longrightarrow 05$
- Round 3: $S_4\ :\ 01 \longrightarrow 05$
  $S_5\ :\ 10 \longrightarrow 07$
  $S_8\ :\ 04 \longrightarrow 07$

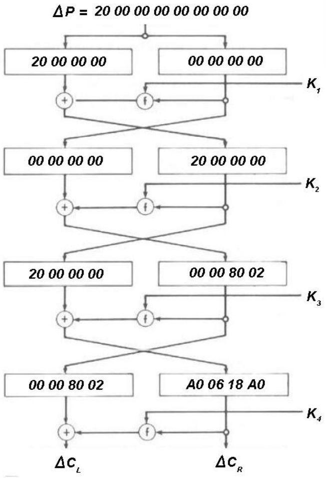


Differential Characteristic of DES

Input Difference: $\Delta P = 20\ 00\ 00\ 00\ 00\ 00\ 00\ 00$

Propagating Difference Pairs

- Round 1: S_i : $00 \longrightarrow 00$
- Round 2: S_1 : $04 \longrightarrow 05$
- Round 3: S_4 : $01 \longrightarrow 05$
  S_5 : $10 \longrightarrow 07$
  S_8 : $04 \longrightarrow 07$



Differential Characteristic of DES

Input Difference: $\Delta P = 20\ 00\ 00\ 00\ 00\ 00\ 00\ 00$

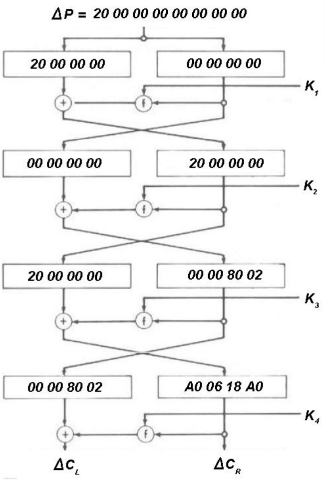
Propagating Difference Pairs

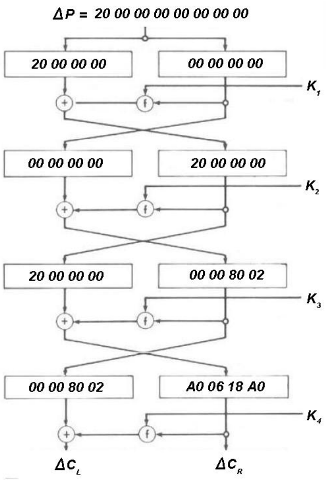
- Round 1: $S_i\ :\ 00 \longrightarrow 00$
- Round 2: $S_1\ :\ 04 \longrightarrow 05$
- Round 3: $S_4\ :\ 01 \longrightarrow 05$
  $S_5\ :\ 10 \longrightarrow 07$
  $S_8\ :\ 04 \longrightarrow 07$

Differential Characteristic

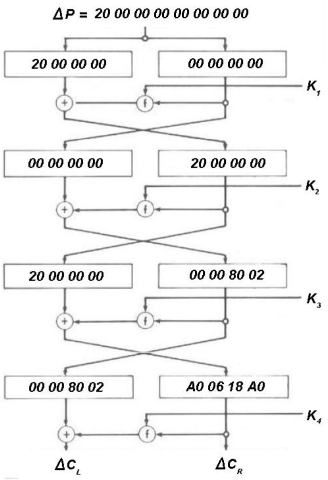
$20\,00\,00\,00\,00\,00\,00\,00 \longrightarrow A0\,06\,18\,A0\,00\,00\,80\,02$

Differential Attack: 4-round DES





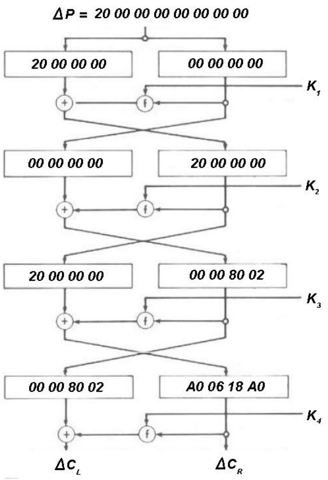
Note: 00 00 80 02 implies difference of 2 bits which are both related to S-Box S_1 in round 4.



