Study and Analysis of BLAKE2 Hash Function

MS Project (ECS 501)

(Under Dr. Shashank Singh)

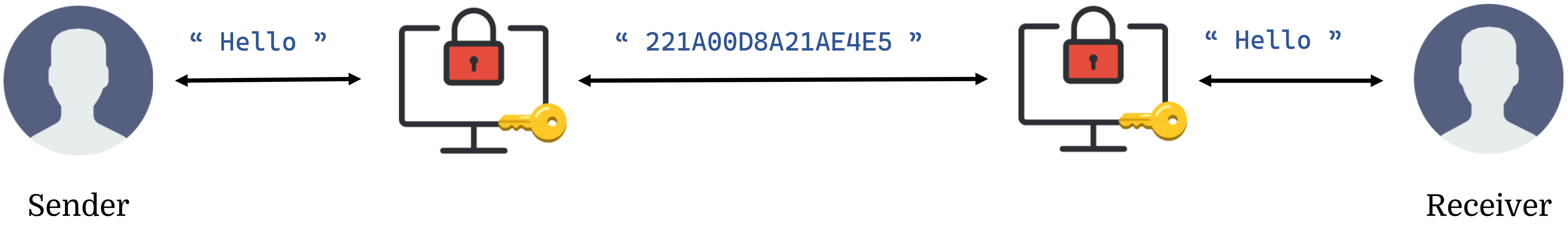
BLAKE2s-256, DES and Differential Cryptanalysis

Ajay Choudhury (18018)

EECS Department

Encryption and Block Ciphers

- Encryption is a way of scrambling data using a cryptographic key, so that only authorized machines or users can interpret the information.
- The scrambled output of the encryption machine is known as ciphertext, it is not in human-interpretable form.
- Block cipher is an encryption method that encrypts data in blocks of fixed size unlike stream ciphers that encrypt data one byte at a time.
- The most common use of encryption can be seen in messaging apps (e.g., WhatsApp, Zoom, etc.), often they use end-to-end encryption to keep communications secure.



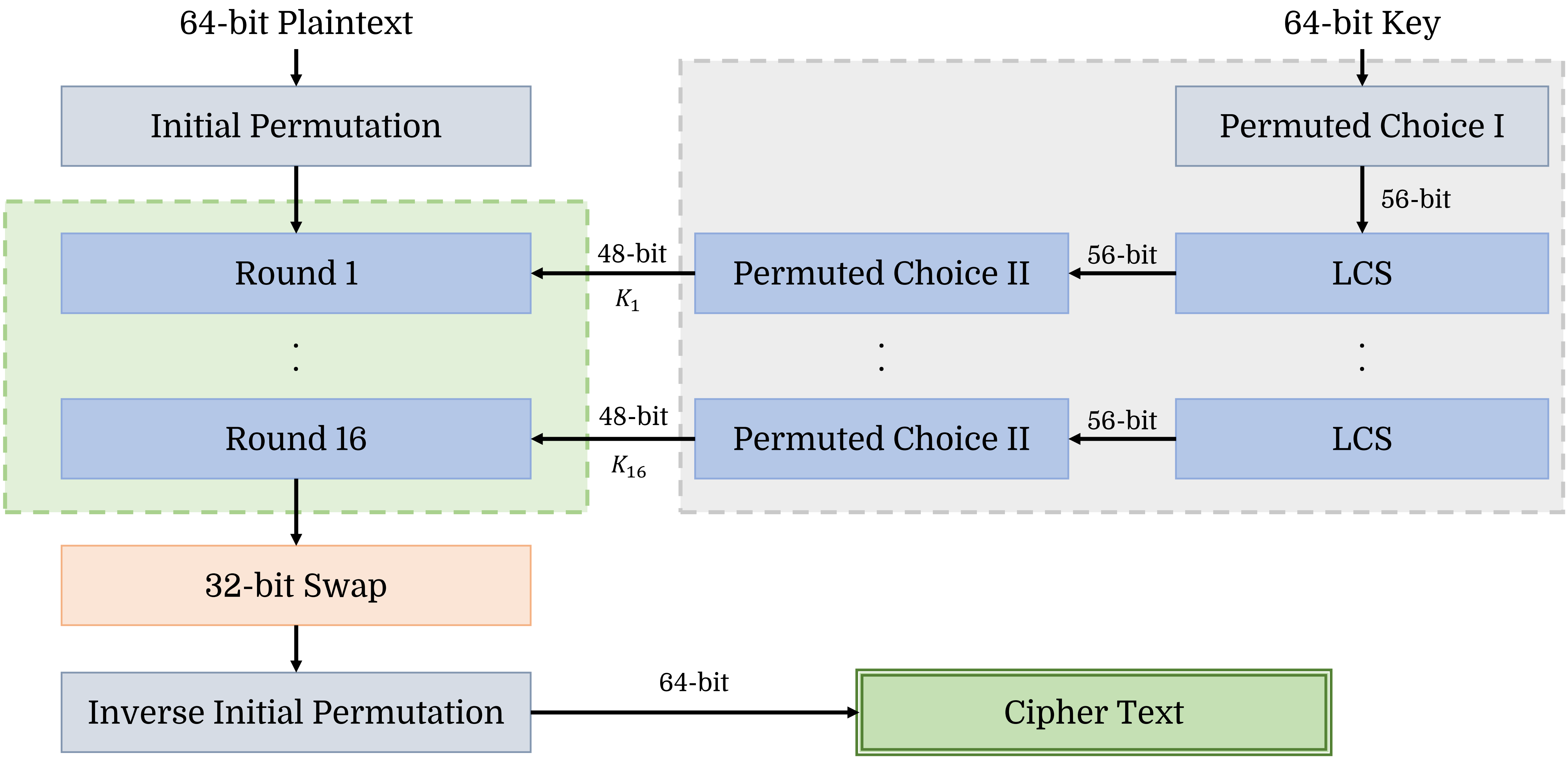
Data Encryption Standard (DES)

