International Journal of communication and computer Technologies, ISSN: 2278-9723

Available at http://www.ijccts.org

New Encryption technique for improving time and speed for Embedded Applications

1E.Kavitha,2B.Saranya,3V.Nathiya,4K.Gomathi,5A.Surendar
1,2,3,4 UG students, Department of Electronics and Communication Engineering,
K.S.R. College of Engineering
5 Assistant Professor, Department of Electronics and Communication Engineering,
K.S.R. College of Engineering

Received: 26-06-2014, **Revised:** 03-08-2014, **Accepted:** 28-10-2014, **Published online:** 24-12-2014

.

Abstract

Lightweight cryptography is an interesting fieldthat strikes the perfect balance in providing security, higher throughput, low-power consumption, and compactness. In recent years, many compact algorithms like PRESENT, CLEFIA, SEA, TEA, LED, ZORRO, Hummingbird, and KANTAN have made the mark to be used as lightweight crypto engines. In this paper, we present the design of a new lightweight compact encryption system based on bit permutation instruction group operation (GRP), which is widely studied and extensively researched. Using the S-box of PRESENT, we have added the confusion property for GRP, because all the existing algorithms using bit permutation instructions do not have this confusion property. By comparing the existing S-boxes of compact algorithms and its cryptanalysis, a new hybrid system is proposed in this paper that provides more compact results in terms of both memory space and gate equivalents. A hybrid cryptosystem, which consists of GRP and S-box of PRESENT, is designed and implemented on a 32-bit processor. This fusion has resulted in a lightweight cipher that is the most compact implementation, till now, in terms of memory requirement. We have tested and verified this on an LPC2129 processor. Various S-boxes of recently used lightweight algorithms, such as PRESENT and CLEFIA, are designed and analyzed to create a perfect fusion that should be resistant to attacks. Using the S-box of PRESENT, it helps in further reducing the gate complexity. This hybrid model results in 2125 gate equivalents, which is better than other light variant models like DESXL, CLEFIA, and AES. Moreover, GRP properties are very helpful not only to attain the desired avalanche effect, but also as it results in a compact implementation in hardware. This paper proposes a novel approach that will have a positive impact in the field of lightweight encryption protocols.

Index Terms— Lightweight cryptography, PRESENT, GRP,embedded security, encryption, bit permutation.

I. INTRODUCTION

The increasing use of pervasive devices in the fieldof electronics has raised the concerns about security. In embedded applications, implementing a full-fledged cryp-tographic environment would not be practical because of the constraints like power dissipation, area and cost. Due to these constraints, the focus is on using lightweight cryptography that needs as less memory space as possible. The main criterion for the lightweight cipher is to have less memory space and that which would result into a less Gate Equiv-alent (GEs) count for an efficient hardware implementation without compromising the requirement of strong security properties. An ISO/IEC standard on lightweight cryptography requires that the design be made with 1000–2000 gate equiv-alents (GEs) [1]. RFID tags may have 1000–10000 GEs out of that only 300–2100 GEs would be available for security aspects [2]. Many algorithms have been designed in the past few years and implemented in the field of pervasive comput-ing. For security applications, total GEs available would be approx 2000–3000. Block ciphers should be limited to less GEs in order to fit in lightweight applications. Ciphers like AES [3], DES [4], [5] would result in high GEs that make them infeasible for small scale real time applications. Light variants of DES such as DESL [6], DESXL [7] have been proposed by slightly modifying the algorithms, by reducing the S-boxes and by using key whitening to increase security levels. These lightweight versions consume around 2200 GEs for encryption. Alternative to this approach of modifying an existing block cipher and to have an efficient hardware model, is an entirely new structure that has been designed called “PRESENT.” PRESENT is a Substitution Permutation network based on 80 bit or 128 bit key size and 64 bit block

size. PRESENT [8] is a block cipher with 31 rounds and its various variants need 2520 to 3010 GEs to provide adequate security levels. PRESENT [8] is also available for ultra small applications with 1000 GEs. PRESENT is one of the leanest lightweight algorithms designed and one that has obtained the ISO/IEC standard for lightweight cryptography. CLEFIA [9], [10] is one more compact algorithm developed by SONY to achieve less GEs and has two confusion and two diffusion properties that results in a higher memory requirement. For small devices like RFID, few lightweight cryptographic algorithms, mostly block ciphers, have been published like HIGHT [11], mCrypton [12], SEA [13], TEA [14] and ICEBERG [15]; and these are summarized in Table I with respect to their GEs. Most of these algorithms result in more than 2300 GEs which is the major limitation for their use in ultra small applications.

In the cryptographic environment, there are two types of instructions, one is the "SP-network" (Substitution Permu-tation network) like AES, PRESENT, etc. and other is the "Feistel network" like TEA, XTEA, etc. In this paper, we have focused on SP-network only, as they provide good resistance against most of the attacks. Stream ciphers are also widely studied in the cryptographic environment because of its faster execution, but they are susceptible to attacks compared to SP network block ciphers. CLEFIA [9], [10] is a generalized Feistel structure with a substitution box. The disadvantage of generalized Feistel

COMPARISON OF LIGHTWEIGHTED ALGORITHMS

TABLE I

Lightweight Algorithm	Block Size	Key Length	GEs
HIGHT	64	128	3048
mCrypton	64	64 96 128	2420 2681 3758
SEA	96	96	3758
TEA	64	128	2355
ICEBERG	64	128	7732
CLEFIA	128	128	2488
PRESENT	64	128	1884

COMPARISON OF GRP WITH TABLE LOOKUP

TABLE II

	Table LOOK UP	GRP
Maximum no. of instructions for 64 bit word size permutation	23	6
Maximum no. of instructions for 128 bit word size Permutation	47	7
No. of cycles for scheduling of permutation instructions on Super Scalar processor	16	8
Maximum number of instructions for permuting 64 bit with control bits	31	13
Memory requirement for 64 bit permutation	16 kb	48 bytes

rounds to make the cipher secure, as the input block size is divided in greater than or equal to three sub blocks. We therefore have implemented SP network block ciphers and ciphers that are based on Generalized Feistel network like CLEFIA on the same platform that is LPC2129, so that comparison would be easy with our design of PRESENT-GRP.