Note: 00 00 80 02 implies difference of 2 bits which are both related to S-Box S_1 in round 4.

Hence, in round 4

$$D' = \Delta C_L \oplus 00\,00\,80\,02$$



Note: 00 00 80 02 implies difference of 2 bits which are both related to S-Box S_1 in round 4.

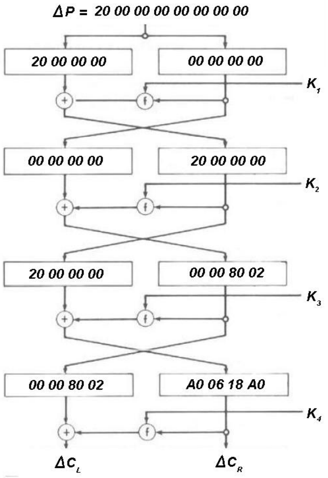
Hence, in round 4

$$D' = \Delta C_L \oplus 00\,00\,80\,02$$

and hence for $S_2, ..., S_8$, outputs are

$$\Delta Y = P^{-1}(\Delta C_L) = S_{box}(E(C_R) \oplus K_4)$$

So, just guess the *target partial subkey* for $S_2, ...S_8$ individually,



Note: 00 00 80 02 implies difference of 2 bits which are both related to S-Box S_1 in round 4.

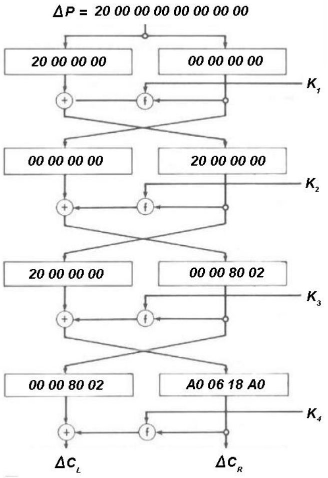
Hence, in round 4

$$D' = \Delta C_L \oplus 00\ 00\ 80\ 02$$

and hence for $S_2, ..., S_8$, outputs are

$$\Delta Y = P^{-1}(\Delta C_L) = S_{box}(E(C_R) \oplus K_4)$$

So, just guess the *target partial subkey* for $S_2, ...S_8$ individually, and compute their predicted ΔY with a number of plaintext pairs.



Note: 00 00 80 02 implies difference of 2 bits which are both related to S-Box S_1 in round 4.

Hence, in round 4

$$D' = \Delta C_L \oplus 00\ 00\ 80\ 02$$

and hence for $S_2, ..., S_8$, outputs are

$$\Delta Y = P^{-1}(\Delta C_L) = S_{box}(E(C_R) \oplus K_4)$$

So, just guess the *target partial subkey* for $S_2, ...S_8$ individually, and compute their predicted ΔY with a number of plaintext pairs.

This will "undo" round 4 and the maximum probability of satisfying the 3 round differential characteristic will indicate a correct guess.

