ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

Optimised Design of Light Weight Block Cipher Lilliput with Extended Generalised Feistal Network (EGFN)

Mumthaz Pookuzhy Ali[1], Geethu T George[2]

P.G. Student, Dept. of Electronics and Communication Engineering, Indira Gandi College of Engineering and Technology for Womens, Nellikuhzy, Kothamangalam, India[1]

Assistant Professor, Dept. of Electronics and Communication Engineering, Indira Gandi College of Engineering and Technology for Womens, Nellikuhzy, Kothamangalam, India[2]

ABSTRACT: Lilliput is a 64-bit block cipher with an 80-bit key. It uses an Extended Generalized Feistel Network (EGFN) with 64-bit state and a round function acting at nibble level. To reduce the delay and increase the speed of operation ,a modification in its layer of algorithm is done. On implementing with a 'PRESENT' cryptography method in its algorithm there is that delay reduction is possible. The light weight cipher thus enhances all its merits in this modified work.

KEYWORDS: Encryption, Feistal network, block cipher, substitution layer.

I. INTRODUCTION

Lightweight cryptography has been a major topic for the last few years, driven by the lack of primitives capable to run on devices with very low computing power. At the core of lightweight cryptography is a trade-off between lightweightness and security and to reach high levels of security using only a small computing power have addressed by suggesting lightweight streamciphers, blockciphers, hash function and recently one-pass authenticated encryption etc.

II. RELATED WORK

1) While a classical Feistel network, such as DES Data Encryption Standard, National Bureau of Standards, U. S. Department of Commerce, 1977.or Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms, divides a plaintext into 2 n-bit-long halves and applies a function F to half of the state before adding the result to the other half, a Generalized Feistel Network (GFN) divides it into k >2 n-bit-long subblocks and potentially applies a different F function to each subblock.

2) Various GFNs exist in the literature. This includes Type-1 as in CAST-256 [Encryption Algorithm] where a single F function is used at each round; Type-2 as in HIGHT [HIGHT: A New Block Cipher Suitable for Low-Resource Device] and CLEFIA [128-Bit Blockcipher] and Nyberg's GFNs [Generalized Feistel Networks] where k=2 F functions are used; Type-3, Source-Heavy (SH) as in SHA-1 [Secure Hash Standard] and Target-Heavy (TH) as in MARS [an AES candidate, NIST AES Proposal] where more than k=2 F functions are used. The pseudorandomness of these constructions is studied in Type-1, Type-2 and Type-3 and in the Pseudorandomness of Top-Level Schemes of Block Ciphers,

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

3) On the Construction of Pseudorandom Permutations: Luby-Rackoff Revisited On Generalized Feistel Networks in Advances in Cryptology - CRYPTO 2010 for SH and TH GFNs. Fig. 1 gives an example of Type-2 GFN. Usually GFNs perform a block-wise cyclic shift in their permutation layer.

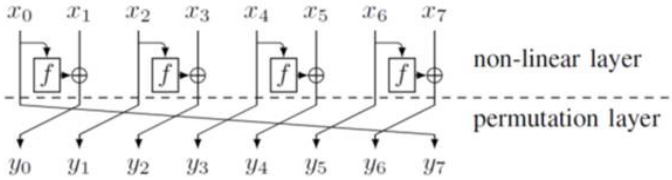

Fig 1: One round of a Type-2 GFN with k=8 blocks

4) In Improving the Generalized Feistel in Fast Software Encryption - FSE 2010, and in Improving the Permutation Layer of Type 1, Type 3, Source-Heavy, and Target-Heavy Generalized FeistelStructure , two more generic studies are provided: the authors analyzed Type-1, Type-2 and Type-3 GFNs regarding non-cyclic permutations.

5) More precisely, in Improving the Generalized Feistel in Fast Software Encryption - FSE 2010, the concept of maximum diffusion round is introduced. It corresponds to the minimum number of rounds such as every output block depends on every input block. It is a quantifiable measure to maximize the diffusion inside the cipher. The authors of Improving the Generalized Feistel in Fast Software Encryption - FSE 2010 exhaustively searched all the optimum permutations for $k < 16$ and found that the diffusion in Type-2 GFNs can be improved. They also showed a lower bound on the maximum diffusion round of Type-2 GFNs and when k is a power of 2, they gave a generic construction based on de Bruijn graphs whose maximum diffusion round is close to the lower bound they found. Besides, they studied the pseudorandomness of these GFNs giving a bound depending on the diffusion delay and their resistance against classical attacks and showed that it is actually improved as well.