- First modern block cipher available to a wide audience.
- Standardized by the NBS, National Bureau of Standards, in 1977.
- Based on the Feistel construction and uses 16 consecutive rounds of Feistel.
- Encrypts 64-bit blocks using 56-bit keys.
- It uses a set of predefined permutations and 8 S-boxes in each of the 16 rounds.
- Each round can be mathematically defined as –

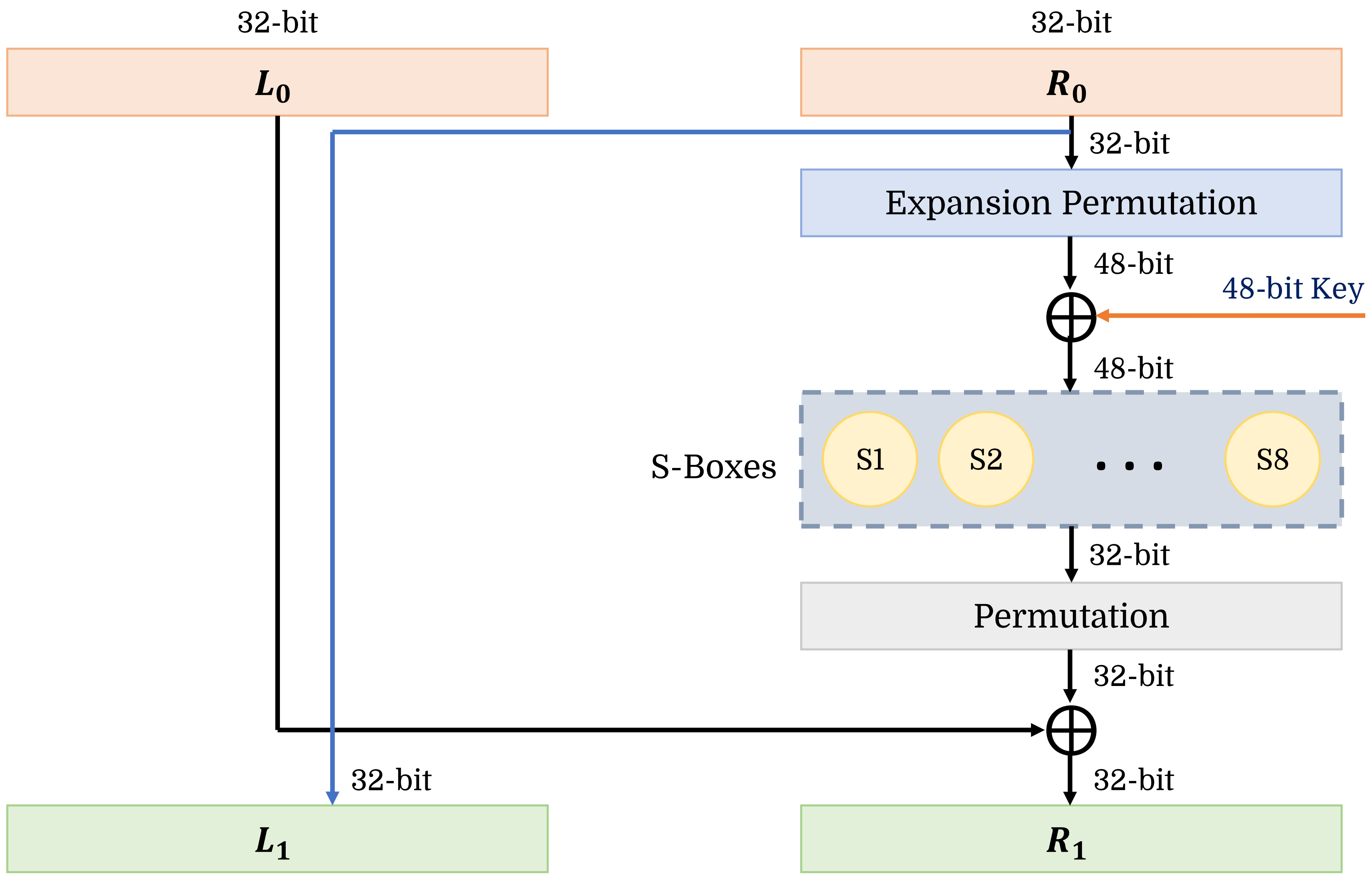
$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{k}) : P \circ S(E(x) \oplus k)$$