Probabilistic Analysis

Constructing the Difference Distribution tables for the S-Boxes and analyzing the probability distribution for differential propagation through SBoxes:

Probabilistic Analysis

Constructing the Difference Distribution tables for the S-Boxes and analyzing the probability distribution for differential propagation through SBoxes:

- Round 1: $S_i : 00 \rightarrow 00$ with $Pr = 1$

Probabilistic Analysis

Constructing the Difference Distribution tables for the S-Boxes and analyzing the probability distribution for differential propagation through SBoxes:

- Round 1: S_i : $00 \rightarrow 00$ with $Pr = 1$
- Round 2: S_1 : $04 \rightarrow 05$ with $Pr = 10/64$

Probabilistic Analysis

Constructing the Difference Distribution tables for the S-Boxes and analyzing the probability distribution for differential propagation through SBoxes:

- Round 1: $S_i \, : \, 00 \longrightarrow 00$ with $Pr = 1$
- Round 2: $S_1 \, : \, 04 \longrightarrow 05$ with $Pr = 10/64$
- Round 3: $S_4 \, : \, 01 \longrightarrow 05$ with $Pr = 16/64$
  $S_5 \, : \, 10 \longrightarrow 07$ with $Pr = 12/64$
  $S_8 \, : \, 04 \longrightarrow 07$ with $Pr = 12/64$

Probabilistic Analysis

Constructing the Difference Distribution tables for the S-Boxes and analyzing the probability distribution for differential propagation through SBoxes:

- Round 1: S_i : $00 \longrightarrow 00$ with $Pr = 1$
- Round 2: S_1 : $04 \longrightarrow 05$ with $Pr = 10/64$
- Round 3: S_4 : $01 \longrightarrow 05$ with $Pr = 16/64$
  S_5 : $10 \longrightarrow 07$ with $Pr = 12/64$
  S_8 : $04 \longrightarrow 07$ with $Pr = 12/64$

Probabilistic Analysis

Constructing the Difference Distribution tables for the S-Boxes and analyzing the probability distribution for differential propagation through SBoxes:

- Round 1: S_i : $00 \rightarrow 00$ with $Pr = 1$
- Round 2: S_1 : $04 \rightarrow 05$ with $Pr = 10/64$
- Round 3: S_4 : $01 \rightarrow 05$ with $Pr = 16/64$
  S_5 : $10 \rightarrow 07$ with $Pr = 12/64$
  S_8 : $04 \rightarrow 07$ with $Pr = 12/64$

Differential Characteristic for ΔC_L: $00\,00\,00\,00 \rightarrow 00\,00\,80\,02$ holds with probability

$$p_D = 10/64 \approx 0.16 \gg 1/2^{32}$$

Probabilistic Analysis

Constructing the Difference Distribution tables for the S-Boxes and analyzing the probability distribution for differential propagation through SBoxes:

- Round 1: $S_i : 00 \rightarrow 00$ with $Pr = 1$
- Round 2: $S_1 : 04 \rightarrow 05$ with $Pr = 10/64$
- Round 3: $S_4 : 01 \rightarrow 05$ with $Pr = 16/64$
  $S_5 : 10 \rightarrow 07$ with $Pr = 12/64$
  $S_8 : 04 \rightarrow 07$ with $Pr = 12/64$

Differential Characteristic for ΔC_L: 00 00 00 00 $\rightarrow$ 00 00 80 02 holds with probability

$$p_D \; = \; 10/64 \; \approx \; 0.16 \; \gg 1/2^{32}$$

- How many pairs do we need?

Probabilistic Analysis

Constructing the Difference Distribution tables for the S-Boxes and analyzing the probability distribution for differential propagation through SBoxes:

- Round 1: S_i : $00 \rightarrow 00$ with $Pr = 1$
- Round 2: S_1 : $04 \rightarrow 05$ with $Pr = 10/64$
- Round 3: S_4 : $01 \rightarrow 05$ with $Pr = 16/64$
  S_5 : $10 \rightarrow 07$ with $Pr = 12/64$
  S_8 : $04 \rightarrow 07$ with $Pr = 12/64$

Differential Characteristic for ΔC_L: $00\,00\,00\,00 \rightarrow 00\,00\,80\,02$ holds with probability

$$p_D = 10/64 \approx 0.16 \gg 1/2^{32}$$

- How many pairs do we need?
- Approximately $2^4 = 16$ for 100% success

Probabilistic Analysis

Constructing the Difference Distribution tables for the S-Boxes and analyzing the probability distribution for differential propagation through SBoxes:

- Round 1: S_i : $00 \rightarrow 00$ with $Pr = 1$
- Round 2: S_1 : $04 \rightarrow 05$ with $Pr = 10/64$
- Round 3: S_4 : $01 \rightarrow 05$ with $Pr = 16/64$
  S_5 : $10 \rightarrow 07$ with $Pr = 12/64$
  S_8 : $04 \rightarrow 07$ with $Pr = 12/64$

Differential Characteristic for ΔC_L: $00\,00\,00\,00 \rightarrow 00\,00\,80\,02$ holds with probability

$$p_D = 10/64 \approx 0.16 \gg 1/2^{32}$$

- How many pairs do we need?
- Approximately $2^4 = 16$ for 100% success

Probabilistic Analysis

Constructing the Difference Distribution tables for the S-Boxes and analyzing the probability distribution for differential propagation through SBoxes:

- Round 1: $S_i \,:\, 00 \rightarrow 00$ with $Pr = 1$
- Round 2: $S_1 \,:\, 04 \rightarrow 05$ with $Pr = 10/64$
- Round 3: $S_4 \,:\, 01 \rightarrow 05$ with $Pr = 16/64$
  $S_5 \,:\, 10 \rightarrow 07$ with $Pr = 12/64$
  $S_8 \,:\, 04 \rightarrow 07$ with $Pr = 12/64$

Differential Characteristic for ΔC_L: 00 00 00 00 $\rightarrow$ 00 00 80 02 holds with probability

$$p_D \;=\; 10/64 \;\approx\; 0.16 \;\gg 1/2^{32}$$

- How many pairs do we need?
- Approximately $2^4 = 16$ for 100% success

For 16 round DES, 2^{47} pairs are needed. [2^{36} "good pairs"]

DEMO: Attack on 4-round DES

Concluding Remarks

Independence of S-Boxes

Independence of S-Boxes

Note that throughout this discussion we have assumed that the S-Boxes were independent. This allowed us to use the Piling Up Principle to combine linear approximations of S-Boxes across rounds.

Independence of S-Boxes

Note that throughout this discussion we have assumed that the S-Boxes were independent. This allowed us to use the Piling Up Principle to combine linear approximations of S-Boxes across rounds.

This assumption worked well for us in practice, but is not necessarily true. We could have proceeded differently without this assumption - [see John Manferdelli's Boolean Functions slides]

Linear Resistance

Let $F : GF(2)^p \rightarrow GF(2)^q$.

Linear Resistance

Let $F : GF(2)^p \rightarrow GF(2)^q$.

Defn: A function F is bent if its Walsh transform $\mathcal{W}\{F\}$ is $\pm 2^{p/2}$.

Linear Resistance

Let $F : GF(2)^p \rightarrow GF(2)^q$.

Defn: A function F is bent if its Walsh transform $\mathcal{W}\{F\}$ is $\pm 2^{p/2}$.

Thm: A function F is perfect nonlinear (ie bias is 0) if and only if it is *bent*.

Linear Resistance

Let $F : GF(2)^p \rightarrow GF(2)^q$.

Defn: A function F is bent if its Walsh transform $\mathcal{W}\{F\}$ is $\pm 2^{p/2}$.

Thm: A function F is perfect nonlinear (ie bias is 0) if and only if it is *bent*.

Thm: Bent functions exist only for $p \geq 2q$ and p even.

Linear Resistance

Let $F : GF(2)^p \rightarrow GF(2)^q$.

Defn: A function F is bent if its Walsh transform $\mathcal{W}\{F\}$ is $\pm 2^{p/2}$.

Thm: A function F is perfect nonlinear (ie bias is 0) if and only if it is *bent*.

Thm: Bent functions exist only for $p \geq 2q$ and p even.