The aim of this paper is to describe the results of a design of a compact cipher with adequate security for applications like pervasive computing. Our research work focuses on a compact hardware implementation of the cipher. Bit permuta-tion instructions are efficient in such kind of implementations and they are complex in nature which gives them an edge in cryptographic environment. Bit permutations are popularly known to be used in permutation block known as diffusion property. We have carefully studied the properties and secu-rity aspects of bit permutation instructions like GRP [16] and OMFLIP [17]. Among all bit permutation instruction GRP proved to be an efficient instruction in terms of cryptographic properties, memory size and total number of gate counts. Bit permutation instructions are widely studied and currently supported by all word oriented processors. GRP is an extensively researched instruction set, its crypt-analysis is well known, and many attacks have been tried in the past on bit permutation instructions. GRP is known for fast bit permutation [16]. Alternative to bit permutation is a simple look up table, but permutation operations are far better than a look up table. A Look up table typically requires nearly 23 numbers of instructions for a 64 bit permutation while GRP does the permutation in only six lines of instructions which is mentioned in Table I. Lookup tables have large footprint area thus increases GEs. GRP is complex in nature that makes it more suitable for cryptographic environment as compared to operations like shifting, multiply or addition. GRP is suitable specifically for encryption in an application like remote sensor continuously encrypting data and sending it to a server location [16]. Moreover, GRP has good differential properties because the paths of data bits totally depends on control bits applied to the structure. Change of even a single control bit will cause all the data bits to change at the output [16]. This property helps to achieve desired avalanche effect and makes design more robust against attacks. Algo-rithms like DES [4], [5], SERPENT [18] and TWO FISH [19] use bit permutation instructions in their operations which helps to resist against linear and differential cryptanalysis. Bit \permutation instructions lack confusion property that is an S-box. According to Shannon having only the diffusion prop-erty is not sufficient to provide a secure cipher [20]. GRP uses subword permutation that not only does permutation efficiently but also accelerates the software cryptography [16].

We have implemented lightweight algorithms on LPC2129 (ARM 7) processor because it has many features which are suitable for small scale embedded applications.

II. GRP: A BIT PERMUTATION INSTRUCTIONS

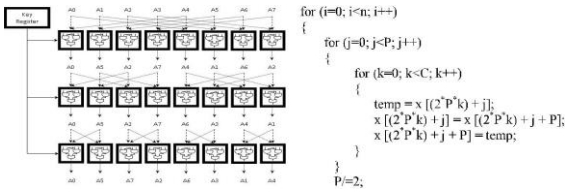
As described earlier GRP [16] is one of the most complicated bit permutation instructions that make it an obvious choice to be used in cryptographic environment. GRP performs n bit permutation with $\log_2(n)$ steps while other instructions take O(n) steps [16]. Research in this field and papers [18], [22] have shown the increased strength of cipher RC5 by introducing GRP instructions. GRP scales very efficiently to 2n bits on n bit system by using instruction Shift right pair instruction (SHRP) in PA, RISC and in IA-64 processors [23], [24]. Table look up is an alternative to bit permutation instructions, but it is much slower as it takes 16 cycles on a superscalar processor for the scheduling of permutation instructions, while GRP does it in only 8 cycles. By loading control bits, GRP requires 13 numbers of instruc-tions while a table lookup needs 31 numbers of instructions. Table II shows the comparison of GRP with Table lookup on various parameters that shows the edge GRP has over a table look up, in terms of memory space and execution, which are the most important aspects in lightweight cryptography.

Paper [16] shows key generation by GRP achieving tremen-dous speed because most of the permutation instructions exist in this block.

Previous Method

III. GRP IMPLEMENTATION

Papers [16], [25] shows various implementations of GRP at hardware and at software levels. In this paper, we have achieved implementation of 128 bit key generation and encryp-tion. The algorithm is explained in paper [16].



GRP algorithm can generate the different keys at different rounds from a given integer sequence. Key generation for respective integer sequence is illustrated with an example in paper [16]. GRP 128 bit permutation is designed by us, and implemented on 32 bit processor LPC2129, where the user has to give an 128 bit input and GRP performs the exact same operations for 128 bits which is performed for an 8 bit encryption in Fig. 1. It is a universal design which generates codeword's for n integers. Fig. 1 indicates the basic GRP encryption operations in terms of AND, OR and NOT gates. In Fig. 1, key register generates the key according to GRP algorithm [16], that is based on the user defined integer sequence and that key is applied as codeword to each of the permutations to do the encryption. In this paper, 128 bit encryption is designed and performed based on bit permuta-tion instructions. The algorithm and the steps involved while deigning permutation box by using GRP are outlined in Fig. 1.

In a scenario, let us assume that the input is a plain text with a bit length w = 128 that needs to be permuted with the help of GRP. To permute 128 bits, the operation needs total 7 stages as 2^7= 128 bits. 7 stages means GRP 128 will perform up to 7 rounds as 2^7= 128. Similarly, for 64 bit and 8 bit permutations, we need total of 6 and 3 stages, respectively. Fig. 1 indicates 8 bit permutation. w indicates word length and n indicates number of stages, where 2^n= w. P = w/2 indicates pairing bits in which if word length is 128 bits then P value will be 64 which depicts in the first stage of permuting 128 bits, first group is the 0^{th} bit and 64^{th} bit second will be 1^{st} and 65^{th} bit, third group will be 2^{nd} and 66^{th} bit and similarly last will be 63^{rd} and 127^{th} bit. C represents combination of pairing bits. For example, for P = 64, C value will be 1, for second stage where P = 32, C value will be 2 and for last stage P = 1, C value will be 64 which means we will be having 64 combinations of single pair. Fig. 2 represents block diagram for 128 bit encryption by using GRP. Algorithm for encrypting 128 bits by using above mentioned variables is mentioned in Fig. 2.

IV. LIGHTWEIGHT CRYPTOGRAPHY

Internet of things (IOT) is one of the most discussed topics today in the digital world. Focus area of researchers is to implement lightweight design to avoid high power dissipation

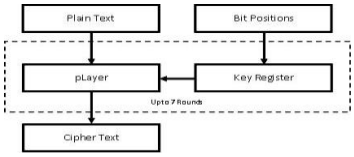


Fig. 2. Block diagram for 128 bit GRP Permutation with key generation.

