13-Aug-2019

Prof. Vineet Sahula
Dept. of ECE
MNIT Jaipur -302017 INDIA

Dear Sir

We gratefully acknowledge your suggestions sir, regarding CrossRef check of our manuscript, entitled- **Energy Efficient Lightweight Cryptography Algorithms for IoT Devices, f**or publication in IETE Journal of Research.

(A) This manuscript is significantly enhanced version of our earlier published paper entitled- **Lightweight Security Algorithm for Low Power IoT** Devices in 5th *International Conference on Advances in Computing, Communications and Informatics (ICACCI-2016)* held at Jaipur INDIA. We have updated the Section 2.3 "Our Contribution" highlighting significant extensions in the present manuscript.

(B) The Tables 3 in current manuscript, has been updated making them distinct now compared to Tables 1 in Conference manuscript. Also, the Figures 9 and 10 are also updated vis-a-vis Figures 3 and 4 in conference paper. Additionally, being senior author, I had also obtained IEEE USA, permissions to reuse part of data in conference paper. The IEEE USA has **granted permission** accordingly (copy attached).

We sincerely hope that you accept our findings relevant, and satisfying copyright requirements.

With best regards

(Vineet Sahula)
Email- sahula@ieee.org

IEEE LICENSE
TERMS AND CONDITIONS

Aug 12, 2019

This Agreement between VIneet Sahula ("You") and IEEE ("IEEE") consists of your license details and the terms and conditions provided by IEEE and Copyright Clearance Center.

License Number	4646841165224
License date	Aug 12, 2019
Licensed Content Publisher	IEEE
Licensed Content Publication	IEEE Proceedings
Licensed Content Title	Lightweight security algorithm for low power IoT devices
Licensed Content Author	Tarun Kumar Goyal; Vineet [::Sahula::]
Licensed Content Date	Aug 29, 0005
Type of Use	Journal/Magazine
Requestor type	non-commercial/non-profit
I am an IEEE member OR the author of this IEEE content.	both
IEEE Member ID	03855129
Format	print and electronic
Portion	figures/tables/graphs
Number of figures/tables/graphs	3
Figures/tables/graphs to be used	Table-1; Figure-3; Figure-4;
In the following language(s)	Original language
Order reference number	
Title of the article	Energy Efficient Lightweight Cryptography Algorithms for IoT Devices
Publication the new article is in	IETE Journal of Research
Publisher of the article	other
Author of new article	Tarun Goyal, Vineet Sahula, Deepak Kumawat
Expected publication date	Jun 2019
Estimated size of the article (pages)	12
Requestor Location	VIneet Sahula Department of ECE, MNIT Jaipur INDIA-302017 Jaipur, Rajasthan 302017 India Attn: VIneet Sahula
Total	0.00 USD
Terms and Conditions	

TERMS AND CONDITIONS FOR REUSE OF IEEE MATERIAL SELECTED FOR LICENSING (THE "LICENSED MATERIAL") BASED ON “TYPE OF USE” AND “FORMAT” SELECTED FOR LICENSING BY USER

By clicking "accept" in connection withcompleting this licensing transaction, you, as "User" do agree that the following terms andconditions apply to the use of the material you selected for licensing (the"Licensed Material"), along with the Billing and Payment

terms and conditionsestablished by Copyright Clearance Center, Inc. ("CCC"), at the timethat you opened your RightsLink account and that are available at any time at http://myaccount.copyright.com.

Grant of Limited License

IEEE hereby grants to you a non-exclusive, non-transferable worldwide license to use the Licensed Material in the "TYPE OF USE" and "FORMAT" that you outlined in the RightsLink form in connection with this transaction, and as outlined in accordance with the terms and conditions of this Agreement. This license does not include any photography, illustrations or advertisements that may appear in connection with the Licensed Material and does not extend to any revision or subsequent edition in which the Licensed Material may appear.

Types of Use

Complete terms and conditions and "TYPES OF USE" can be found in the License Agreement that will be available to you during the online order process. Certain terms of use in some IEEE licenses may take precedence over and supersede certain rights granted through RightsLink services. IEEE also reserves the right to restrict the types and total number of items that may be reused in any type of publication or medium.For additional information on the types of use available by IEEE and through RightsLink, please refer to http://www.ieee.org/publications_standards/publications/rights/rightslink_usetypes.html .

Authorized Use

The license granted is granted for a one-time use for REPUBLICATION IN THE "TYPE OF USE" and "FORMAT" that you outlined in the RightsLink form in connection with this transaction, to be completed within one (1) year from the date upon which this license is effective, with a maximum distribution equal to the number that you identified in the RightsLink form in connection with this transaction.