To sumarize: When $p \geq 2q$ and p even, differential-resistant is equivalent to linear resistant, and to vectorial Bentness.

'Moral of the story'

If you are mounting the attacks, remember the following:

If you are mounting the attacks, remember the following:

- Analyze the main non-linear component, S-Boxes for example

If you are mounting the attacks, remember the following:

- Analyze the main non-linear component, S-Boxes for example
- For linear attack, try to find the linear/affine approximation

If you are mounting the attacks, remember the following:

- Analyze the main non-linear component, S-Boxes for example
- For linear attack, try to find the linear/affine approximation
- For differential attack, find the best characteristic

If you are mounting the attacks, remember the following:

- Analyze the main non-linear component, S-Boxes for example
- For linear attack, try to find the linear/affine approximation
- For differential attack, find the best characteristic
- Combine through levels to get the total approximation/characteristic

If you are mounting the attacks, remember the following:

- Analyze the main non-linear component, S-Boxes for example
- For linear attack, try to find the linear/affine approximation
- For differential attack, find the best characteristic
- Combine through levels to get the total approximation/characteristic
- Analyze piling up probabilities to estimate number of required pairs

If you are mounting the attacks, remember the following:

- Analyze the main non-linear component, S-Boxes for example
- For linear attack, try to find the linear/affine approximation
- For differential attack, find the best characteristic
- Combine through levels to get the total approximation/characteristic
- Analyze piling up probabilities to estimate number of required pairs
- Verify if this is better than any other existing attack

If you are designing a block cipher, note:

If you are designing a block cipher, note:

The problems of DES originated in two aspects

- 6 bit S-Boxes are too easy to identify with linear or affine functions with high bias

If you are designing a block cipher, note:

The problems of DES originated in two aspects

- 6 bit S-Boxes are too easy to identify with linear or affine functions with high bias
- Diffusion between rounds is just not fast enough

If you are designing a block cipher, note:

The problems of DES originated in two aspects

- 6 bit S-Boxes are too easy to identify with linear or affine functions with high bias
- Diffusion between rounds is just not fast enough

If you are designing a block cipher, note:

The problems of DES originated in two aspects

- 6 bit S-Boxes are too easy to identify with linear or affine functions with high bias
- Diffusion between rounds is just not fast enough

Prescription

- Create S-Boxes as bent functions [inefficient as $p \geq 2q$]

If you are designing a block cipher, note:

The problems of DES originated in two aspects

- 6 bit S-Boxes are too easy to identify with linear or affine functions with high bias
- Diffusion between rounds is just not fast enough

Prescription

- Create S-Boxes as bent functions [inefficient as $p \geq 2q$]
- Make the diffusion step faster to create an avalanche

If you are designing a block cipher, note:

The problems of DES originated in two aspects

- 6 bit S-Boxes are too easy to identify with linear or affine functions with high bias
- Diffusion between rounds is just not fast enough

Prescription

- Create S-Boxes as bent functions [inefficient as $p \geq 2q$]
- Make the diffusion step faster to create an avalanche

If you are designing a block cipher, note:

The problems of DES originated in two aspects

- 6 bit S-Boxes are too easy to identify with linear or affine functions with high bias
- Diffusion between rounds is just not fast enough

Prescription

- Create S-Boxes as bent functions [inefficient as $p \geq 2q$]
- Make the diffusion step faster to create an avalanche

AES, which replaced DES, provides an 8-Bit S-Box in the SubBytes step and higher diffusion through the ShiftRows and MixColumns steps.

Just for Fun!

What if you play around with the S-Box ordering?

What if you play around with the S-Box ordering?

- Matsui prescribed a rearrangement of S-Boxes specifically against linear attack.

What if you play around with the S-Box ordering?

- Matsui prescribed a rearrangement of S-Boxes specifically against linear attack.
- Matsui prescribed another rearrangement of the S-Boxes, which is good enough to prevent both linear and differential attacks.