- Here, ***E*** is a linear expansion from 32 to 48 bits, ***S*** a non-linear transform consisting of 8 S-boxes from 6 to 4 bits each and ***P*** is a bit permutation on 32 bits and ***k*** is the 48-bit key.

Overview of Working of DES Block Cipher



Overview of a Round Function in DES

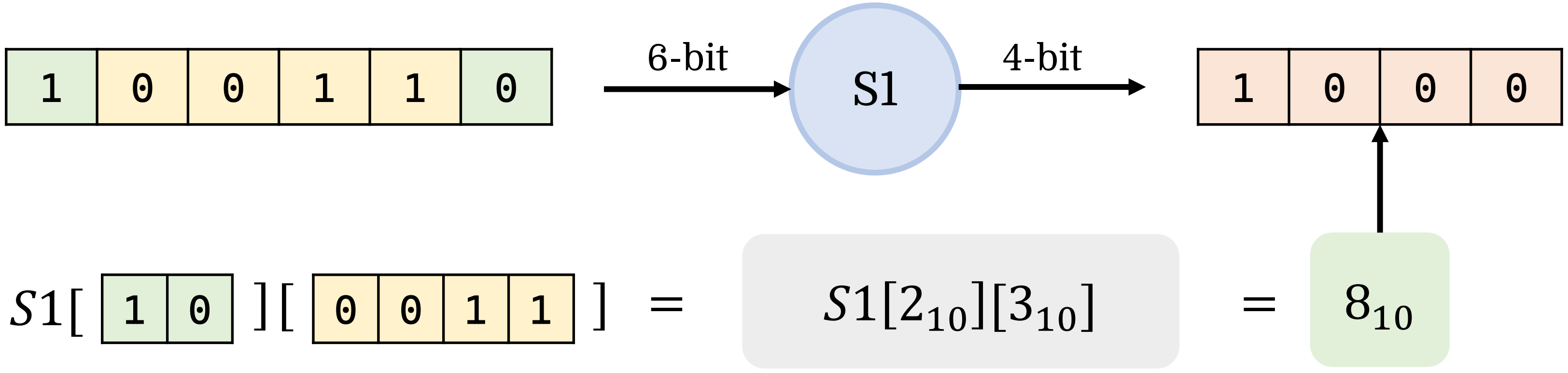


Overview of a S-Box in DES

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Table 1.1: S-box S1

The S-box takes a 6-bit input and gives 4-bit output based on the predefined S-box table, 8 such tables for each S-box (S1 to S8) is defined.