Restrictions on Use

All uses not specifically authorized in this license and specified in the options for reusing IEEE Licensed Material available through the RightsLink service are prohibited, including (i) altering or modifying the Licensed Material in any manner, translating the Licensed Material into another language or creating any derivative work based on the Licensed Material; (ii) storing or archiving the Licensed Material in any electronic medium or in any form now invented or devised in the future, except where permission is granted to do so. For additional information on the types of use available by IEEE and through RightsLink, please refer to http://www.ieee.org/publications_standards/publications/rights/rightslink_usetypes.html .

If the Licensed Material is altered or modified in any manner, it must be within the scope of the license granted and it must not alter the meaning of the Licensed Material or in any way reflect negatively on the IEEE or any writer of the Licensed Material.

Any use of the Licensed Material in any way that would be considered libelous, defamatory, abusive or obscene, in violation of any applicable law or the proprietary rights of a third party and or used in connection with the advertising or promotion of any product or service is also prohibited.

You agree to use your best efforts to prevent unauthorized use of the Licensed Material.

License Effective Only Upon Payment and Author Approval

The license granted to you is effective only upon (i) receipt of full payment fromyou as provided in CCC's Billing and Payment terms and conditions and (ii) yourhaving obtained the author's approval of your proposed use of the LicensedMaterial as described here.

IEEE Intellectual Property Rights

You agree that IEEE is the owner of all right, title and interest in the Licensed Material and/or has theright to license the Licensed Material, including all copyright rights

andother intellectual property rights under United States and international law.

Termination

In the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms andconditions, the license granted herein shall be terminated immediately. Any useof the Licensed Material after termination, as well as any use of the LicensedMaterial beyond the scope of these terms and conditions, may constitutecopyright infringement and IEEE reserves the right to take any and all actionto protect its rights in the Licensed Material.

Copyright Notice

You must include the following copyright and permissionnotice (WITH DETAILS FILLED IN BY YOU) in connection with the Authorized Use ofthe Licensed Material:
© [Year] IEEE. Reprinted, with permission, from [complete publication information].

Warranty and Indemnity

You warrant that you have all rights necessary to enter into this agreement and hereby indemnify andagree to hold harmless IEEE and CCC, and their respective officers, directors,employees and agents, from and against any and all claims arising out of youruse of the Licensed Material other than as specifically authorized pursuant tothis license.

Limited Warranty and Limitation of Liability

THE RIGHT TO USE THELICENSED MATERIAL IS GRANTED ON AN "AS IS" BASIS AND IEEE MAKES NO WARRANTY, EXPRESS OR IMPLIED WITH RESPECT TOTHE LICENSED MATERIAL, INCLUDING ALL WARRANTIES OF QUALITY, ACCURACY AND/ORFITNESS FOR A PARTICULAR PURPOSE, AND IEEE SHALL NOT BE LIABLE UNDER ANYCIRCUMSTANCES FOR ANY LOSSES RESULTING FROM YOUR RELIANCE ON OR USE OF ANYINFORMATION CONTAINED IN THE LICENSED MATERIAL.

No Transfer or Assignment of License

This license is personal to you and may not be sublicensed, assigned, or transferred by you to any otherperson without IEEE's written permission.

Objection to Contrary Terms

IEEE hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement orother writing prepared by you, which terms are inconsistent with these termsand conditions or CCC's Billing and Payment terms and conditions. These termsand conditions, together with CCC's Billing and Payment terms and conditions(which are incorporated herein), comprise the entire agreement between you andIEEE concerning this licensing transaction. In the event of any conflictbetween your obligations established by these terms and conditions and thoseestablished by CCC's Billing and Payment terms and conditions, these terms andconditions shall control.

Payment Terms

If you would like to pay for this license now, please remit this license along with your payment made payable to "COPYRIGHT CLEARANCE CENTER" otherwise you will be invoiced within 48 hours of the license date. Payment should be in the form of a check or money order referencing your account number and this invoice number. Payments should be sent to the address noted below:
Copyright Clearance Center
29118 Network Place
Chicago, IL 60673-1291
Once you receive your invoice for this order, you may pay by credit card. Additional information is provided to users at the time their credit card order is placed.
For suggestions or comments regarding this order, contact RightsLink Customer Support: customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Comments and Questions for IEEE

All comments and/or questions related to RightsLink permission services or comments and questions regarding IEEE licensing policies should be sent to

discoverservices@ieee.org. Users may also call Author Support & Content Discovery staff at 732.562.3965.

A copy of both the license and these terms should be retained for your files.

Other Terms and Conditions:

Updated 12/2011 nbd-IEEE

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.

Replies to reviewers suggestions/comments

Authors wish to express their gratitude and thank anonymous Editor(s)/reviewers for the valuable comments & further advice. We present, in following paragraphs, our views, explanations, replies and incorporated-changes in the manuscripts.