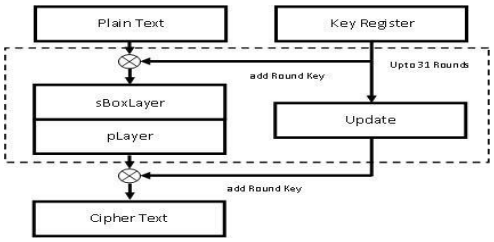
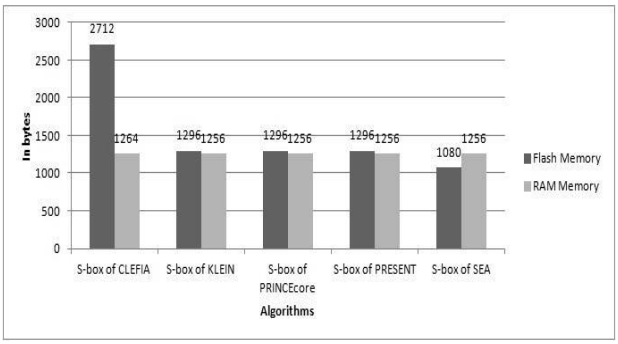
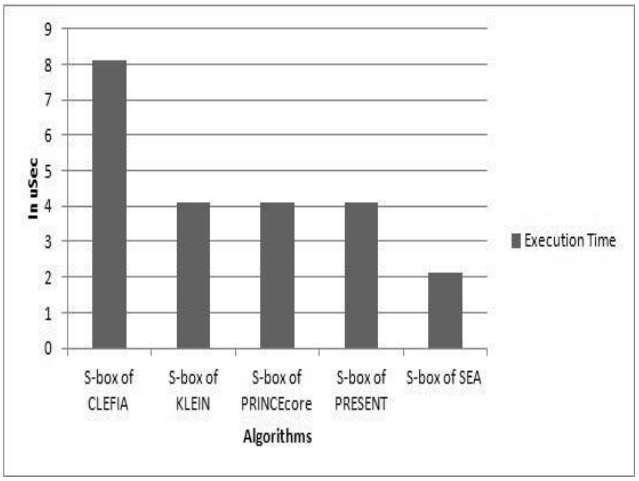


Fig. 3. Block diagram of PRESENT.

and large memory requirement. RFID tag is one of the fastest growing technologies that would be useful in IOT [1], [2]. To provide a security at RFID level, there is need to have a





lightweight cryptoalgorithm whose coverage area would be nearly 2100 GE. The standard algorithm like AES [3], DES [4], [5] have huge memory requirement and would not be feasible to be implemented for embedded system design. Many lightweight algorithms have been designed in the past and various attacks have been proven on them. Two of algorithms that got adopted as the ISO/IEC (29192-2P:2012) standards for lightweight cryptography are PRESENT [8] and CLEFIA [9], [10]. In this paper, that is the reason why, we have discussed and implemented PRESENT and CLEFIA rather than TEA, SEA, XTEA, HIGHT. Various implemen-tations of PRESENT and CLEFIA have been designed in the past to be fitted in a small area, which means with a very less gate count and less memory requirements. In this

TABLE III

Algorithm	Key Size	E/D	Memory Size (in Bytes)	
			Flash Memory	RAM Memory
PRESENT	128 bit	E	3200	1320
	128 bit	D	3252	1384
CLEFIA	128 bit	E	4708	1256
	128 bit	D	4880	1256

PRESENT AND CLEFIA MEMORY REQUIREMENT

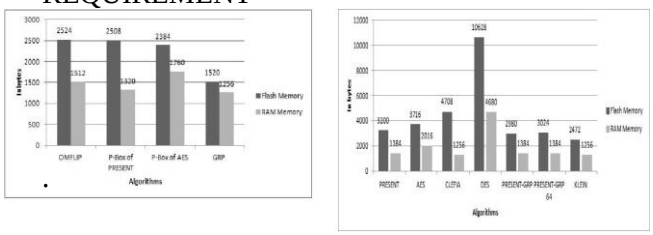


Fig. 4.S-box comparison over memory requirement.

paper, we too have implemented a structure of PRESENT and CLEFIA on LPC2129 to figure out memory requirements. PRESENT [8] and CLEFIA [9], [10] have shown resistance to different attacks when we analyzed differential and linear cryptanalysis of them, from various papers [8]–[10]. These lightweight ciphers can perform encryption well with adequate security levels of key size of at least 80 bits. Fig. 3 shows block diagram for PRESENT.

In this paper, the algorithm of PRESENT with 64 bit block size and key size of 128 bits is coded in embedded C with Keil 4.0 simulator and results are verified and tested on 32 bit processor. PRESENT structure is Substitution and Permutation (SP) design which consists of S and P layer and consists of 31 rounds. In this paper, separate analysis is done for S-box as well as for P-box. For this, we referred papers

[8], [26], S-box of PRESENT is resistant to brute force attack and has shown good resistance against linear and differential cryptanalysis. Similarly, CLEFIA is also designed and implemented on 32 bit processor (LPC2129) which is based on Feistel network. Table III shows memory requirement for PRESENT and CLEFIA. This result was verified on 32 a bit processor by using KEIL 4.0 simulator. E denotes encryption and D denotes decryption.

It is clearly evident from Table III that CLEFIA has higher memory requirement than PRESENT. CLEFIA consists of two confusion and two diffusion properties while PRESENT has only one confusion property and one diffusion property.

V. THE IMPLEMENTATION OF EXISTING

CIPHERS ON A PROCESSOR

In this paper, separate S-boxes of PRESENT and CLEFIA are designed and compared to study the lightweight model. Fig. 4 shows comparison of S-boxes of PRESENT, CLEFIA and other lightweight ciphers on LPC2129 (32 bit processor) in terms of memory requirement. It is clearly seen from figure 4, that the S-box of PRESENT is quite compact in nature as compared to other lightweight ciphers excluding SEA. Flash memory needed for S-box of PRESENT is nearly half of the memory requirement of S-box of CLEFIA. SEA emerges as a most compact cipher due to use of 3 bit S-box. SEA does the small encryption routine efficiently with a limited throughput and it requires higher number of GEs overall which is shown in Fig. 10. SEA is based on Feistel structure.

Fig. 5 shows execution time of S-boxes of algorithms like PRESENT [8], CLEFIA [9], [10], KLEIN [27], PRINCEcore [28] and SEA [13]. These all algorithms are implemented on a 32 bit processor in order to compare different S-boxes and their execution times. We studied and implemented them individually and have made a comparison to understand their relative compactness over memory space and execution time.

