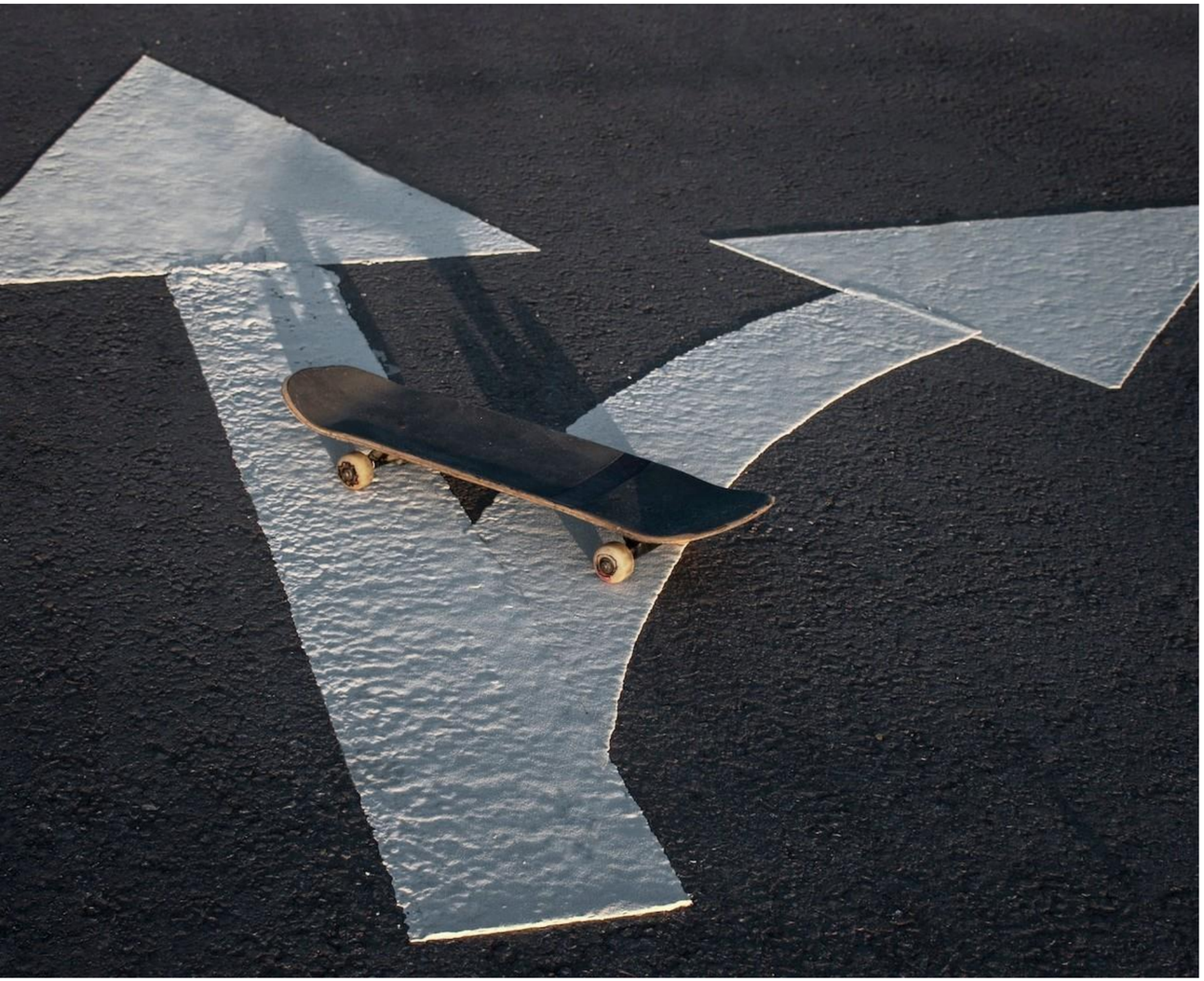


Blowfish ALGORITHM

History

- **Blowfish** is an encryption technique designed by **Bruce Schneier** in 1993 as an alternative to DES Encryption Technique.