What if you play around with the S-Box ordering?

- Matsui prescribed a rearrangement of S-Boxes specifically against linear attack.
- Matsui prescribed another rearrangement of the S-Boxes, which is good enough to prevent both linear and differential attacks.
- Biham and Shamir illustrated a rearrangement where differential attack becomes way more efficient.

IBM Knew in 1974

Date: Wed, 19 Feb 92 09:43:31 EST
From: "Don Coppersmith" <copper@watson.ibm.com>

Adi,

We have kept quiet about the following for 18 years, and decided it's time to break the silence.

We (IBM crypto group) knew about differential cryptanalysis in 1974. This is why DES stood up to this line of attack; we designed the S-boxes and the permutation in such a way as to defeat it.

IBM Knew in 1974

Date: Wed, 19 Feb 92 09:43:31 EST
From: "Don Coppersmith" <copper@watson.ibm.com>

Adi,

We have kept quiet about the following for 18 years, and decided it's time to break the silence.

We (IBM crypto group) knew about differential cryptanalysis in 1974. This is why DES stood up to this line of attack; we designed the S-boxes and the permutation in such a way as to defeat it.

S-Box Design criteria

- 1 bit input difference produces 2 bits output difference

IBM Knew in 1974

Date: Wed, 19 Feb 92 09:43:31 EST
From: "Don Coppersmith" <copper@watson.ibm.com>

Adi,

We have kept quiet about the following for 18 years, and decided it's time to break the silence.

We (IBM crypto group) knew about differential cryptanalysis in 1974. This is why DES stood up to this line of attack; we designed the S-boxes and the permutation in such a way as to defeat it.

S-Box Design criteria

- 1 bit input difference produces 2 bits output difference
- Minimize the difference between the numbers of 1's and 0's when any input bit is held constant

IBM Knew in 1974

Date: Wed, 19 Feb 92 09:43:31 EST
From: "Don Coppersmith" <copper@watson.ibm.com>

Adi,

We have kept quiet about the following for 18 years, and decided it's time to break the silence.

We (IBM crypto group) knew about differential cryptanalysis in 1974. This is why DES stood up to this line of attack; we designed the S-boxes and the permutation in such a way as to defeat it.

S-Box Design criteria

- 1 bit input difference produces 2 bits output difference
- Minimize the difference between the numbers of 1's and 0's when any input bit is held constant
- $S(X) \neq S(X \oplus 11ab00)$

IBM Knew in 1974

Date: Wed, 19 Feb 92 09:43:31 EST
From: "Don Coppersmith" <copper@watson.ibm.com>

Adi,

We have kept quiet about the following for 18 years, and decided it's time to break the silence.

We (IBM crypto group) knew about differential cryptanalysis in 1974. This is why DES stood up to this line of attack; we designed the S-boxes and the permutation in such a way as to defeat it.

S-Box Design criteria

- 1 bit input difference produces 2 bits output difference
- Minimize the difference between the numbers of 1's and 0's when any input bit is held constant
- $S(X) \neq S(X \oplus 11ab00)$

IBM Knew in 1974

Date: Wed, 19 Feb 92 09:43:31 EST
From: "Don Coppersmith" <copper@watson.ibm.com>

Adi,

We have kept quiet about the following for 18 years, and decided it's time to break the silence.

We (IBM crypto group) knew about differential cryptanalysis in 1974. This is why DES stood up to this line of attack; we designed the S-boxes and the permutation in such a way as to defeat it.

S-Box Design criteria

- 1 bit input difference produces 2 bits output difference
- Minimize the difference between the numbers of 1's and 0's when any input bit is held constant
- $S(X) \neq S(X \oplus 11ab00)$

Other good stuff

- 2 input difference bits mapped to 3 by $E(R)$

IBM Knew in 1974

Date: Wed, 19 Feb 92 09:43:31 EST
From: "Don Coppersmith" <copper@watson.ibm.com>

Adi,

We have kept quiet about the following for 18 years, and decided it's time to break the silence.