Differential Cryptanalysis of DES Block Cipher

- A general form of cryptanalysis applicable primarily to block ciphers, but also to stream ciphers and cryptographic hash functions.
- It is a chosen plaintext attack.
- It studies how differences in information input can affect the resultant difference at the output.
- It traces differences and non-random behavior to recover the secret key for block ciphers like DES.

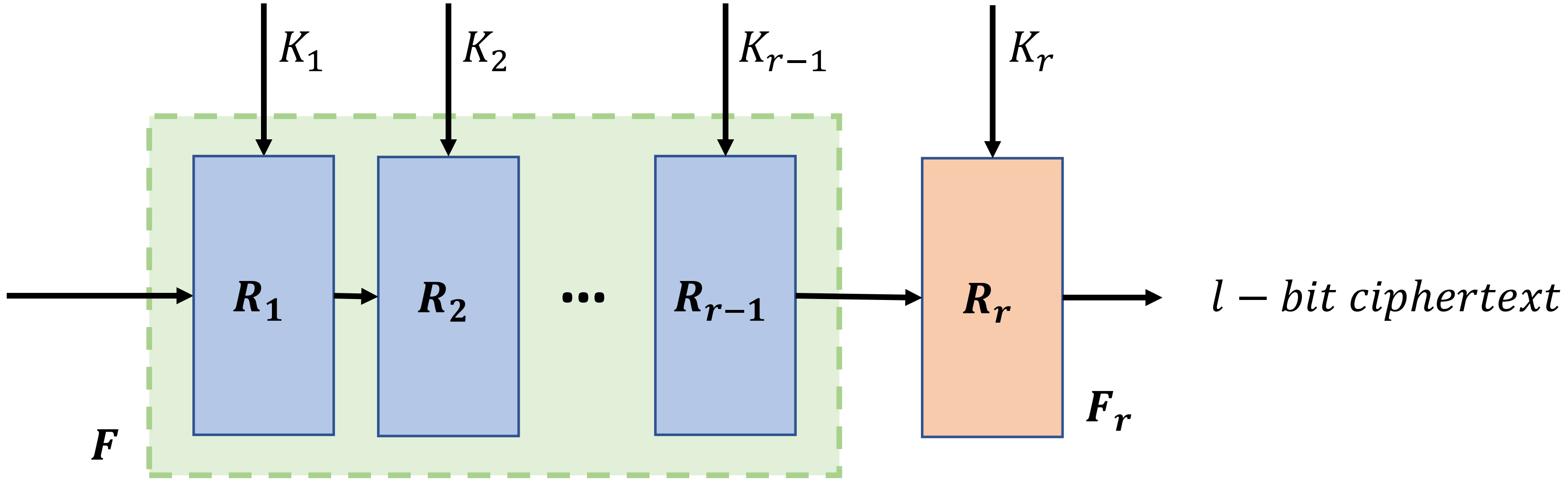


Fig: DES Function

Difference Distribution Table

Considering the S-box given here, we need to find such α and β such that $\alpha \longrightarrow \beta$.

$$x \xrightarrow{\text{6-bit}} \boxed{\boldsymbol{S-box}} \xrightarrow{\text{4-bit}} y$$

$\boldsymbol{\alpha} \downarrow \& \boldsymbol{\beta} \rightarrow$	0	1	2	3	4	5	6	7	. . .	F
0	64	0	0	0	0	0	0	0	. . .	0
1	0	0	0	6	0	2	4	4	. . .	4
2	0	0	0	8	0	4	4	4	. . .	2
3	14	4	2	2	10	6	4	2	. . .	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
3F	4	8	4	2	4	0	2	4		2