- It is significantly faster than DES and provides a good encryption rate with no effective cryptanalysis technique found to date.

- It is one of the first, secure block cyphers not subject to any patents and hence freely available for anyone to use. It is symmetric block cipher algorithm.

- The name "Blowfish" is not related to the fish but rather comes from the fact that Schneier named many of his early encryption algorithms after fish. The specific choice of "Blowfish" was likely influenced by the fact that it sounded interesting and distinctive.

KEY FEATURES

1. **Symmetric-Key Algorithm:** Blowfish uses the same key for both encryption and decryption processes, making it a symmetric-key algorithm. This means that the party encrypting the data and the party decrypting it must possess the same secret key.

2. **Block Cipher:** Blowfish operates on fixed-size blocks of data. The standard block size is 64 bits, but it can work with smaller blocks as well. If the input data is not a multiple of the block size, padding is typically applied to the data before encryption.

3. **Variable-Length Key:** One of the unique features of Blowfish is its ability to accept variable-length encryption keys, making it adaptable to different security requirements. The key length can range from 32 to 448 bits, and it's expanded during encryption to generate a series of subkeys.

4. **Feistel Network Structure:** Blowfish employs a Feistel network structure in which data is divided into two halves, subjected to a series of rounds of operations, and then recombined. This structure allows for efficient encryption and decryption processes.

5. **F-Function:** The F-function is a core component of the Blowfish algorithm. It involves a combination of XOR (exclusive OR), substitution, and permutation operations, which contribute to the algorithm's strength and security.

6. **Key Expansion:** Before the actual encryption process, Blowfish generates a series of subkeys based on the provided key. These subkeys are used during the encryption and decryption rounds to introduce complexity and security.

7. **Complexity and Security:** Blowfish is designed to be highly secure against various cryptographic attacks. The complex F-function and key expansion process make it resistant to brute force and differential cryptanalysis.

How does The Blowfish Algorithm work?

Step 1. Key Generation and Subkey Creation

- 18 subkeys{P[0]...P[17]} are needed in both encryption as well as decryption process and the same subkeys are used for both the processes.
- These 18 subkeys are stored in a P-array with each array element being a 32-bit entry.
- It is initialized with the digits of pi(?).
- The hexadecimal representation of each of the subkeys is given by:

32-bit hexadecimal representation of initial values of sub-keys

P[0]	243f6a88	P[9]	38d01377
P[1]	85a308d3	P[10]	be5466cf
P[2]	13198a2e	P[11]	34e90c6c
P[3]	03707344	P[12]	c0ac29b7
P[4]	a4093822	P[13]	c97c50dd
P[5]	299f31d0	P[14]	3f84d5b5
P[6]	082efa98	P[15]	b5470917
P[7]	ec4e6c89	P[16]	9216d5d9
P[8]	452821e6	P[17]	8979fb1b

How does The Blowfish Algorithm work?

Step 2: Initialise Substitution Boxes:

- 4 Substitution boxes (S-boxes) are needed{S[0]...S[4]} in both encryption as well as decryption process with each S-box having 256 entries{S[i][0]...S[i][255]} where each entry is 32-bit.
- It is initialized with the digits of **pi** after initializing the P-array.