--------Replies to comments (of Editor/Editorial office)----------

Para	Query	Reply of Author(s)
1	Similarity includes sentences and sentence fragments particularly in the following parts –	
(a)	Contents of Table 3 of the manuscript appears to be similar to contents of table 1 of the matched conference paper.	- The contents borrowed have been cited with sources ref [1,10] - An IEEE permission to use this Table-3 in present manuscript (Table-1 in Conference paper) have been obtained, (copy attached) - Citations are updated in caption of Table-3 in manuscript
(b)	Legend of fig. 9 and 10 of the manuscript appears to be similar to the legend of fig. 3 and 4 of the matched conference paper.	- The captions of Figures 9 and 10 are updated using more relevant words, shown in manuscript in BLUE color - The contents (diagram, pictures, plotting) in Figures 9 and 10 are updated - Still, an IEEE permission to use these Figures-9, 10 in present manuscript (Figures 3 and 4 in conference paper) have been obtained, (copy attached)
2	In light of the above, we require you to explain the differences between two papers/report and what new material your submission brings to the subject area. Please include explanation in your manuscript. Please also revise your article to decrease the similarity to the above paper.	- Thanks, now we have rephrased our contribution in Section 2.3 separately in more relevant words, shown in BLUE ink - The present manuscript deals with more efficient symmetric algorithms implementation and evaluation for IoT applications- requiring faster computation, with low power consumption and low area requirements.

Energy Efficient Lightweight Cryptography Algorithms for IoT Devices

Tarun Kumar Goyal, Vineet Sahula, Deepak Kumawat

Tarun Kumar Goyal has been with the Malaviya National Institute of Technology, Jaipur;
He is currently working in Western Digital, Bengaluru, INDIA (email: tarunkgoyal11@gmail.com)
Vineet Sahula is with the Malaviya National Institute of Technology, Jaipur;
He is Fellow of IETE INDIA (e-mail: sahula@ieee.org, ORCID- 0000-0001-9431-4518).
Deepak Kumawat has been with the Malaviya National Institute of Technology, Jaipur;
He is currently an entrepreneur (e-mail: deepakkumawat619@gmail.com).

ABSTRACT

Over few decades, people have been working on providing security solutions, whereas attackers too have been working simultaneously. We present an evaluation of security algorithms, comparing performances and robustness. These comparisons are performed after hardware implementation and use crypt-analysis. The targeted devices are wrist watches, RFID tags, IoT devices and others, which don't have a lot of areas (million of gate equivalent). While performing this, the primary concern has been exploring to find an algorithm that can work in these constrained limits. This led to a search for an algorithm that has low hardware footprints, low power consumption, and better speed but at the same time implements adequate security. PRESENT has been found to be one such suitable algorithm. It has been also included in the new international standard for light-weight cryptographic methods under ISO/IEC 29192-2:2012 for its straight forward and light design. Our paper reports hardware implementation results of PRESENT, AES, ECDH, DH and RSA cryptography algorithms. We have implemented these algorithms with standard gate library of UMC-90nm. Each algorithm has its own architecture and hence requires different crypt-analysis techniques like brute force, Pollard's Rho, and biclique for "difficulty to break" measurement. It is a measure in term of time and data complexity of efforts required of a cryptographic attack. We have obtained 1.7× improvement in area and 63× improvement in power for modified PRESENT algorithm as compared to AES. It has been also been observed that the proposed PRESENT algorithm has a time complexity of break-attack as 2^{127} for 128 bit key length.

Keywords:
Cryptography, crypt analysis, low power, hardware implementation, break-attack, key length, IoT

1. Introduction

Nowadays the internet of things (IoT) has gained a valuable attention from industry and academia. It allows people and things to be connected anytime, anyplace, with anything and anyone, ideally using any path/network any service. One vision of the future is that IoT becomes a utility with increased sophistication in sensing, actuation, communication, control and in creating knowledge from a vast amount of data. This will result in qualitatively different lifestyles from today. For example, smart city, digital world, e- education, e- health many more are some of the applications Authors in [1], highlight the vision and scope of IoT. The cities will be smart lot of things should be embedded, and communicate with each other. It may be happening with washing machine and refrigerators. They would talk to each other, might have vision and thinking capability. Authors in [2], discuss a kind of platform to host deployment of IoT applications which looks like app store or thing-store. They have presented IoT management system, which helps to manage apps using event query language. Authors in [3], have presented a social use of IoT, for example smart city. An architecture of ALMANAC smart city was presented. In which they discussed four layers- the API, virtualization, data management, and smart city resource adaptation.

In a smart city lots of utilities like cleaning, power, water, education, traffic, transport and weather monitoring are connected to peoples. The communication is through internet. When things are severed via internet then security is an issue.