Table 1.1: Difference distribution table for S-box S1

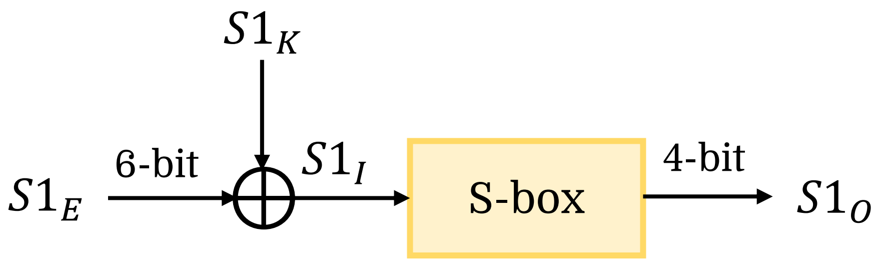
We choose a 6-bit input x_1 and x_2 such that $x_1 \oplus x_2 = \alpha$ and increment the value in table at $(\alpha, y_1 \oplus y_2)$ or at (α, β) by 1.

Key Recovery of a Single S-Box using Input and Output XORs

Output XOR ($S1'_O$)	Possible Inputs ($S1_I$)
1	03, 0F, 1E, 1F , 2A, 2B, 37, 3B
2	04, 05, 0E, 11, 12, 14, 1A, 1B, 20, 25, 26, 2E, 2F , 30, 31, 3A
3	01, 02, 15, 21, 35, 36
4	13, 27
7	00, 08, 0D, 17, 18, 1D, 23, 29, 2C, 34, 39, 3C
8	09, 0C, 19, 2D, 38, 3D
D	06, 10, 16, 1C, 22, 24, 28, 32
F	07, 0A, 0B, 33, 3E, 3F

Table 1.1: Possible input values for the input XOR S1 ′ I = 34x by the output XOR (in hexadecimal)

Assuming we know $S1_E = 1_x$, $S1_E{}^* = 35_x$ and $S1'_O = D_x$ and we need to find $S1_K$. Here taken $S1'_E = S1'_I = 34_x$. By the table 3.2, we get 8 possibilities for input thus 8 possible keys (as $S_K = S_E \oplus S_I$ and $S_E \oplus S_I{}^*$). These keys further may become distinguishable by using a pair with a different input XOR.



Attack on 3-Round DES

The similar process of key retrieval can be deployed on further rounds of DES block cipher but with different input and output XOR values $(S'_I \text{ and } S'_O \text{ respectively})$. Appropriate input and output XORs are generated as shown below for 3 and 6 rounds of DES block cipher.