VI. NEW HYBRID CRYPTOSTRUCTURE,

IMPLEMENTATION AND COMPARISON

After studying carefully properties of bit permutation instructions like GRP [16], OMFLIP [17], CROSS [29], we found that GRP has an edge over all other bit permuta-tion instructions and is useful in accelerating cryptographic implementations. GRP does the fast bit permutation with less memory space and its complex structure makes it an attractive choice in cryptographic algorithms over instructions like multiply and rotate [16]. The disadvantage of bit permu-tation instruction is that it lacks an S-box, which is the most essential ingredient in designing secure block ciphers. This element directed our attention towards lightweight crypto-graphic algorithms and we carried out many experiments and studied the properties of algorithms like PRESENT [8], CLEFIA [9], [10], DES [4], [5], AES [3], SEA [13], TEA [14] and KANTAN [26], [30], [31].

\

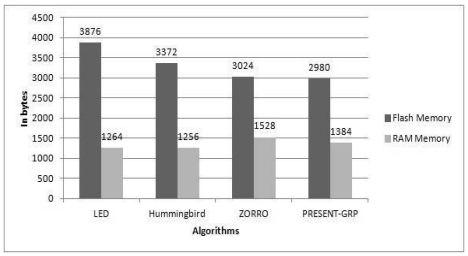


Fig. 7.S-box of PRESENT with P-box of lightweight algorithms.

As stated the aim of this paper is to present the design of a compact cipher with adequate security for applications like pervasive computing. In this paper, we present results of our research work which focuses on the compact hardware imple-mentation of a cipher. Bit permutation instructions are efficient in such types of implementations. Bit permutation instructions are complex in nature, and that gives them an edge in the cryptographic environment. The well known cipher like DES is also implemented with the help of bit permutation instructions but falls prey to attacks because of short key lengths. Among all bit permutation instructions, GRP proved to be an efficient instruction in terms of cryptographic properties, memory size and total number of gate counts. OMFLIP has poor differential properties and its structure is easily susceptible to attacks. Bit permutation instructions are widely studied and currently supported by all word oriented processors. We found com-pactness and mapping interface of GRP with PRESENT and CLEFIA. In this paper, we present the results of implementing most of the standard algorithms in order to identify their mem-ory requirements, gate equivalents and power consumption. All standard algorithms are implemented and compared on the same platform which is LPC2129 a 32 bit processor by NXP (Philips).

We have implemented AES 128 bit, GRP for 128 bit and 64 bit, PRESENT for 64 bit, CLEFIA for 128 bit and DES for 128 bit block size with different key combinations of 64, 128

Fig. 6.P-box comparison of standard algorithms. and Graphical representation of standard algorithms implemented on LPC2129.

and 80 bits. Fig. 6 indicates memory requirement of P-box of AES [3] and PRESENT [8] with GRP [16] and OMFLIP [17] computed based on KEIL 4.0 simulator and LPC2129.

In Fig. 6, OMFLIP and GRP are themselves P-boxes. They are compared with the standard algorithms. Results from Fig. 6

clearly show GRP to be more compact in nature in terms of memory requirements.

Fig. 7 shows merger of S-box of PRESENT with OMFLIP, AES and GRP. Results clearly show that PRESENT-GRP has much less memory requirement as compared to other hybrid systems. LED [32] is the latest cipher which is combination of PRESENT-AES which is also SP-network. It has 64 bit block size and 128 bit key length. LED is faster than PRESENT in software but it is slower in hardware. Moreover, LED consumes high energy per bit in embedded applications which dissipates and consumes more power.

Fig. 8 shows graphical representation of the memory size requirements of standard algorithms on a 32 bit processor. The hybrid module structure PRESENT-GRP needs less memory as compared to other standard algorithms. In this hybrid module, we have designed the permutation box (P-Box) by using GRP for 128 and 64 bit block size. This result shows the compactness of hybrid cryptosystem which is the com-bination of PRESENT-GRP. 2980 bytes of flash memory is needed for PRESENT-GRP while the PRESENT alone which is lightweight compared to other structure needs 3200 bytes of memory. CLEFIA also has higher memory requirements which consume nearly 4708 bytes. PRESENT-GRP in Fig. 8 indicates 128 bit block size and 128 bit key generation while PRESENT-GRP 64 indicates 64 bit block size and 128 bit key. KLEIN [27] algorithm is also implemented on 32 bit processor and its comparison is also shown in Fig. 8.

VII. PRESENT-GRP: A NEW HYBRID LIGHTWEIGHT DESIGN

In this paper, a hybrid cryptosystem is design is discussed, which combines the S-box of PRESENT and the P-box of GRP, to produce a compact secure structure with adequate security. This fusion structure has both the properties of PRESENT-GRP and results into a tinier version than the origi-nal algorithm PRESENT. Cryptanalysis for both the structures is studied and analyzed properly and it shows good resistance to linear and differential attacks. The same PRESENT-GRP design is compared with various standard algorithms like PRESENT [8], AES [3], DES [5] and CLEFIA [9], [10]. All these algorithms were implemented in embedded C on the same platform. This was done so that we do not

TABLE IV
OPTIMIZED GRP DESIGN

Memory Size of Old GRP 128	Memory Size of Optimized GRP128
3224 bytes	2944 bytes

any kind of artifacts due to the platform issues, either in the results or in the comparisons. Results of all algorithms were tested and verified through 8 bit UART module of ARM7. An 'UART' module was used in the design to merely act as a demonstrator in

order to verify and to be able to see the encryption/decryption outputs. The baud rate for this application was set at 9600 bps.

In PRESENT-GRP module, 64 bit/128 bit blocks were passed through the S-box of PRESENT and after mapping according to PRESENT, the output was passed to the permutation layer which performed encryption based on GRP algorithms. Keys at each stage were applied based on key generation method of GRP. In GRP key generation, inputs will be the bit positions given by user, and based on that GRP generates a sequence of 0's and 1's which serve as key to the encryption and decryption process. GRP has very robust mechanism of key generation which is the necessity in

cryptographic environment. GRP does the key generation as well as encryption with fast bit permutations.