A number of cryptography algorithms developed based on asymmetric and symmetric public keys. Different applications use default algorithms. IoT devices are miniaturized in term of size or area. Due to area constraints it doesn't not employ to heavy processing elements. So conventional security solutions are not suitable for IoT devices. However, this security concern requires such kind of lightweight solution which can be suitable or fit according to a low footprint. Authors in [4], describe the hardware implementation of asymmetric algorithms like Diffie-Hellman (DH), Rivest Shamir Adleman (RSA), & Elliptic Curve Diffie-Hellman (ECDH), and Compared their performance, area, and power. A ECDH is found to be far better than DH and RSA in term of power and area. Which can support some of the IoT devices but is not ultra lightweight. These security algorithms are known to be software efficient compared to hardware implementation. Only few architectures have been tailored for silicon implementation. Authors in [5, 6, 7, 8, 9] obtained better performance and area by effecting improvement in one of the parts or modules of complete Advanced Encryption Standard (AES) like S- box, key expansion, and Mixcolumn. These algorithms might be a solution for those IoT devices which suffer from area and power. An algorithm proposed in [10] called PRESENT, is claimed to be suitable for lightweight & resource constrained IoT hardware. A lot of applications are based on IoT now day's, some of them have append in [1, 2, 3, 11].

1.1 Motivation

For IoT devices, security requirements are high with a low footprint. The domain of security is verse. It has been developing since last few decades. After the development of VLSI technology, a lot of work has been done on the hardware implementation of cryptography. In this scenario security architecture and VLSI design space are rapidly growing. For many VLSI designs, speed and cost requirements vary over different market segments of the targeted applications. High-end segments, such as server CPUs/center node and hardware router/node ASICs, demand high performance, while low-end but high-volume markets such as ASICs for IoT devices demand low cost and power consumption (not necessarily high performance except security). Hence depending on the application (security & low footprint), there is a strong motivation to explore the energy efficient lightweight security solution. Those solutions should target IoT devices, which are suffering from security. Before finalizing, the solution must have adequate security and should be robust. The targeted IoT devices, which have only 2-3k gates space for security purpose and remaining space for functionality and task like sensing, actuation, communication, and control.

1.2 Robustness Comparison

Security of a cryptography algorithm depends on key size. It is measured in terms of time and data complexity. Processes to find time and data complexity differ from algorithm to algorithm. Crypt-analysis for block ciphers (i.e. AES and PRESENT) has been done with help of biclique crypt-analysis. Crypt-analysis for stream cipher (i.e. ECC) has been done with help of Pollard's rho and baby step- giant step. AES, PRESENT, RSA and ECDH are robust and complex. It is shown in Figure 1, complexity in AES & ECDH algorithms are exponential (2^n) and sub-exponential ($\sqrt{2^n}$) respectively, while for RSA is defined by n, where n is shown in equation 1 and L is the number of bits.

$$n = \frac{1.923 * \sqrt[3]{L * ln(2)} * \sqrt[3]{ln(L * ln(2))^2} - 4.69}{ln(2)} \quad (1)$$

The rest of the paper is organized as follows. In Section 2., we discuss related work on lightweight security schemes e.g. symmetric public key crypto-algorithms like PRESENT, AES and asymmetric public key crypto-algorithms like ECDH, RSA. In Section 3., we analyze robustness, i.e. difficulty in breaking the security scheme, for various security algorithms using biclique crypt-analysis and Pollard's rho. Section 4., provides brief idea about hardware implementation methodology of ECDH and PRESENT. In this section, we describe architectures (iterative and parallel) for PRESENT and efficient modulus calculation

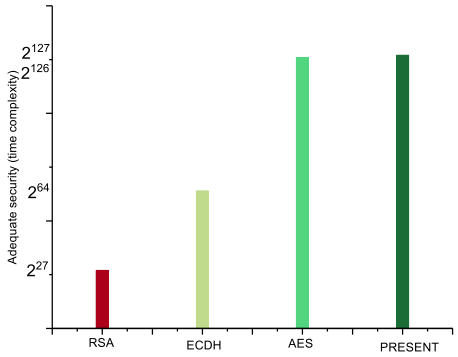


Figure 1: Robustness in terms of time complexity of break-attack (128 bit key)

of a fractional number used in ECDH. In Section 5., we provide results of implementation of PRESENT with optimized S- Box and efficient ECDH with improved modulus method. In this section, we also identify the desired security solution, which possesses a low footprint (lightweight, low power and high throughput) with adequate security. This energy efficient solution is quite useful for IoT devices.