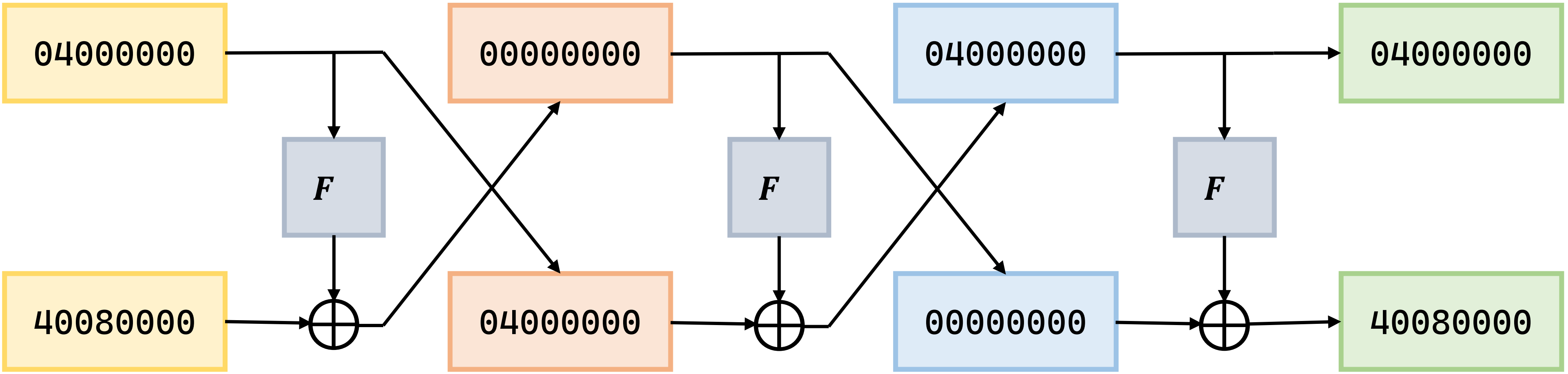
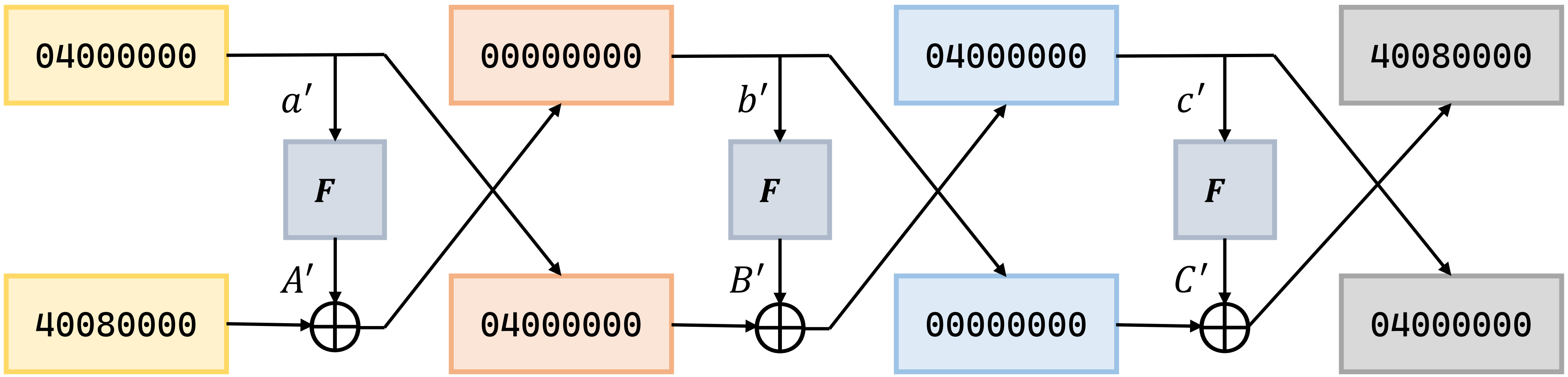


Table 1.1: Input and Output XORs for 3-round DES

Attack on 6-Round DES



Here, $d' = (40\ 08\ 00\ 00)_x$ and $\mathrm{E}(d') =$ 001000 000000 000001 010000 000000 000000 000000 000000

Considering l' *and* r' the left and right 32-bit blocks of output XOR after sixth round. We know

$$l'||r' = swap32bits(IP^{-1}(ciphertext_1) \oplus IP^{-1}(ciphertext_2))$$

The corresponding output XORs in the sixth round can be found by-

$$l' = F' \oplus e'$$
$$e' = D' \oplus c'$$
$$F' = D' \oplus c' \oplus l' \text{ for 5 of the S-boxes}$$

Hash Functions

Hash functions are just functions that take arbitrary-length strings and compress them into shorter strings. Mathematically they can be represented as:

$$h : \{0,1\}^{\infty} \longrightarrow \{0,1\}^{n}$$

Hashing is a process of scrambling a piece of information or data beyond recognition. It is designed to be computationally expensive and practically irreversible.

Features of a Cryptographic Hash Function

Collision Resistance

A hash function H is said to be collision resistant if it is computationally impractical to find $\boldsymbol{x}$ and $\boldsymbol{x}'$ such that $\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{x}')$.

Preimage Resistance

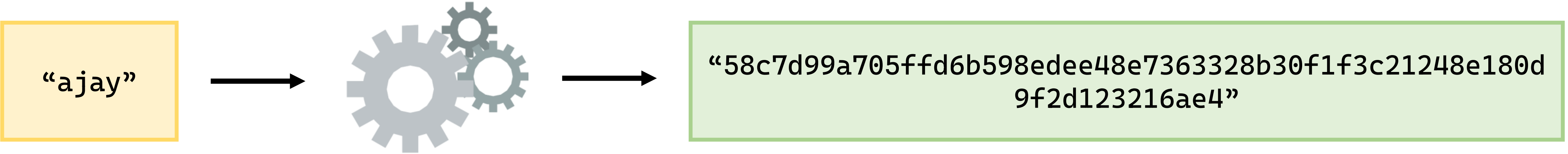
A hash function is said to be preimage resistant if it is computationally impractical for a polynomial-time algorithm to predict the message $\boldsymbol{x}$ given its hash $\boldsymbol{y}$ such that $\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{y}$.