We have designed an optimized version of GRP by doing small changes at the algorithmic level. The GRP design presented in previous papers [25], [33] need 3224 bytes of FLASH memory. In this paper, a compressed version of GRP is reported, a design that consumes only 3088 bytes of memory. This optimized structure has been achieved by reducing many arrays to just two arrays. We have designed an entirely new logic which supports very less arrays and works very fast as compared to existing structure. Execution time for our design is nearly half of the old GRP design at software level. Few observations we made during optimizing code which are listed as follows:

- Using character data type instead of integer data type
- Reduced instructions i.e. making one instruction that can perform three operations, instead of using three different instructions for different operations
- Using local variable instead of global variable, global variable consumes more space
- Using minimum functions
- Making binary to hex/hex to binary functions instead of using "Power" functions (defined)
- Using minimum variables as Recursive
- Making complex logic for long processes
- Passing only one data type into the function instead of multiple. For example, GRP (a)

All these steps help to achieve a more optimized structure of GRP which results in 1789 GEs for 64 bit permutation. Table IV shows memory requirements for past implementations of GRP and optimized GRP design. To the best of our knowledge, this is the most optimized design for GRP 128 bit.

Fig. 9. New recent lightweight algorithms implemented and compared with PRESENT-GRP on LPC2129.

TABLE V

GATE COUNT OF UMCL18G212T3 LIBRARY

Standard Cell	Process	Library	Cell Name	GE
NOT	0.18µm	UMCL18 G212T3	HDINVBD1	*0.67*
AND	0.18µm	UMCL18 G212T3	HDAND2D1	*1.33*
OR	0.18µm	UMCL18 G212T3	HDOR2D1	*1.33*
MUX	0.18µm	UMCL18 G212T3	HDMUX2D1	*2.33*

S-box of PRESENT is a very compact design that uses a 4 × 4 box in order to reduce gate the complexity and the power consumption. Larger S-boxes result into high GEs that is seen in case of DES [4], [5]. The higher bit S-box has more Boolean equations resulting in a high gate count. Moreover, implementation of the S-box of PRESENT has low hardware cost and less GEs.

Fig.9 shows comparison of PRESENT-GRP with the latest lightweight ciphers. LED [32] and ZORRO [34] are recent lightweight ciphers. All ciphers in Fig. 9 are SP-network. ZORRO is similar to the AES algorithm. ZORRO has 8 bit S-box while others have 4 bit S-box. It has 128 bit block size and 128 bit key. As ZORRO is having 8 bit S-box, it needs higher GEs compared to other ciphers using 4 bit S-box. ZORRO

needs 24 rounds to secure its struc-ture which results in higher memory requirement as shown in Fig. 9. Hummingbird [35] has 128 bit block size and 896 bit key length. It is a combination of block cipher and stream cipher. Hummingbird requires a long initialization process as compared to block ciphers like PRESENT which introduces latency and higher execution time. Hummingbird has less encryption speed and its authentication mechanism is less efficient.

A 4 bit to 4 bit S box of PRESENT needs 21 to 28 GEs when it is implemented with UMCL18G212T3 library. Table V indicates the gate count of standard cell of UMCL18G212T3 library [36].
For the 64 bit operation in our design, we have used 16 four bit

RFID tags. Total no of GE's available in pervasive design for security purpose would range from 200–2100 GEs. Based on the GE limit, our hybrid design is making a mark in that the GE range is consuming less memory as compared to all the standard lightweight and other cryptographic algorithms at the software level. Hybrid structure, if mapped properly, always has an edge over other structures as the structure carries good and robust property of individual design. Paper [38] suggests combination of GRP and DDR (Data Dependent Rotation) may show good resistance against differential and linear cryptanalysis which is later proved by combining RC5 and GRP [39].

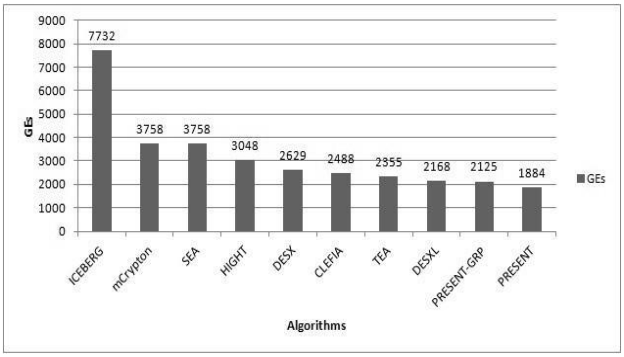
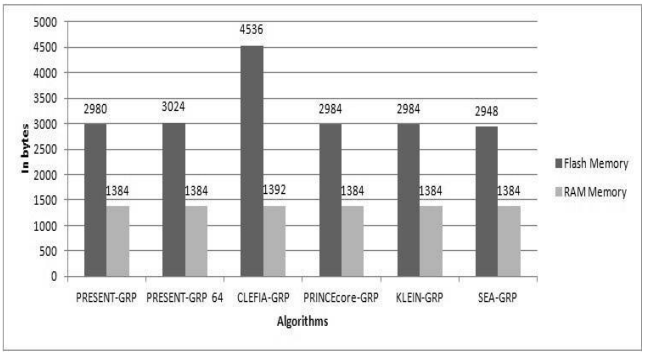


Fig. 11.Hybrid Crypto structure comparison chart.



GRP for permutation. By combining PRESENT-GRP, the total gate count needed are 2125 GEs by considering S-box gate count as 21 [8], [37]. PRESENT's S-box consumes nearly 21 GEs for a single 4 bit S-box. PRESENT-GRP gate counts results in 2125 which is less than other algorithms except for PRESENT which is implemented based on the 'round based' architecture and whose gate count is 1884 [8], [37].

GEs for other algorithmic design like CLEFIA [9], [10], SEA [13], TEA [14], HIGHT [11], mCRYPTON [12], ICEBERG [15], DESX and DESXL [7] based on the specific library functions, is computed and a comparison is made which is illustrated in Fig. 10.