2. Related work

2.1 Previous work

Earlier works done in the area of symmetric algorithm (i.e. DES & AES) to obtained better performance and area by effecting improvement in an area, high throughput (pipeline) and low power. There are various techniques for power optimization. One of the best technique is switching activity. According to equation 2 dynamic power, can control with help of V_{dd} & α. V_{dd} voltage depends on technology and α is switching factor that is created with the help test patterns (*.saif file). Other work targeted crypt-analysis techniques of a block cipher and stream cipher. A lot of work has been done on biclique crypt-analysis, which helps to compare a security analysis for block ciphers (i.e. AES, PRESENT, LED and many more). Crypt-analysis techniques for the stream cipher is Pollard's Rho, Baby Step- Giant Step and Brute Force. Brute Force is applicable for all block and stream ciphers. It is a benchmark. Other techniques obtained better performance in term of time complexity.

$$P = \alpha V_{dd}^2 C_{load} f \tag{2}$$

2.2 Low Foot-Print Approaches

A lot of symmetric key cryptography (DES, AES & PRESENT) were developed. AES is unbreakable up to today's or can say that no attack was successful to achieve a security key. This security depends on some rounds which are 10, 12, 14 for AES-128, AES-192, AES-256 and 31 for PRESENT-128. In this section we discussed a post developed research overview of AES and PRESENT. AES

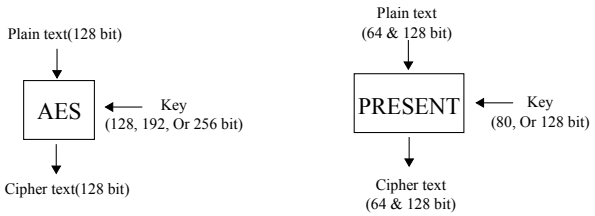


Figure 2: AES symmetric cipher

and PRESENT was developed in 2001 and 2007. Authors obtained better methodology for area, power and performance.

2.2.1 Advanced Encryption Standard

Advanced Encryption Standard (AES) is private key symmetric block cipher, which is faster and stronger than triple DES. In AES 128 Bit plain text converts into 128-bit cipher text with the help of 128 bit, 192 bit, and 256 bit key. It is shown in Figure 2. To increase the complexity, its operation repeated for round of 10, 12, and 14 ($Nr = \frac{Keysize}{4} + 6$). In every round requires four-word byte so total requirement $W_0 - W_{44}$ for 128 bit size. First four word is the key & then they are expanded to calculate remaining forty words for ten round. This technique is called as key expansion. Each round includes Sub Bytes, Shift rows, Mix column, add round key and last round is slightly different in which mix column is not present.

Authors in [5], proposed a hardware implementation of AES, The objective of authors was to get a minimum area and power without worrying about throughput. The novelty of the work was mix columns implementation. Authors presented a submodule (modified multiplier), which calculates a quarter of mix column and inverse mix column operation in one cycle instead of four multipliers. This results in reduction of area. The work has been implemented on 350nm CMOS technology.

However, authors in [6], presented a hardware implementation of AES Encryption. They presented a two S-box, which help to improve throughput (one round computed in 16 clock cycle, so total clock cycle was 176). Using two S-Box(one for subbyte and another for key expansion) area will increase but it reduced with the help of efficient MixColumn / InvColumn, Byte permutation, and 130nm CMOS library.

Authors in [7], presented an AES architecture, whose outcome was the lowest energy per encryption. S-box is lesser contributes in area, so author uses two S-box similarly in [6]. Authors enhance its performance using native S-box and native key expansion. The area is reduced by nine gate counts compared to previous without affecting the performance. Design technology is 65nm CMOS. The approach had a higher efficiency, of 0.83 pJ/bit at 0.32 V.

Authors in [8], presented a compact, low power AES core using small S-box and an improved key expansion block. In this author presented two optimizations, first, the S-box optimization and second Rcon block optimization. The S-box is transformed from GF(2^8) architecture to GF(2^8)/GF(2^4)/GF(2^2) and inverse. However, Rcon

Table 1: Comparison of hardware implementation of AES Algorithm

AES-128	***Technology*** (μm)	***Area (No. of Gates)***	***Throughput*** ($Mbps$)	***Frequency*** (MHz)	***Power*** ($\mu W/MHz$)
[5]	0.35	3400	9.9	80	4.5
[6]	0.13	3900	232	290	62
[7]	0.065	0.012 mm^2	-	11	14.6
[8]	0.18	2900	-	50.5	34
[9]	0.13	5500	-	12	99
[12]	0.18	2421	0.61	0.1	-