Source:
https://github.com/Ray784/Blowfish-S-boxes

1	S-box 1							
2	d1310ba6	98dfb5ac	2ffd72db	d01adfb7	b8e1afed	6a267e96	ba7c9045	f12c7f99
3	24a19947	b3916cf7	0801f2e2	858efc16	636920d8	71574e69	a458fea3	f4933d7e
4	0d95748f	728eb658	718bcd58	82154aee	7b54a41d	c25a59b5	9c30d539	2af26013
5	c5d1b023	286085f0	ca417918	b8db38ef	8e79dcb0	603a180e	6c9e0e8b	b01e8a3e
6	d71577c1	bd314b27	78af2fda	55605c60	e65525f3	aa55ab94	57489862	63e81440
7	55ca396a	2aab10b6	b4cc5c34	1141e8ce	a15486af	7c72e993	b3ee1411	636fbc2a
8	2ba9c55d	741831f6	ce5c3e16	9b87931e	afd6ba33	6c24cf5c	7a325381	28958677
9	3b8f4898	6b4bb9af	c4bfe81b	66282193	61d809cc	fb21a991	487cac60	5dec8032
10	ef845d5d	e98575b1	dc262302	eb651b88	23893e81	d396acc5	0f6d6ff3	83f44239
11	2e0b4482	a4842004	69c8f04a	9e1f9b5e	21c66842	f6e96c9a	670c9c61	abd388f0
12	6a51a0d2	d8542f68	960fa728	ab5133a3	6eef0b6c	137a3be4	ba3bf050	7efb2a98
13	a1f1651d	39af0176	66ca593e	82430e88	8cee8619	456f9fb4	7d84a5c3	3b8b5ebe
14	e06f75d8	85c12073	401a449f	56c16aa6	4ed3aa62	363f7706	1bfedf72	429b023d
15	37d0d724	d00a1248	db0fead3	49f1c09b	075372c9	80991b7b	25d479d8	f6e8def7
16	e3fe501a	b6794c3b	976ce0bd	04c006ba	c1a94fb6	409f60c4	5e5c9ec2	196a2463
17	68fb6faf	3e6c53b5	1339b2eb	3b52ec6f	6dfc511f	9b30952c	cc814544	af5ebd09
18	bee3d004	de334afd	660f2807	192e4bb3	c0cba857	45c8740f	d20b5f39	b9d3fbdb
19	5579c0bd	1a60320a	d6a100c6	402c7279	679f25fe	fb1fa3cc	8ea5e9f8	db3222f8
20	3c7516df	fd616b15	2f501ec8	ad0552ab	323db5fa	fd238760	53317b48	3e00df82
21	9e5c57bb	ca6f8ca0	1a87562e	df1769db	d542a8f6	287effc3	ac6732c6	8c4f5573
22	695b27b0	bbca58c8	e1ffa35d	b8f011a0	10fa3d98	fd2183b8	4afcb56c	2dd1d35b
23	9a53e479	b6f84565	d28e49bc	4bfb9790	e1ddf2da	a4cb7e33	62fb1341	cee4c6e8
24	ef20cada	36774c01	d07e9efe	2bf11fb4	95dbda4d	ae909198	eaad8e71	6b93d5a0
25	d08ed1d0	afc725e0	8e3c5b2f	8e7594b7	8ff6e2fb	f2122b64	8888b812	900df01c
26	4fad5ea0	688fc31c	d1cff191	b3a8c1ad	2f2f2218	be0e1777	ea752dfe	8b021fa1
27	e5a0cc0f	b56f74e8	18acf3d6	ce89e299	b4a84fe0	fd13e0b7	7cc43b81	d2ada8d9
28	165fa266	80957705	93cc7314	211a1477	e6ad2065	77b5fa86	c75442f5	fb9d35cf
29	ebcdaf0c	7b3e89a0	d6411bd3	ae1e7e49	00250e2d	2071b35e	226800bb	57b8e0af
30	2464369b	f009b91e	5563911d	59dfa6aa	78c14389	d95a537f	207d5ba2	02e5b9c5
31	83260376	6295cfa9	11c81968	4e734a41	b3472dca	7b14a94a	1b510052	9a532915
32	d60f573f	bc9bc6e4	2b60a476	81e67400	08ba6fb5	571be91f	f296ec6b	2a0dd915
33	b6636521	e7b9f9b6	ff34052e	c5855664	53b02d5d	a99f8fa1	08ba4799	6e85076a