In this paper, we have also designed and implemented CLEFIA-GRP, where S-box of CLEFIA is used and GRP as permutation layer. CLEFIA-GRP is compared with PRESENT-GRP to understand compactness of this hybrid structures. Fig. 11 represents hybrid implementations and their comparisons with standard lightweight design like PRESENT and CLEFIA. In Fig. 11, PRESENT-GRP and PRESENT-GRP 64 are compared with other lightweight standard algorithms like SEA-GRP, PRINCEcore-GRP, KLEIN-GRP and CLEFIA-GRP. The hybrid structure of PRESENT-GRP has very less memory requirement as compared to the other algorithms. SEA-GRP algorithm results in less memory size as compared to PRESENT-GRP as it has 3 bit of S-Box while others have 4 bit of S-Box. CLEFIA-GRP fusion results in 4536 bytes of flash

memory requirements which depicts a heavy cryptographic design and may not be suitable for lightweight applications like

VIII. SECURITY ANALYSIS

One of the most important aspects in designing a block cipher is to do cryptanalysis on it. Cryptanalysis helps us to know whether the plain text can be derived from the cipher text or if some of the bits in the key can be identified. The most popular attack is brute force attack which decrypts the text by using all possible keys. Risk of this attack can be reduced by increasing the key size, as then the possible combinations will be more and therefore difficult to compute. But, these attacks need a lot of time to compute robustness. Two properties or attacks which are of utmost importance are differential cryptanalysis and linear cryptanalysis. Differential cryptanalysis works on properties of a round function. For example, consider O =A operation B, where A denotes the bits, B denotes how to permute bits and O denotes permuted bits and operation indicated GRP operation in this paper. In differential cryptanalysis, the two plain texts are selected with some difference _D and it is measured by 'exclusive OR' operation, while two plain texts will be converted to cipher texts with their difference as _E. The pair (_D, _E) is known to be differential characteristics, where _E should have a larger value than the average probability [16]. In linear cryptanalysis, we will find a relation between certain bits of plain text and the key that has probability p _= 1/2 called linear approximation. Based on these two properties GRP is analyzed and evaluated. Table VI denotes the differential characteristic of GRP [16].

e_s represents n bit word that is zero except for a single one in bit position s. Three types of permutation operation shown in Table VI are Type A, B and C. Type A denotes how the single bit s is moved when the control bits are randomly chosen and the probability determines how closely s is moved to bit t in Z. For Type B and C, diffusion effect is compared by computing the hamming weight of output difference _ [16]. A larger hamming weight represents avalanche effect which shows good differential properties. Type A, B and C property of GRP shows that a small change in input results into large changes at the output; the so called avalanche effect. E(|_|) represents expected value of difference when the input sequence is random. As shown in Table VI, from Type A property, the probability of GRP is totally depends on s and t which has probability between 0 and 0.5. Type B and C property of GRP bring about n/4 different output bits for any t. For Type A characteristics, p is always $1/2^n$ for any value of s and t which is much smaller and can be neglected. For Type B characteristics which are based on hamming weight, the approximate value of _O is n/4. For Type C, a single bit difference in A does not result in too much deviation. With these characteristics, GRP achieves the necessary avalanche effect which makes the GRP design robust against differential attacks. GRP shows good differential properties better than other bit permutation instructions as data bits path is totally dependent on control bits. Table VII shows linear cryptanalysis of GRP permutation instruction [16].

TABLE VII

LINEAR APPROXIMATION OF GRP PERMUTATIONS

Operation	Type L (e_s, 0, e_t)	Type M (e_s, e_u, e_t)
GRP	$b \le 1/4 + 1/2^{n+1}$ Maximum with s = t = 0	$b \le 1/4 - 1/2^{n+1}$ Maximum with s = u = t = 0

For linear approximation, the bias is 0.5 since p is equal to 1. Linear approximation is always in the form of triplets with the probability p. In linear approximation, Type L and Type M compare the permutation instructions, based on the control bits in B. Type L and Type M compare the permutation instructions based on any control bits Y are indulged in approximation. b in given Table VII represent bias. Type L shows how closely the permutation moves the bit around and the probability of moving of As to O_t is p, then the probability of A_s=O_t is 0.5 + p/2. So the bias for triplet (e_s,0,e_t) is p/2. For GRP, the maximum bias is achieved when s = t = 0. Type M determines how important the control bit is to determine path of X to Z. The maximum bias for Type M is achieved at s = t = u = 0. For GRP operation, the bias of linear approximation is
$1/4 + 1/2^{n+1}$. For n-bit GRP operations, the bias of linear approximation p(A B = O) is $1/4 - 1/2^{n+1}$. Table VII
depicts GRP has some linear approximation large

with
be considered for
bias, but these can lightweighted cryp-
tography where standard algorithm AES costs
like more
power
in terms of memory requirement and consump-
tion [3]. Research papers [16], [18] shows that by inducing GRP in standard block ciphers like RC5 will tremendously increase the efficiency of a cipher with same security levels. Papers [13], [18], [40] shows RC5 performance with GRP which suggest the importance of GRP in lightweighted crypto-graphic applications. RC5-GRP uses only 6 rounds to achieve THE desire security level.

PRESENT has an edge over CLEFIA, in that it contains a linear bitwise permutation and non linear substitution layer. A non linear S-box uses 4 bit structure which yields into less GE and less power consumption. Extra properties in S-box help PRESENT to achieve the desired avalanche effect. The theorem which shows the effect of differential cryptanalysis
on S-box of PRSENT is that "Any five differential char-acteristics of PRESENT has a minimum number of ten active S-boxes" and results from papers [8], [26] shows that PRESENT has very good and compact S-box. There are 16 S-boxes of PRESENT which are divided into four groups. From papers [8], [26], the characteristics of S-box are outlined below:

1) The input bit to an S-box comes from 4 well defined S-boxes of the same group.
2) The input bits to a group of four S-boxes come from 16 different S-boxes.
3) The four output bits from a particular S-box enter into four well defined S-boxes, each of them belongs to a distinct group of S-boxes in the subsequent round.
4) The output bits of S-boxes in distinct groups will be fed to distinct S-boxes.

An advanced technique allows the cryptanalyst to remove the outer rounds from a cipher to exploit a shorter character-istic. However, if the attackers consider only 25 rounds out of 31 rounds, then the security bounds are still more than the given requirement. So, over 25 rounds of PRESENT must have at least $5 \times 10 = 50$ active S-boxes. The maximum differential probability of a PRESENT S-box is 2^{-2} and so the probability of a single 25-round differential characteristic is bounded by 2^{-100}. As far as Linear cryptanalysis of PRESENT is concerned, the interpretation mentioned is that "Let A be the maximal bias of a linear approximation of four rounds of PRESENT, then $A \leq 1/2^7$." To bound the maximal bias of a 28-round linear approximation by

$$2^6 \times A^7 = 2^6 \times (2^{-7})^7 = 2^{-43}$$

Therefore, by assumption that a cryptanalyst need only approximate 28 of the 31 rounds from linear cryptanalysis in order to mount an attack the cipher requires 2^{84} known plain text/cipher text which crosses the available text limit [8]. There are also more attacks which are proven like structural attacks, algebraic attacks, and key schedule attacks, other than linear and differential attacks. Some ciphers have strong word-like structures, where the words are typically bytes. Algebraic attacks have had better success when applied to stream ciphers than block ciphers. Algebraic attacks and key schedule attacks are unlikely to pose a threat to PRESENT. These properties of PRESENT are sufficient to resist key schedule-based attacks.