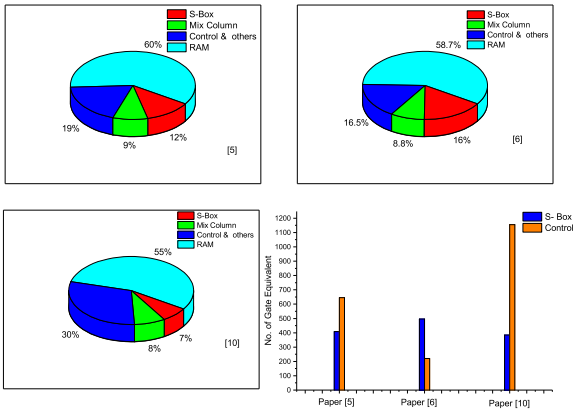


Figure 3: A comparison of S box and Control Gates [5, 6, 9]

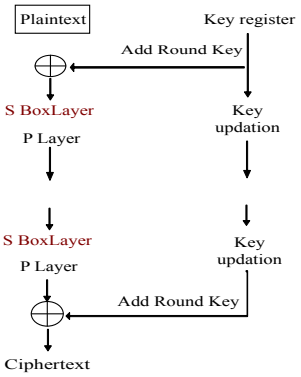


Figure 4: Block diagram of PRESENT [10]

block is optimized by the simple Boolean optimization method.

Moreover authors in [9], implemented single S-box operator using the composite field. Single S-box was used for subbyte and key expansion to save the resource. Resource sharing requires a lot of control gates and circuits. Thereby, increasing the overall gate counts. But saving of one S-box, gate was not too much. Figure 3 show a comparison of gate equivalent of S-box and control for three algorithms.

The discussed methods have been tabulated in Table (1). The methods are expressed in terms of area, power and performance optimization. On basis of cryptanalysis, AES is still robust. Authors in [12], satisfy a lightweight property (No. of GE is 2412) but their throughput was 0.61 Mbps. According to us this is not suitable for targeted IoT devices like nodes, sensors, RFID tags and wrist watch. Authors in [6], achieve a desirable performance but they pay extra penalty for area. According to previous works, we compared their area, power and performance. This results far away from our proposed solution.

2.2.2 PRESENT

PRESENT is a symmetric block cipher having light-weight cipher properties. It was developed in 2007 by the Orange Labs (France), Ruhr-University Bochum (Germany) and the Technical University of Denmark in 2007 [10], designed by Andrey Bogdanov, Axel Poschmann, Christof Paar, C. Vikkelsoe, Gregor Leander, Lars R. Knudsen, Matthew J. B. Robshaw, Yannick Seurin, and. This algorithm is famous for its compact design. The design of PRESENT is evolved from the most hardware efficient AES finalist SERPENT [13]. Even the name PRESENT is a witty manipulation of the word SERPENT.

PRESENT is built on the Substitution-Permutation or simply SP-network model [14] consisting 32 rounds of operation. The size of the block used for *plaintext* is 64 bits with two variants of keys, one of 80 bits and another of 128 bits. With *tag* based constrained devices in mind, 80-bit key version is adequate for security purposes. This cipher was developed for situations like low computational power and low chip area. It uses 4-bit S-Box for hardware optimizations in comparison to heavier ciphers that uses 8-bit S-Box design. It also uses a Permutation Layer or simply *pLayer* for performing *bitwise* permutations.

Encryption routine consists a key update and addition process, a linear *bitwise* permutation layer and a non-linear S-Box layer. Now the first 31 rounds perform XOR operation between the round key K_i for $1 \leq i \leq 32$ and the updated block while the 32^{nd} round is just for key addition. The non-linear S-Box layer employs a single 4-bit S-Box S running parallel 16 times in every round. The key schedule process produces 32 round keys using the pre-supplied key. Figure 4 describes the complete encryption routine algorithm of PRESENT cipher shown in Algorithm 1.

In [4], presented an efficient inversion module and key exchange scheme for ECC. Among which the most critical modules of ECC was inversion method. Key exchange scheme was DH [15]. Similarly authors in [16] target a inversion modules. They compare four inversion methods and select one of the best method (iterative Frobenius map) based on performance. ECC is not suitable for low power devices. It was implemented with millions of gate

Algorithm 1 PRESENT Encryption

1. Input block provided by User $b_{63}...b_0$ is XORed with the input key $K_i = \kappa^i_{63}...\kappa^i_0$, $b_j \rightarrow b_j \oplus \kappa^i_j$.
2. This intermediate output is processed for substitutions by S Box Layer.
3. Then P Layer performs the permutations of the bits, $P(i) = \begin{cases} i * 16 \; mod \; 63, & i \in \{0, \ldots, 62\} \\ 63, & i = 63 \end{cases}$
4. Key Updation takes place in parallel for next rounds.
5. Above steps are repeated for 32 Rounds.

equivalents. Authors in [17] use partitioning schemes. In which some are implemented in software and remaining are hardware implementation this is called as hardware and software co-design.