6) One of these Type-2 GFNs is used in TWINE : A Lightweight Block Cipher for Multiple Platforms in Selected Areas in Cryptography - SAC 2012 . In 15] Improving the Permutation Layer of Type 1, Type 3, Source-Heavy, and Target-Heavy Generalized Feistel Structures , Yanagihara and Iwata gave the same kind of constructions for Type-1, Type-3, SH and TH GFNs with non-cyclic permutation. For Type-1 and Type-3 GFNs, they showed that the maximum diffusion delay can be improved by changing the permutation while for SH and TH GFNs it cannot.

III. EXTENDED GENERALIZED FEISTEL NETWORKS (EGFNS)

Introducing Extended Generalized Feistel Networks (EGFNs) as one possible generalization of GFNs. We then introduce a generic EGFN family with good full diffusion delay and cheap cost and we instantiate it for $k = 16$ and a particular permutation layer. For a GFN $M = PF$, to achieve quicker diffusion, one can increase the number of round-functions in F. However, this also makes costlier GFNs. The other possibility is to look at the permutation layer P. Definition 1 already allows for block-wise permutations. A possible generalization is to use a linear mapping instead, thus looking for GFNs $M= GF$ with G an invertible kxk matrix.

This is however much costlier than a simple block-wise permutation and it loses the quasiinvolutive property. What we propose is to have a G which is itself a GFN but with the identity mapping as round-function. In other words, we write $G = PL$ where P is a permutation matrix and L is matrix similar to F but with I off-diagonal non-zero coefficients instead of F. We call this matrix L the linear layer. In that case, the whole Feistel network matrix becomes $M= PLF$, e.g. Fig. 3. Because matrices L and F have common structure, we regroup them into a single matrix $N = LF$, and write $M = PN$. The matrix N is the new function part of the GFN that contains the non-linear roundfunctions but now has two formal parameters: F for non-linear round-functions to provide cryptographic security and I for identity round-functions to provide quick diffusion. We call these new schemes Extended Generalized Feistel Networks. For GFNs, to be

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

considered an EGFN we require that matrixM= PN is invertible and that det(M) does not depend on F nor I, which translates into det(N) =1. Again, we choose to focus on EGFNs that are quasi involute.

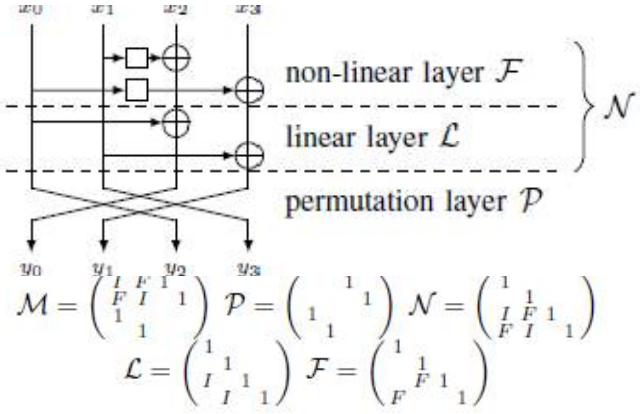

Fig. 2.An EFGN with three layers and corresponding matrix

III.a. MODIFIED FORM OF EGFN WITH SUBSTITUTION LAYER

One important type of iterated block cipher known as a substitution-permutation network (SPN) takes a block of the plaintext and the key as inputs, and applies several alternating rounds consisting of a substitution stage followed by a permutation stage—to produce each block of ciphertext output. The non-linear substitution stage mixes the key bits with those of the plaintext, creating Shannon's confusion. The linear permutation stage then dissipates redundancies, creating diffusion.