We (IBM crypto group) knew about differential cryptanalysis in 1974. This is why DES stood up to this line of attack; we designed the S-boxes and the permutation in such a way as to defeat it.

S-Box Design criteria

- 1 bit input difference produces 2 bits output difference
- Minimize the difference between the numbers of 1's and 0's when any input bit is held constant
- $S(X) \neq S(X \oplus 11ab00)$

Other good stuff

- 2 input difference bits mapped to 3 by $E(R)$
- Each output difference bit permuted to 1 each to S-boxes of next round

Another Email with Don Coppersmith

Date: Mon, 14 Oct 91 09:12:35 EDT
From: "Don Coppersmith" <copper@watson.ibm.com>
To: biham@cs.technion.ac.il
Subject: press reports

Eli,
The press reports (NY Times, Oct 4 and again Oct 13) both imply a very small amount of chosen plaintext
...
Yet my understanding of DES and differential cryptanalysis is that massive amounts of chosen plaintext is required, more like 10^15, give or take an order of magnitude.

I understand you don't want to publicize the actual parameters.
But please confirm my suspicion that the reporters have misunderstood on this point; that it's closer to 10^15 than to 10.

Thanks. Don

Another Email with Don Coppersmith

```
Date: Mon, 14 Oct 91 09:12:35 EDT
From: "Don Coppersmith" <copper@watson.ibm.com>
To: biham@cs.technion.ac.il
Subject: press reports

Eli,
The press reports (NY Times, Oct 4 and again Oct 13) both imply a very
small amount of chosen plaintext
...
Yet my understanding of DES and differential cryptanalysis is that
massive amounts of chosen plaintext is required, more like 10^15,
give or take an order of magnitude.

I understand you don't want to publicize the actual parameters.
But please confirm my suspicion that the reporters have misunderstood
on this point; that it's closer to 10^15 than to 10.

Thanks.   Don
```


Average of 10 and 10^{45} is $5.10^{44} + 5 > 10^{44} \approx 2^{47}$

Another Email with Don Coppersmith

```
Date: Mon, 14 Oct 91 09:12:35 EDT
From: "Don Coppersmith" <copper@watson.ibm.com>
To: biham@cs.technion.ac.il
Subject: press reports

Eli,
The press reports (NY Times, Oct 4 and again Oct 13) both imply a very
small amount of chosen plaintext
...
Yet my understanding of DES and differential cryptanalysis is that
massive amounts of chosen plaintext is required, more like 10^15,
give or take an order of magnitude.

I understand you don't want to publicize the actual parameters.
But please confirm my suspicion that the reporters have misunderstood
on this point; that it's closer to 10^15 than to 10.

Thanks.   Don
```

© Eli Biham - March 16, 2006 37 FSE 2006, Graz, Austria

Average of 10 and 10^{45} is $5.10^{44} + 5 > 10^{44} \approx 2^{47}$

Don was wrong. 2^{47} is closer to 10 than to 10^{15}.

- Eli Biham

References:

M. Matsui, "Linear Cryptanalysis Method for DES Cipher", *Advances in Cryptology - EUROCRYPT '93 (Lecture Notes in Computer Science no. 765),* Springer-Verlag, pp. 38

E Biham And A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems", *Journal of Cryptology*, vol 4, no 1, pp. 3-72, 1991.

H. Heys, "A Tutorial on Linear and Differential Cryptanalysis"

F. Chabaud And S. Vaudenay, "Links Between Differential and Linear Cryptanalysis", *Advances in Cryptology - EUROCRYPT '94 (Lexture Notes in Computer Science no. 950)*, Springer-Verlag, pp. 356-365, 1995.

J. Manferdelli, UW Cryptography Lecture Notes

The End.

Questions?