An RSA [18] algorithm can be divided into three partitions RSA, exponentiation, and modular arithmetic. According to a previously known fact, software suffers in case of performance compared to hardware. If area is high, then the cost will be too much. Which solution is best? It is decided on basis of the problem, and some solution is according to partitioning. In partitioning, some part of the design are implemented with software and some with hardware. The author presented codesign which satisfy performance and flexibility trade-off. An RSA algorithm has exponentiation calculation which consumes a lot of area compared to other ECC, AES, and PRESENT. This area penalty makes a difficult choice to choose RSA for Low area devices like IoT.

This paper's primary focus is achieve to lightweight & low power algorithm for security purpose. We choose a best one algorithm from each side (Symmetric and Asymmetric public-key cryptography [19]) second thing selection and comparison are based on same security level or difficulty level. By these things, we decide to take existing algorithm like ECDH and PRESENT. Our work targets to make them hardware efficient.

First, we targets low throughput, high complexity blocks in existing algorithm like modulation of a fractional number and S Box (requires RAM). We designed the S-Box with Boolean expression using logic gates despite block RAM. IoT devices don't have any room for RAM/ROM. If S-Box is designed using look up table instead of combinational circuit then pay high area penalty. Suppose this S-Box stores in ROM, ROM have 8*256 bit = 2048 bit area. Store requirement has much more impact on the overall size of the circuit.

Second, In ECDH architecture modulus of fractional number consumes too much area. So designed an efficient hardware block (based on Gauss's algorithm) which calculates a modulus of fractional numbers.

Third, IoT devices haves high frequency. But existing algorithm is slow. To increase performance (throughput and frequency) of an algorithm, we presented parallelism in architecture and low power design. Our work includes both iterative and parallel architecture of PRESENT algorithm which gives the highest throughput compare to any existing algorithm.

Fourth, what is the process to find security level (robustness) of the algorithm is crypt-analysis? Our work explores the crypt-analysis techniques for symmetric and asymmetric public keys in which we present differential and linear attacks and their difficulty or loophole of keys based on Brute force, Baby step- giant step, pollard's Rho and biclique crypt-analysis techniques.

After this design, we have implemented an application-specific integrated circuit (ASIC) of ECDH & PRESENT algorithm (iterative and parallel architecture) and identified a efficient security solution for IoT devices which suffers from an area, power and performance. In recent years, many other lightweight variants of block ciphers have been proposed- GIFT [20], SKINNY [21], MIDORI [22] and SIMON [23]. However, they differ in robustness against attacks.

2.3 Our contribution

We have focused on improving the performance & area/power features of PRESENT algorithm. The ECDH has traditionally found suitable, primarily for key exchange.

Our contribution lies in proposing certain modifications on PRESENT/ECDH. The contributions of present manuscript are as follows in contrast to reported results in [4].

- Improved S-Box (PRESENT). Comparative evaluation of standard S-box and proposed Optimized S-box is performed. Using the proposed method, we could save 60 LUT's (FPGA implementation). Enhanced throughput and reduced area have been achieved, as area of hardware is a major concern for IoT devices.
- We have also explored PRESENT algorithms in its 64-bit as well as 128-bit version, for its normal iterative as well as faster parallel version realized in hardware; establishing that parallel version provides 10× throughput.
- Comparison of symmetric algorithms for low footprint, which could support an IoT device
- We compare the security strength of symmetric algorithms, asymmetric algorithms based on theoretic framework of crypt-analysis

We have evaluated security strength i.e. robustness against attack. The strength is presented for various algorithms in terms of equivalent security strength provided by certain length of key (bits). We have used theoretic framework for crypt-analysis, i.e. indicative order of time complexity needed to break the respective cryptography protocol [24, 25].

3. Crypt-analysis Techniques

The general purpose of cryptography is to secure the data (plain text) from cyber-thieves (like called foes, assailants, interceptors). Cyber terrorists are accepted to have complete access to the correspondence between the sender and beneficiary. Crypt-analysis is the process of recovery of the plain text of a message without access to the key. It likewise may discover shortcomings in a cryptosystem that inevitably prompt the past results. In this

section, an attempt has been given to get an idea of hardness. Lets see an example of four digit lock having 10^4 combinations to unlock. Here a digit can be 0-9 so there are 10 possibilities for all n digit. Now discussing about binary numbers (0 and 1). For N bit key size 2^N combinations are possible according to brute force. Differential, linear, integral, and impossible differential analyses are most practical in term of data & time complexity. A meet in the middle (MITM) attack is suitable for block ciphers like AES, LED, Picallo, and PRESENT. A biclique crypt-analysis technique is based on MITM attack. The computations for registering discrete logarithms on elliptic curve- the baby step, giant step method, and Pollard's rho method have been borrowed from [4, 26] and tabulated in Table 3.