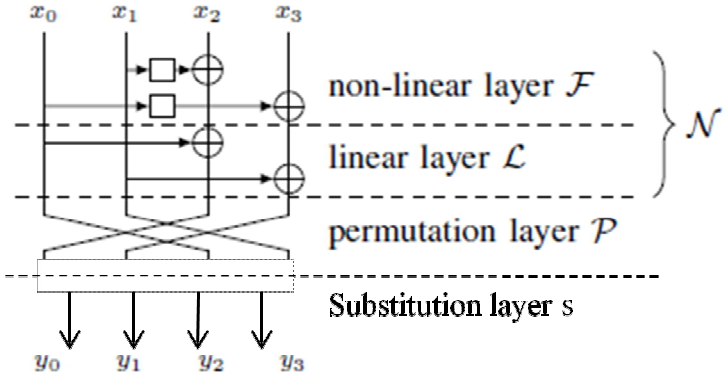

Fig 3: The modified EGFN with S-Layer

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

The S-box used in present is a 4-bit to 4-bit S-box S : F42→ F42. The action of this box in hexadecimal notation is given by the following table.ForsBoxLayer the current state b63 . . . b0 is considered as sixteen 4-bit words w15 . . .w0 where wi = b4∗i+3||b4∗i+2||b4∗i+1||b4∗i for 0 ⩽i⩽ 15 and the output nibble S[wi] provides the updated state values in the obvious way.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

A substitution box (S-box) substitutes a small block of input bits with another block of output bits. This substitution must be one-to-one, to ensure invertibility (hence decryption). A secure S-box will have the property that changing one input bit will change about half of the output bits on average, exhibiting what is known as the avalanche effect—i.e. it has the property that each output bit will depend on every input bit.At each round, the round key (obtained from the key with some simple operations, for instance, using S-boxes and P-boxes) is combined using some group operation, typically XOR.

IV. LILLIPUT LIGHT WEIGHT CIPHER ENCRYPTION

LILLIPUT is a 64-bit block cipher with an 80-bit key. Thewhole encryption process is depicted in Fig. 5. As previously explained, LILLIPUT uses an Extended Generalized FeistelNetwork (EGFN) with 64-bit state and a round function actingat nibble level. The state X is seen as 16 nibbles. Thesenibbles are denoted X15; ……;X0. It is composed of 30 rounds,i.e. 30 repetitions of a single EGFN called OneRoundEGFN, depicted in Fig. 4 where each Fi for i from 0 to 7 is defined as Fi = S(X7-I RKi) where S is an S-box that acts at nibble level and where RKi is the nibble of position i of the 32-bit subkeyRKj of the round j. The 30 32-bit subkeysRKj are generated from the master key using the key schedule. Note that the last round misses a PermutationLayer for involution reasons.

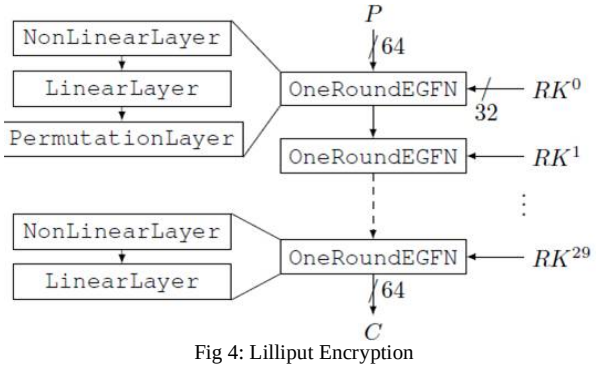

Fig 4: Lilliput Encryption

V. KEY SCHEDULE

The key schedule depicted produces the 30 subkeys RK0 to RK29 from the master key K and is designed to allow on-the-fly computations. It uses an 80-bit Linear Finite State Machine (LFSM). Its inner state is denoted Y and is initialized with the master key K. The subkeys are extracted from the LFSM state Y using ExtractRoundKey, a layer of parallel S-boxes. Subkey RK0 is extracted from the LFSM initial state (i.e. the master key K). The LFSM state Y is then updated using RoundFnLFSM and the next subkey RK1 is extracted. And so on until RK29. The state Y is made of 20 nibbles Y19; _ _ _ ; Y0 split among 4 LFSRs, L0 to L3, acting on 5 nibbles each: L0 acts on Y0 to Y4, L1 on Y5 to Y9 and so on. We use here LFSRs inspired by the results on LFSRs. As described our 4 LFSRs are nibble oriented and seen as a Feistel network. More precisely, using the matrix representation introduced it is possible to construct word-oriented LFSRs with word feedbacks well chosen to fit with a Feistel representation generalizing the classical