Second Preimage Resistance

A hash function is said to be second preimage resistant if given $\boldsymbol{x}$, it is not feasible for a polynomial-time algorithm to compute $\boldsymbol{y}$ such that $\boldsymbol{H}(\boldsymbol{x}) = \boldsymbol{H}(\boldsymbol{y})$.

BLAKE2s Hash Function

- Cryptographic hash function optimized for 8- to 32-bit platforms
- Produces digests of any size between 1 and 32 bytes
- Similar security standards as SHA-III standard (as claimed by the developers)
- Improvement in speed over BLAKE hash function (close to MD5)
- Variants are: BLAKE2b, BLAKE2s, BLAKE2X, BLAKE2Xb, BLAKE2bp and BLAKE2sp
- Designed by Jean-Philippe Aumasson, Samuel Neves, Zooko Wilcox-O'Hearn and Christian Winnerlein



Message

BLAKE2s Hash Function

Hash Digest (256-bits)

BLAKE vs BLAKE2 Hash Function

Several improvements and modifications over original BLAKE-256 hash function:

- Speed closer to MD5 and less memory requirement
- Fewer rounds, 10 in this case, which is lesser than 14 in BLAKE-256
- Minimal padding and finalization flags
- Fewer constants, BLAKE2s uses 8 word constants instead of 24 in BLAKE-256
- Little endian format, as majority of the target platforms are little-endian
- Counter variable in bytes instead of bits
- Parameter block is xored with the word-constant prior to processing
- On Sandy Bridge, BLAKE2b is 71.99% faster than BLAKE-512, and BLAKE2s is 40.26% faster than BLAKE-256

Working of BLAKE2s Hash Function

I. Initialization

```
IV0 = 0x6A09E667     IV1 = 0xBB67AE85     IV2 = 0x3C6EF372     IV3 = 0xA54FF53A
IV4 = 0x510E527F     IV5 = 0x9B05688C     IV6 = 0x1F83D9AB     IV7 = 0x5BE0CD19
```

$$\begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{pmatrix} \longleftarrow \begin{pmatrix} h_0 & h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 & h_7 \\ IV_0 & IV_1 & IV_2 & IV_3 \\ t_0 \oplus IV_4 & t_1 \oplus IV_5 & f_0 \oplus IV_6 & f_1 \oplus IV_7 \end{pmatrix}$$

Offset	0	1	2	3
0	Digest length	Key length	Fanout	Depth
4	Leaf length			
8	Node offset			
12	Node Offset(cont.)		Node depth	Inner length
16 to 20	Salt			
24 to 28	Personalization			

Parameter block is XORed with the h_0 and later it is passed on to the state variable v_0.

II. Update and Compression

The buffer is set to the message and rest of the 64-bytes are filled with 0 or null bytes. Assuming the message is “ajay”, so the buffer variable will be assigned as:

$$b = 01100001\ 01101010\ 01100001\ 01111001\ 00000000\ \ldots 00000000$$

or

$$b = ajay00 \ldots 00000000$$

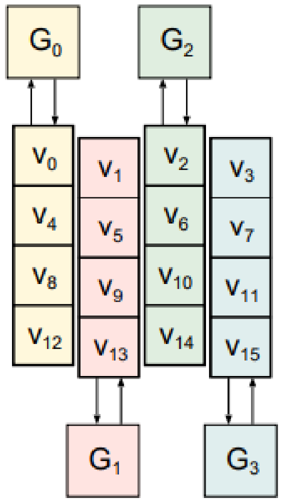
Here, only null bytes are padded after the original message.

- In the compression function, the core function G is run 10 times and the computed hash digest is stored in the $h[8]$ variable.
- The computed hash is transformed to little-endian form and stored as the output hash digest.