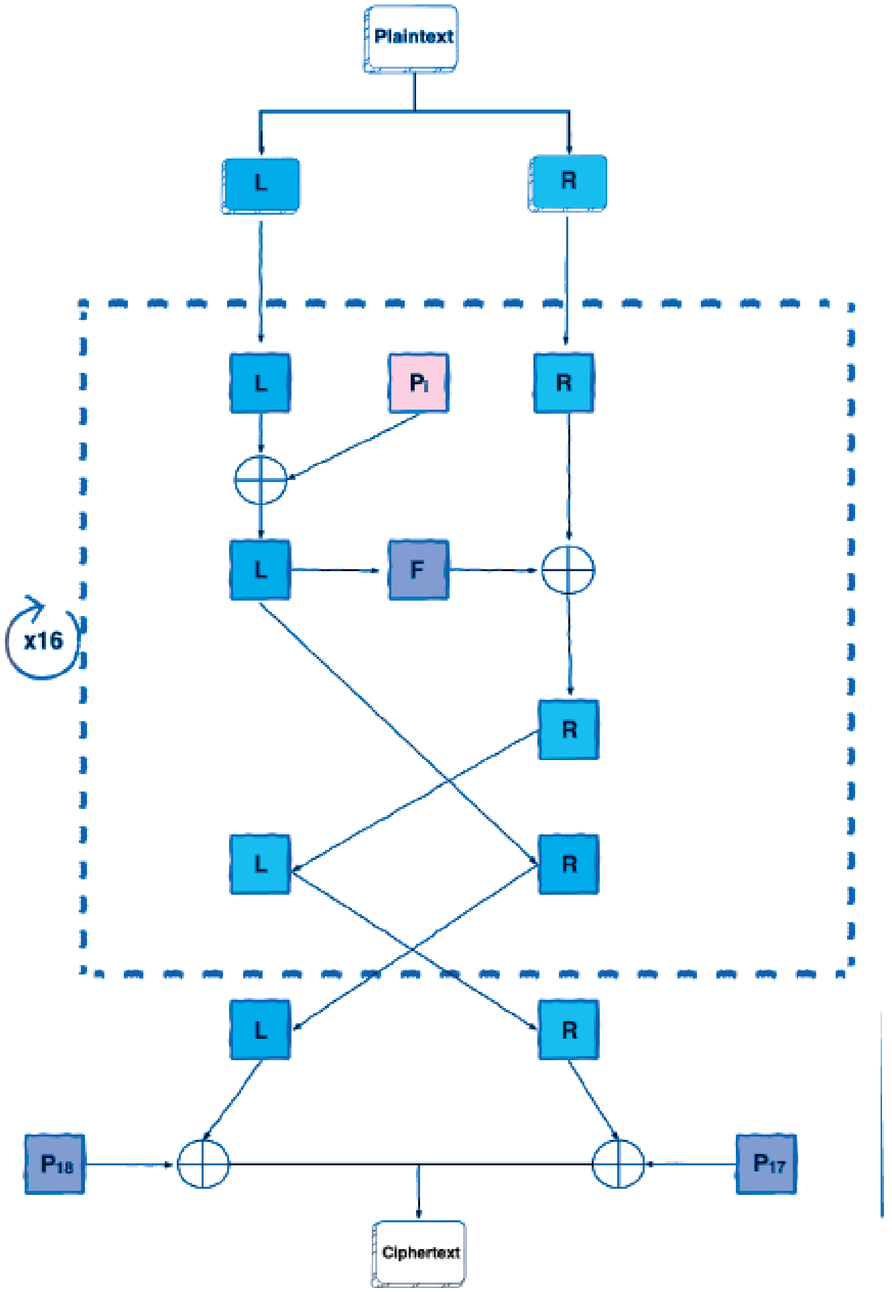
How does The Blowfish Algorithm work?

Step 3: Encryption:

Once the subkeys are generated, the algorithm proceeds with the encryption of the data block. The data block is divided into two 32 bit halves, L (left) and R (right). A series of rounds (typically 16) are performed on these halves to ensure strong encryption.

Feistel Network Rounds:

The algorithm employs a Feistel network structure, which involves applying a series of operations to the L and R halves in each round. These operations include XOR (exclusive OR) with the current subkey, applying the F function to R, and swapping L and R.