IX. CONCLUSION

Bit permutation instructions increases strength of a block cipher by allowing them to perform any arbitrary permutations efficiently with 'log(n)' steps as compared to 'n'. It performs fast bit permutation and uses subword sorter that makes the operation faster and can increase the throughput in applications like scanning an image, performing bubble sort and in the permutations layer in block ciphers. GRP generates the control words faster, that which helps in increasing the performance of many embedded systems. Block ciphers like RC5, RC6 use DDR instructions which make them vulnerable to differential attacks. This further increases the number of rounds and memory requirements. But, by replacing DDR with GRP no
only adds cryptographic strength to the cipher, but also reduces the memory requirements and the power consumption. GRPs have better differential and linear cryptanalysis proper-ties. Though they cannot completely replace DDR, but they add strength to the

block cipher by removing the weaknesses of DDR. Other ciphers like hash functions and stream ciphers may get benefited by one introducing the bit permutation instructions in them. GRP have all these good properties that provide strength in cryptographic environment. But, it lacks S-box which is necessary to provide a more secure design. This shifted our focus to find a lightweighted S-box that can be mapped onto GRP to get a secure and efficient hybrid crypto structure.

In the search for an S-box, our research got motivation from the ISO standard for lightweight cryptography and we shifted our focus to lightweight design. In this paper, we present the design with a permutation box (P-box) by using GRP for 128 and 64 bit block size. Key generation is also achieved through GRP. For designing a lightweight secure cipher, a confusion property is a must and it should be well mapped with the diffusion property. The fusion of S-box of various lightweight block ciphers and P-box of GRPs were made and were compared in terms of memory space and GEs. We have implemented latest lightweight ciphers and interfaced them with GRP. In order to achieve a very compact implementation of cipher as reported in this paper, we have carefully designed the permutation box that has resulted in a much lower gate count. We have carefully analyzed and studied linear and differential cryptanalysis of P-box of GRP and found it resistant to attacks like brute force attacks. Similarly, S-box of latest ciphers like PRESENT, CLEFIA, TEA etc, have been considered and implemented on a 32 bit processor. We already know that GRP design is hardware efficient and needs very few GEs which meets requirement of security in a lightweight cryptographic design. PRESENT is an engineered cipher whose S-box is the most compact substitution box among all the light variants and has good linear and differential properties. PRESENT's S-box results in a very compact implementation that consumes merely 21 GEs for a single 4 bit S-box. RAM and Flash memory requirements for PRESENT-GRP implementation results in very less bytes as compared to other lightweight algorithms and even with PRESENT individually.

This paper proposes a novel approach by introducing a compact hybrid system in terms of memory requirements that is best suited for lightweight cryptographic design.

ACKNOWLEDGMENT

The authors would like to thank Symbiosis Institute of Technology, Pune, Symbiosis International University, Pune and Prof. AyanMahanolobis from IISER, Pune, and eminent professionals from the automotive embedded domain, in Pune for providing suggestions and valuable inputs that helped them to carry out this research successfully. They would also like to thank Axel York Poschmann and Zhijie Jerry Shi, whose thesis and work motivated this research and also provided them with some ideas to carry on the work in the future.

REFERENCES

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications inContactless Smart Cards and Identification*. Hoboken, NJ, USA: Wiley,2003.

[2] A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols," in *Advances in Cryptology*. Berlin Germany: Springer-Verlag, 2005, pp. 293–308.

[3] National Institute of Standards and Technology (NIST). (Nov. 26, 2001). *Advanced Encryption Standard (AES)*, Federal Information Processing Standards Publication 197. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[4] National Institute of Standards and Technology (NIST). (Dec. 30, 1993). *Data Encryption Standard (DES)*, Federal Information Processing Standards Publication 46-2. [Online]. Available: http://www.umich.edu/~x509/ssleay/fip46/fip46-2.htm

[5] National Institute of Standards and Technology (NIST). (October 25, 1999). *Data Encryption Standard (DES)*, Federal Information Processing Standards Publication 46-3.[Online]. Available: http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

[6] A. Poschmann, G. Leander, K. Schramm, and C. Paar, "New lightweight crypto algorithms for RFID," in *Proc. IEEE Int. Symp. Circuits Syst.(ISCAS)*, May 2007, pp. 1843–1846.

[7] T. Eisenbarth and S. Kumar, "A survey of lightweight-cryptography implementations," *IEEE Des. Test. Comput.*, vol. 24, no. 6, pp. 522–533, Nov./Dec. 2007.

[8] A. Bogdanov*et al.*, "PRESENT—An ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science), vol. 4727, P. Paillier and I. Verbauwhede, Eds. Berlin, Germany: Springer-Verlag, 2007, pp. 450–466.

[9] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128 bit blockcipher CLEFIA," in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 4593, A. Biryukov, Ed. Berlin, Germany: Springer-Verlag, 2007, pp. 181–195.

[10] *The 128 Bit Blockcipher CLEFIA: Algorithm Specification*,Sony Corporation, Tokyo, Japan, 2007.

[11] D. Hong *et al.*, "HIGHT: A new block cipher suitable for low-resource device," in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science), vol. 4249, L. Goubin

and M. Matsui, Eds. Berlin, Germany: Springer-Verlag, 2006, pp. 46–59.

[12] L. Brown, J. Pieprzyk, and J. Seberry, "LOKI—A cryptographic primitive for authentication and secrecy applications," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 453, J. Pieprzyk and J. Seberry, Eds. Berlin, Germany: Springer-Verlag, 1990, pp. 229–236.

[13] F.-X. Standaert, G. Piret, N. Gershenfeld, and J.-J. Quisquater, "SEA: A scalable encryption algorithm for small embedded applications," in *Smart Card Research and Applications* (Lecture Notes in ComputerScience), vol. 3928, J. Domingo-Ferrer, J. Posegga, and D. Schreckling, Eds. Berlin, Germany: Springer-Verlag, 2006, pp. 222–236.