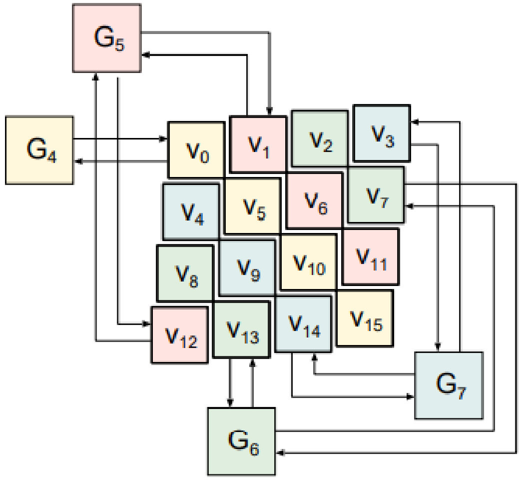
III. The core function G

For the BLAKE2s-256 hash function, the core function is:

$$a \longleftarrow a + b + m_{\sigma_r(2i)}$$
$$d \longleftarrow (d \oplus a) \ggg 16$$
$$c \longleftarrow c + d$$
$$b \longleftarrow (b \oplus c) \ggg 12$$
$$a \longleftarrow a + b + m_{\sigma_r(2i+1)}$$
$$d \longleftarrow (d \oplus a) \ggg 8$$
$$c \longleftarrow c + d$$
$$b \longleftarrow (b \oplus c) \ggg 7$$



Column Step

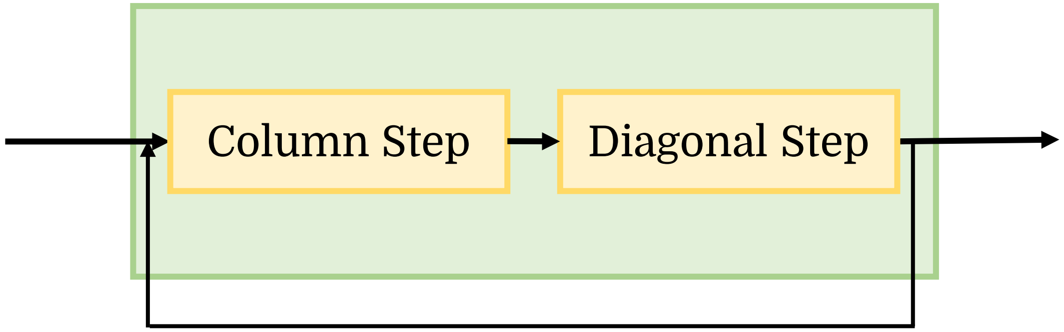


Diagonal Step

Here, ***G*** is applied to the word-matrix in two ways, firstly, ***G*** is applied to each column with 4 state variables and next the core function ***G*** is applied diagonally. Then new round begins.

- r is the round number and varies between $0\ to\ 9$,
- i varies between $0\ to\ 7$,
- " + " here refers to addition

Round function



Here, $(a \ggg s)$ means right shift operation by $\boldsymbol{s - bit}$ on $\boldsymbol{a}$.

Work Done and Work Planned

Work Done

- Understood working of hash functions by implementing MD5 hash function.
- Got an idea of encryption and differential cryptanalysis by implementing DES block cipher and differential cryptanalysis on 6-round DES.
- Implemented BLAKE-256 to derive idea about the working of BLAKE hash functions.
- Implemented BLAKE2s-256 and derived a perception of modifications done in BLAKE2.

Work Planned

- Study of differential cryptanalysis of BLAKE2s hash function.
- Implementing the differential attack on BLAKE2s.
- Exploring other attacks, the BLAKE2s hash function may be susceptible to.
- Finding the scope of improvement in current attacks.

References

- Differential Cryptanalysis of the Data Encryption Standard by Eli Biham, Adi Shamir, Springer-Verlag New York, Inc. 1993.
- SHA-3 proposal BLAKE, version 1.3. [Source]- SHA-3 proposal BLAKE (aumasson.jp).
- Aumasson, JP., Neves, S., Wilcox-O'Hearn, Z., Winnerlein, C. (2013). BLAKE2: Simpler, Smaller, Fast as MD5.