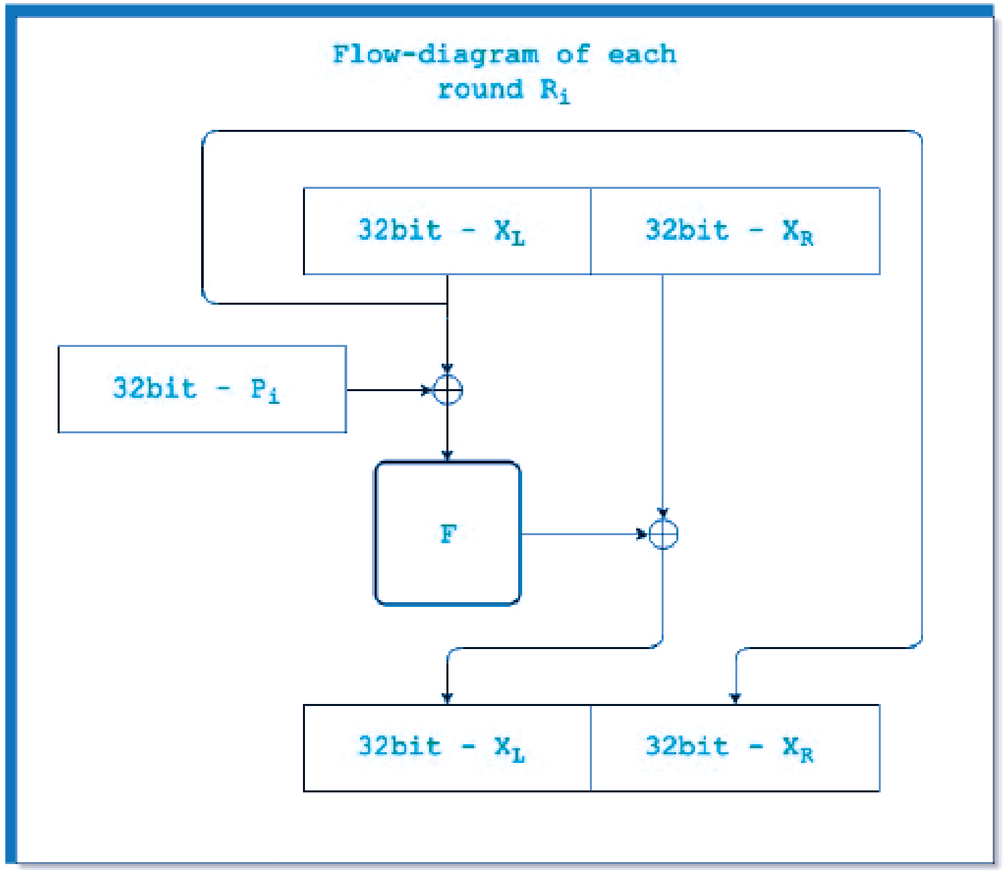


How does The Blowfish Algorithm work?

Step 3: Encryption:

F function Operation:

- The current subkey Pi is XORed with L.
- The F function takes the 32 bit output of the XOR operation and applies several steps



How does The Blowfish Algorithm work?