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

Fibonacci and Galois representations. Moreover, the operations used to compute the feedbacks are simple wordoriented operations such as shifts or rotations. The two main advantages of those constructions are a clear speed-up of diffusion and -due to the Feistel construction- a simple reverse. The 4 LFSRs used in the LILLIPUT key schedule are Feistel-like word-oriented LFSRs acting on 5 nibbles. They are completely described in the following . Thus, the RoundFnLFSM function seen as 4 Feistel-like word-oriented LFSRs can be divided into two transformations: MixingLFSM which holds the feedbacks, followed by PermutationLFSM which is the word-wise cyclic shift, as depicted. The precise description of both transformations is given below.

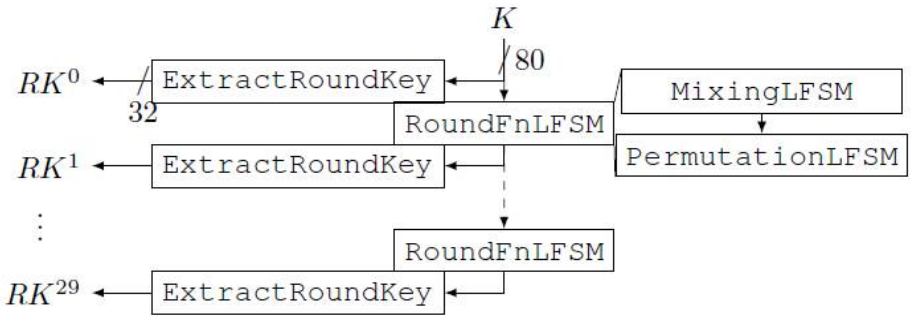

Fig 5: Lilliput key schedule

MixingLFSM: For each of the 4 parallel LFSRs L0 to L3 it consists in xoring some nibbles to some others nibbles in a Feistel-like manner:

For L0: Y0 ←Y0⊕ (Y4>>>1) and Y1 ・Y1⊕ (Y2>>3)
For L1: Y6 ←Y6⊕ (Y7<<3) and Y9 ・Y9⊕ (Y8<<<1)
For L2: Y11 ←Y11⊕ (Y12>>>1) and Y13 ・Y13⊕ (Y12>>3)
For L3: Y16 ←Y16⊕ (Y15<<3) ⊕ (Y17<<<1)

PermutationLFSM: For each of the 4 parallel LFSRs L0 to L3, it consists in a left cyclic shift of its 5 nibbles, i.e. Yi Yi-1 mod 5. ExtractRoundKey; each round of encryption, the subkeyRKi is extracted from the LFSM inner state using a non linear extracting function in a bitsliced way. First, some nibbles of the state are extracted: Z(32) Y18jjY16jjY13jjY10jjY9jjY6jjY3jjY1 and let Z31;…….;Z0 be the bits of Z then RKi j = S(Zj jjZ8+j jjZ16+j jjZ24+j) where S is the same S-box as in the datapath.

VI. LILLIPUT DECRYPTION

As LILLIPUT is a Feistel network, decryption is quite analogous to encryption but uses the inverse block permutation and the subkeys in the reverse order. Note that NonLinearLayer and LinearLayer commute, hence can be computed in any order. Besides, as the LFSM of the key schedule is made of Feistel-like LFSRs, the same argument leads to the ability to compute the different subkeys in the reverse order starting from the LFSM inner state Y.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

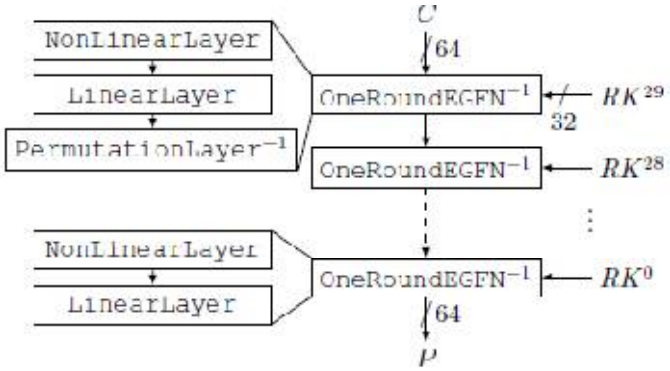

Fig 6 Lilliput Decryption

As done in other lightweight block ciphers, such as PRESENT, TWINE ,LBlock or SIMON , an 80-bit register initially loaded with the master key is sequentially updated and the subkeys are extracted from that register. However, we chose to split the key into 4 smaller LFSRs that are updated in parallel because small LFSRs mix their content faster and increase performance. The downside is that each LFSR could be attacked independently if their contents were not combined back during the subkey extraction which is not the case here. Yet this design is not entirely new as a similar situation happens in the DES key schedule.