[14] D. J. Wheeler and R. M. Needham, "TEA, a tiny encryption algorithm," in *Fast Software Encryption* (Lecture Notes in Computer Science), vol. 1008, B. Preneel, Ed. Berlin, Germany: Springer-Verlag, 1994, pp. 363–366.

[15] F.-X. Standaert, G. Piret, G. Rouvroy, J.-J. Quisquater, and J.-D. Legat, "ICEBERG : An involutional cipher efficient for block encryption in reconfigurable hardware," in *Fast Software Encryption*, B. Roy and W. Meier, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 279–298.

[16] Z. Shi and R. B. Lee, "Bit permutation instructions for accelerating software cryptography," in *Proc. IEEE Int. Conf. Appl. Specific Syst.,Archit. Process. (ASAP)*, Jul. 2000, pp. 138–148.

[17] X. Yang and R. B. Lee, "Fast subword permutation instructions using omega and flip network stages," in *Proc. Int. Conf. Comput. Design*, Sep. 2000, pp. 15–22.

[18] R. Anderson, E. Biham, and L. Knudsen, "Serpent: A proposal for the advanced encryption standard," *NIST AES Proposal*, vol. 174, Jun. 1998. [Online]. Available: http://www.cl.cam.ac.uk/~rja14/serpent.html

[19] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "TwoFish: A 128-bit block cipher," *NIST AES Proposal*, vol. 15, Jun. 1998.

[20] C. E. Shannon, "Communication theory of secrecy systems," *Bell Syst.Tech. J.*, vol. 28, no. 4, pp. 656–715, 1949.

[21] A. Poschmann, "Lightweight cryptography: Cryptographic engineering for a pervasive world," Ph.D. dissertation, Faculty of Electrical Engineer-ing and Information Technology, Ruhr-University Bochum, Germany, Feb. 2009.

[22] R. L. Rivest, "The RC5 encryption algorithm," in *Fast Software Encryption*(Lecture Notes in Computer Science), vol. 1008. Berlin, Germany: Springer-Verlag, Dec. 1994, pp. 86–96.

[23] *IA-64 Application Developer's Architecture Guide*, Intel Corp.,Santa Clara, CA, USA, May 1999.

[24] R. Lee, M. Mahon, and D. Morris, "Pathlength reduction features in the PA-RISC architecture," in *37th IEEE Comput. Soc. Int. Conf., Dig.Papers*, Feb. 1992, pp. 129–135.

[25] G. Bansod, G. Aman, G. Arunika, B. Gajraj, S. Chitrangdha, and A. Harshita, "Experimental analysis and implementation of bit level permutation instructions for embedded security," *WSEAS Trans. Inf. Sci.Appl.*, vol. 10, no. 9, pp. 303–312, 2013.

[26] A. Bogdanov*et al.*, "PRESENT: An ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science), vol. 4727. Berlin, Germany: Springer-Verlag, 2007, pp. 450–466.

[27] Z. Gong, S. Nikova, and Y. W. Law, "KLEIN: A new family of lightweight block ciphers," in *RFID. Security and Privacy*, A. Juels and C. Paar, Eds. Berlin, Germany: Springer-Verlag, 2011. [Online]. Avail-able: http://www.rfid-cusp.org/rfidsec/files/RFIDSec2011DraftPapers.zip

[28] J. Borghoff*et al.*, "PRINCE—A low-latency block cipher for pervasive computing applications," in *Advances in Cryptology*. Berlin, Germany: Springer-Verlag, 2012, pp. 208–225.

[29] X. Yang, M. Vachharajani, and R. B. Lee, "Fast subwordpermuta-tion instructions based on butterfly networks," *Proc. SPIE*, vol. 3970, pp. 80–86, Jan. 2000.

[30] G. Leander, C. Paar, A. Poschmann, and K. Schramm, "A family of lightweight block ciphers based on DES suited for RFID applications," in *Proc. FSE*, vol. 4593. 2007, pp. 196–210.

[31] C. De Canniere, O. Dunkelman, and M. Knezevic, "KATAN and KTANTAN—A family of small and efficient hardware-oriented block ciphers," in *Cryptographic Hardware and Embedded Systems* (Lec-ture Notes in Computer Science), vol. 5747. Berlin, Germany: Springer-Verlag, 2009, pp. 272–288.

[32] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," in *Cryptographic Hardware and Embedded Systems* (Lecture Notes in Computer Science), vol. 6917. Berlin, Germany: Springer-Verlag, 2011, pp. 326–341.

[33] G. V. Bansod, "Audio sub word sorter unit on sorter network for sense transmission," in *Proc. IEEE Int. Conf. Central Syst., Comput. Eng.(ICCSCE)*, Penang, Malaysia, Jan. 2012, pp. 127–131.

[34] J. Guo, I. Nikolic, T. Peyrin, and L. Wang, "Cryptanalysis of Zorro," in *Proc. IACR Cryptol.ePrint Archive*, 2013, p. 713.

[35] D. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith, "Hummingbird: Ultra-lightweight cryptography for resource-constrained devices," in *Proc. 14th Int. Conf. Financial Cryptogr. Data Security*, 2010, pp. 3–18.

[36] *0.18 μm VIP Standard Cell Library Tape Out Ready, Part Number UMCL18G212T3, Process: UMC Logic 0.18 μm Generic II Technology: 0.18 μm*, Virtual Silicon Inc., Sunnyvale, CA, USA, Jul. 2004.

[37] B. S. Kaliski and Y. L. Yin, "On differential and linear cryptanalysis of RC5 encryption algorithm," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 963. Berlin, Germany: Springer-Verlag, 1995, pp. 171–184.

[38] B. S. Kaliski and Y. L. Yin, "On the security of the RC5 encryption algo-rithm," RSA Labs., Bedford, MA, USA, Tech. Rep. TR-602, Sep. 1998. [Online]. Available at www.rsa.com/rsalabs/aes

[39] S. Devadas and S. Malik, "A survey of optimization techniques targeting low power VLSI circuits," in *Proc. ACM/IEEE Conf. Design Autom.*, 1995, pp. 242–247.

[40] Z. J. Shi and R. B. Lee, "Bit permutation instructions: Architec-ture, implementation, and cryptographic properties," Ph.D. dissertation, Depart. Elect. Eng., Princeton Univ., Princeton, NJ, USA, 2004.