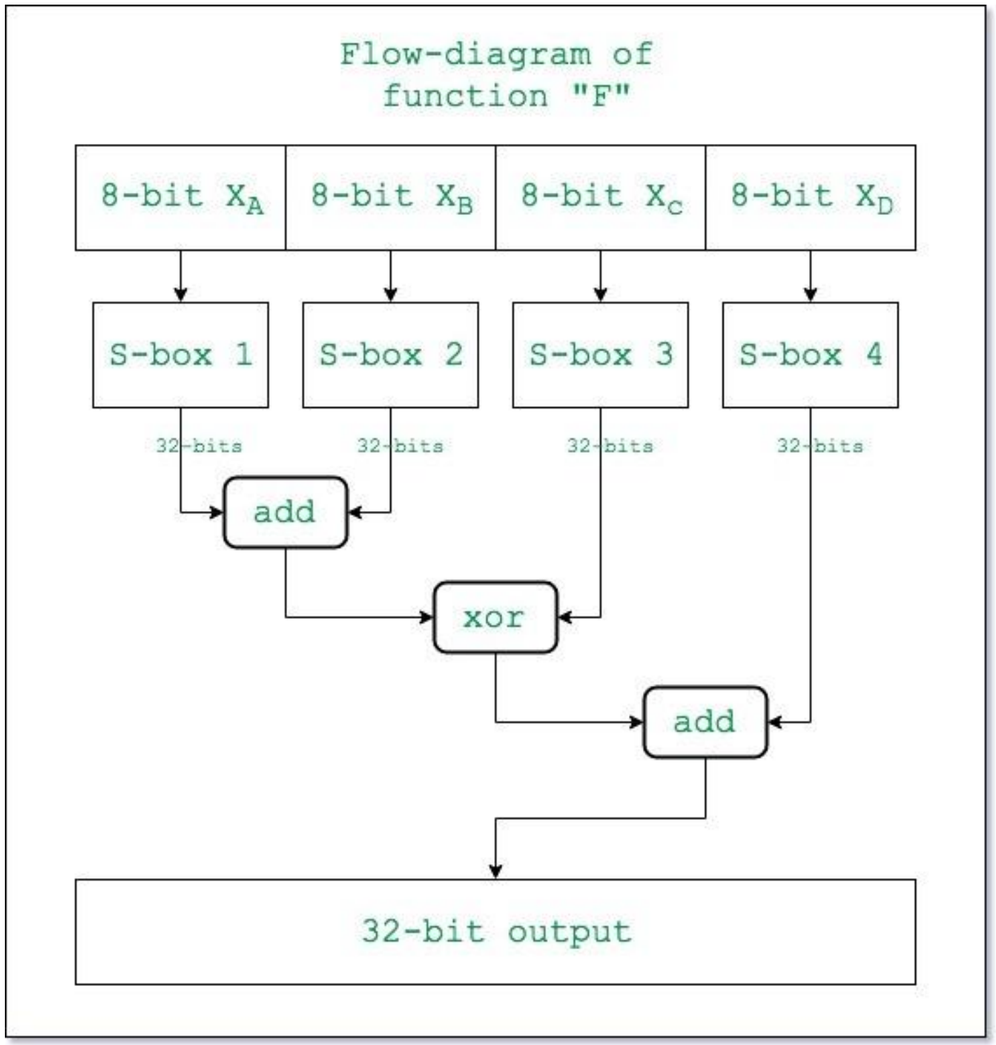
Step 3: Encryption:

F function Operation:

- L is divided into four 8 bit quarters. Each quarter is used to index a specific S box, and the resulting values are combined.