To choose the S-box that will be finally implemented, we have first implemented these 4 S-boxes separately with the goal to choose those with the smallest gate count. It appeared that all the 4 Sboxes have around the same area (around 23 GEs1). So, we have implemented LILLIPUT with PRESENT block ciphers s box as explained above .

VII. EXPERIMENTAL RESULTS

In order to simulate the encryption and decryption processes we have taken the Verilog programming environment. The Model sim softwares could run the Verilog program for the encryprion with modified sboxand simulatedouput that gives the LILLIPUT cipher is shown in fig 7. The decrypted output that gives the plain text back is shown in fig 8.

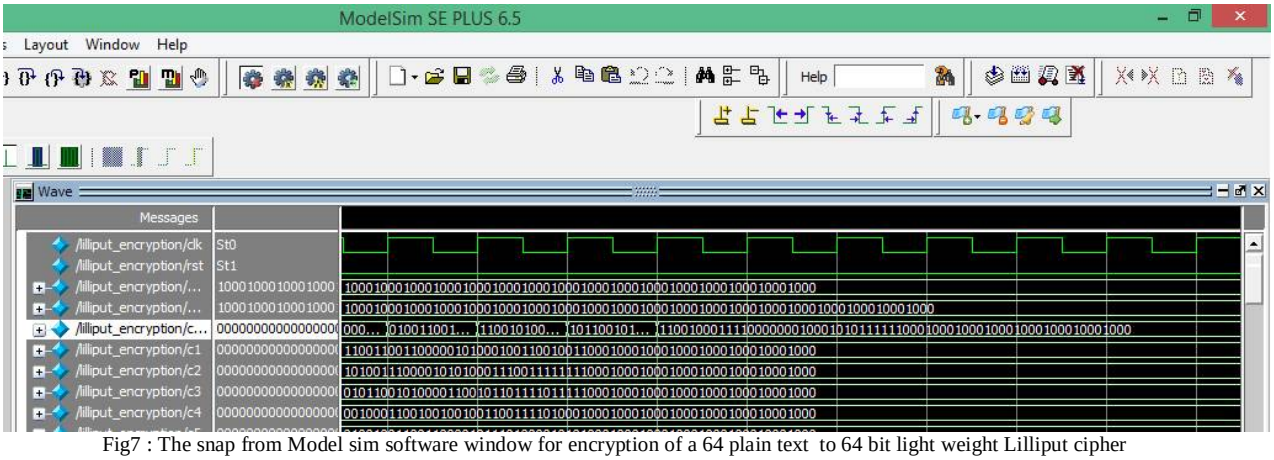

Fig7 : The snap from Model sim software window for encryption of a 64 plain text to 64 bit light weight Lilliput cipher

The plain text given isa 64 bit binary
number :1000100010001000100010001000100010001000100010001000100010001000
The corresponding cipher text is produced as shown in the snap.

ISSN(Online) : 2319-8753
ISSN (Print) : 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