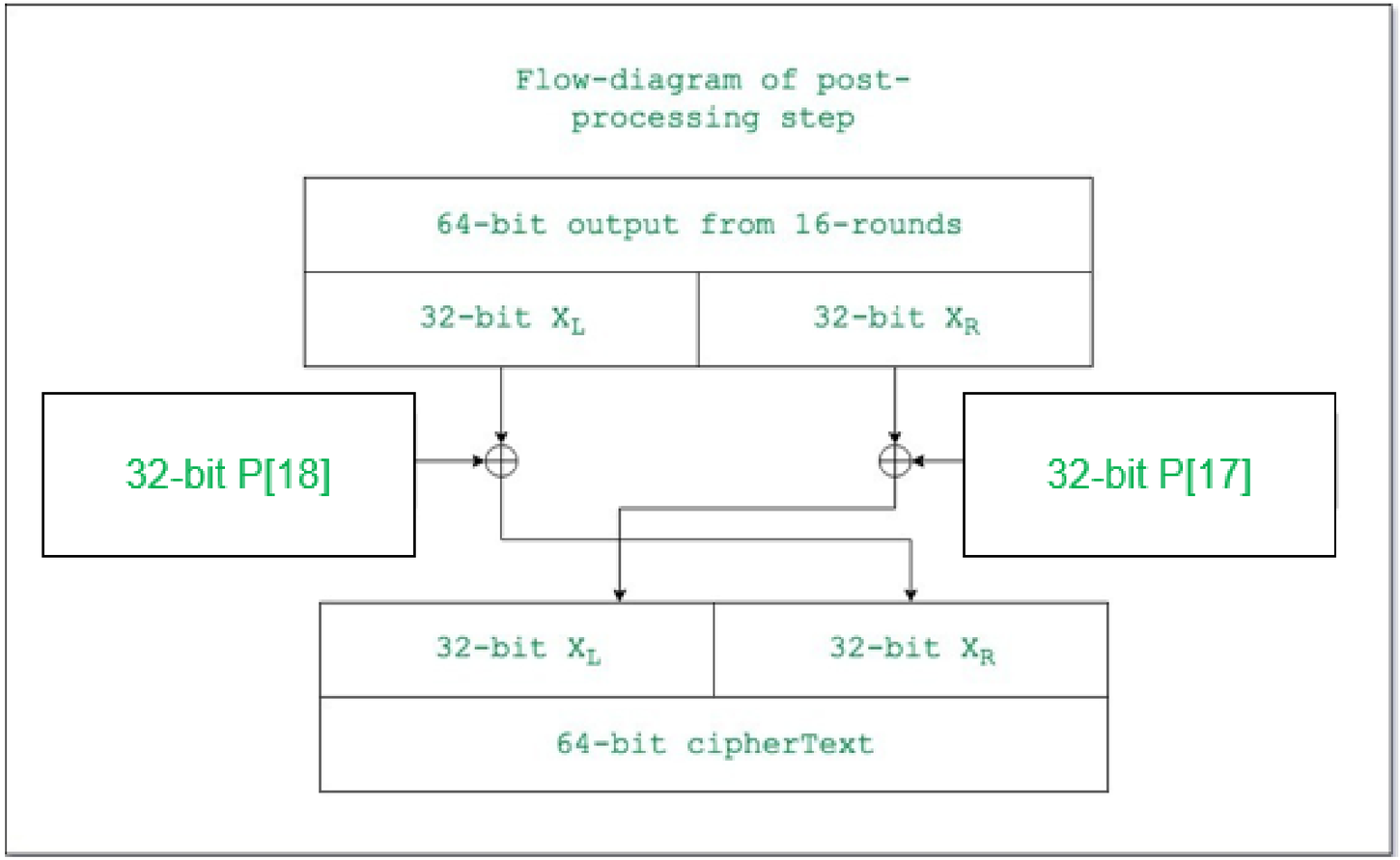
- Here the function “add” is addition modulo 2^32

Permutation The results from the S boxes are combined and transformed using the P array.



How does The Blowfish Algorithm work?

Step 4: Post Processing: The output after the 16 rounds is processed as shown below:



Decryption Side

Similar to encryption, rounds involve applying operations to L and R, but this time in reverse order using the corresponding subkey.

1.F function Operation (Decryption) The F function is applied in reverse, with the subkey XOR and S box steps inverted. This reverse operation successfully decrypts the data block.

2.Final Round (Decryption) After all decryption rounds, the decrypted L and R halves are combined to obtain the original data block.

Conclusion

- The Blowfish algorithm’s security lies not only in its use of the Feistel network structure and the F function but also in its intricate subkey generation process. By meticulously expanding the original key into a series of subkeys and performing numerous rounds of operations, Blowfish ensures that the encrypted data remains secure and resistant to various attacks.

- Blowfish is considered secure and has not been **"cracked"** in the sense of a practical cryptanalysis attack that would compromise its security significantly. However, it's essential to note that Blowfish is an aging algorithm, and its successor, Twofish, was designed by the same author, Bruce Schneier, to provide a higher security margin.

- While Blowfish remains unbroken, it's recommended to use more modern encryption algorithms like Advanced Encryption Standard (AES) for new applications due to their widespread adoption, extensive analysis, and ongoing support. AES has become the industry standard for symmetric-key encryption and is considered highly secure.

- Keep in mind that the security of any cryptographic algorithm can be affected by advances in technology, computing power, and new cryptographic attacks. Therefore, it's essential to stay informed about the latest developments and consider using algorithms that are widely recognized and recommended by the cryptographic community.

Thanks!

Do you have any questions?

c.sridhar89@gmail.com
www.iamsridhariyer.com
youtube: @sridhariyer