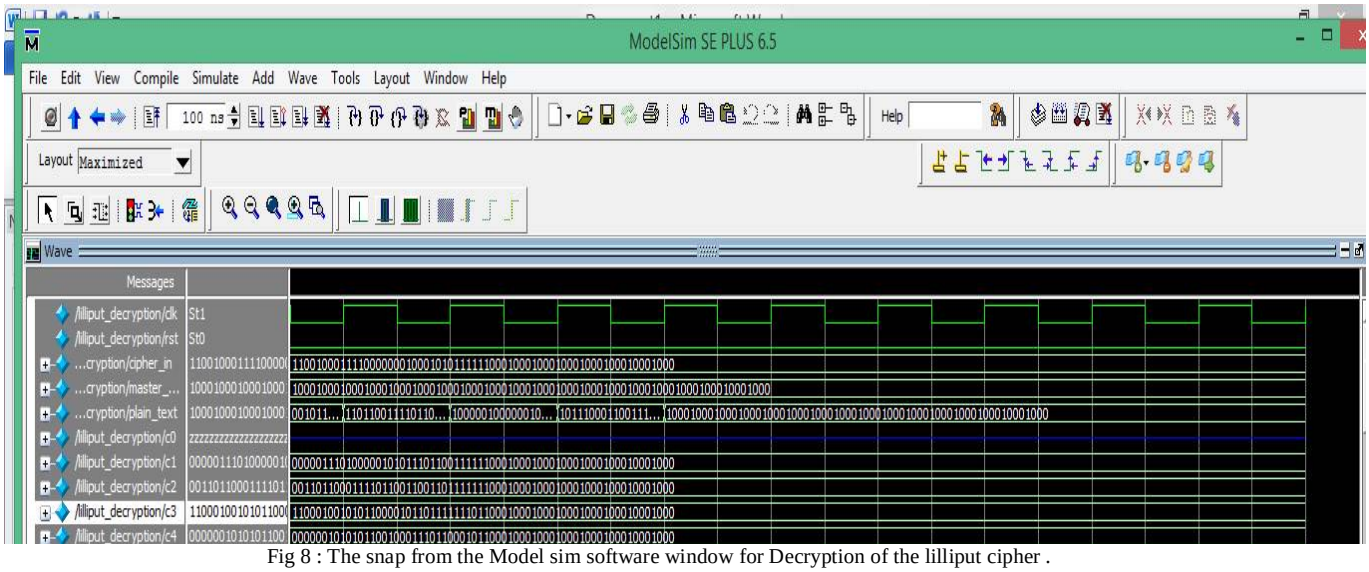

Fig 8 : The snap from the Model sim software window for Decryption of the lilliput cipher .

VIII. CONCLUSION

This topic encourage all the cryptographic community to perform independent security analysis on LILLIPUT.Along with the modification in the substitution layer the speed of operation can be increased. The light weight cryptography became moreusefull by using the methods like explained here.

REFERENCES

[1] T. Berger, M. Minier, and G. Thomas, "Extended Generalized Feistel Networks using Matrix Representation," in Selected Areas in Cryptography - SAC 2013, ser. LNCS, vol. 8282. Springer, 2013, pp. 289–305.
[2] Data Encryption Standard, National Bureau of Standards, U. S. Department of Commerce, 1977.
[3] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita, "Camellia: A 128-Bit Block Cipher Suitable for Multiple Platforms - Design and Analysis," in Selected Areas in Cryptography - SAC 2000, ser. LNCS, vol. 2012. Springer, 2000, pp. 39–56.
[4] C. Adams and J. Gilchrist, "The CAST-256 Encryption Algorithm," Network Working Group, RFC 2612, june 1999, http://tools.ietf.org/html/rfc2612.
[5] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee, "HIGHT: A New Block Cipher Suitable for Low-Resource Device," in Cryptographic Hardware and Embedded Systems - CHES 2006, ser. LNCS, vol. 4249. Springer, 2006, pp. 46–59.
[6] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-Bit Blockcipher CLEFIA (Extended Abstract)," in Fast Software Encryption - FSE 2007, ser. LNCS, vol. 4593. Springer, 2007, pp. 181–195.
[7] K. Nyberg, "Generalized Feistel Networks," in Advances in Cryptology - ASIACRYPT '96, ser. LNCS, vol. 1163. Springer, 1996, pp. 91–104.
[8] SHS, "Secure Hash Standard," in FIPS PUB 180-4, Federal Information Processing Standards Publication, 2012.
[9] C. Burwick, D. Coppersmith, E. D'Avignon, R. Gennaro, S. Halevi, C. Jutla, S. M. M. Jr, L. O'Connor, M. Peyravian, D. Stafford, and N. Zunic, "MARS - an AES candidate," NIST AES Proposal, 1999.
[10] Y. Zheng, T. Matsumoto, and H. Imai, "On the Construction of Block Ciphers Provably Secure and Not Relying on Any Unproved Hypotheses," in Advances in Cryptology - CRYPTO '89, ser. LNCS, vol. 435. Springer, 1989, pp. 461–480.