3.1 Biclique Crypt-analysis

Firstly this concept is used for hash function crypt-analysis. Then it is implemented on block cipher algorithms. It is obvious that every crypt-analysis was taking a reference of worst case which is brute force and trying to create a new benchmark. A biclique is characterized by its length and dimension. The advantages of biclique, over brute force: cost of constructing biclique and matching computation. Biclique crypt-analysis is two types: long biclique and independent biclique. Independent biclique is more efficient so it is considered. In which partial rounds are counted in bicliques and remaining are counted with brute force. Example attack on m (out of r) rounds with help of biclique and remaining (r-m) with brute. Probability of success for independent biclique is 1.

$$C_{total} = 2^{k-2d}(C_{biclique} + C_{precomp} + C_{recomp} + C_{falsepos}) \quad (3)$$

Here, $d = 3$, $C_{biclique}$ is the cost for constructing a biclique ($= 2^{3+1} * \frac{4}{31} = 2^{1.05}$), $C_{precomp}$is the precomputation of matching check for most 4 bits ($= 2^3 * \frac{27}{31} = 2^{2.8}$), C_{recomp} is the complexity of recomputing ($2^{2d} = 2^{2*3}$) values in both direction forward and reverse, $C_{falsepos}$ is the computational complexity by false positive ($2^{2*3-4} = 2^2$)[27].

Authors in [28, 29, 30, 31, 32], presented biclique crypt-analysis for PRESENT and AES. It's result improved 3 to 5 time compared to brute force. This is slightly good but in terms of 2^n, it is too small. A comparison of their results is shown in Table 2 using equation 3. Select the best result for PRESENT, which is data complexity 2^{23} and computational complexity $2^{127.32}$. Similarly for AES data complexity is 2^{56} and computational complexity is $2^{126.13}$. This results show that difficulty (Security) was unaffected. On bases of difficulty we picked one by one algorithm for ASIC implementation and compare their performances, area and power.

3.2 Baby Step-Giant Step

Before entering the points of interest of the calculation, a fast thought: we can simply compose any number x as $x = am + b$, where a, m and b are three arbitrary integers. For example, we can write $22 = 3 * 5 + 7$.

Table 2: Biclique Crypt-analysis: Complexity of needed attack-efforts

Target algorithm	*Rounds*	*Data Complexity*	*Time Complexity*
PRESENT-128[28]	Full (31)	2^{19}	$2^{127.81}$
PRESENT-128 [29]	Full (31)	2^{44}	$2^{127.37}$
PRESENT-128 [32]	Full (31)	2^{23}	$2^{127.32}$
AES-128 [30]	Full (10)	2^{88}	$2^{126.18}$
AES-128 [31]	Full (10)	2^{56}	$2^{126.13}$

The baby-step giant-step is a "meet in the middle" algorithm. It is shown by Algorithm 2

Algorithm 2 Baby Step-Giant Step

we can simply compose any number$x$ as $x = a * m + b$, $Q = x * P = (a * m + b) * P$

1. Calculate $m = [\sqrt{n}]$where x, a, m $and n$ are integers
2. For every b in $0, 1, 2.....m$ calculate bP and store in the hash table.
3. For every a in $0, 1, 2.....m$ calculate amp & $Q - amp$
4. check hash table if there is exist a $Q - amp = bP$ If it exists then we found $x = am + b$

3.3 Pollard's Rho

Given P and Q then find x such that $Q = xP$. With Pollard's rho [26], we will solve a slightly different problem: find integer a, b, A, B such that shown by Algorithm 3

Algorithm 3 Pollard's Rho

1. Given P and Q then find x such that $Q = xP$ With Pollard's rho, we will solve a slightly different problem: find integer a, b, A, B such that $aP + bQ = AP + BQ$ put $Q = xP$
2. Then $aP + bxP = AP + BxP$ simply $(a - A)P = (B-b)xP$ Presently we can dispose of P , But before doing as such, recollect that our subgroup is cyclic with order n
3. $a - A = (B - b)x\ (mod\ n)$ find out value of x $x = (a - A)(B - b)^{-1} mod\ n$

The principle of operation of Pollard's rho is simple: we define a pseudo-random sequence of (a,b) pairs just 109-bit long curves have been effectively broken. The most recent fruitful endeavor was made in 2004. The prize was honored on 8 April 2004 at a gathering of around 2600 individuals spoke to by Chris Monico. They likewise utilized a variant of a parallelized Pollard rho strategy, taking 17 months of calendar time.

$$17\ months * \frac{\sqrt{2^{192}}}{\sqrt{2^{109}}} \approx 5 * 10^{13}$$

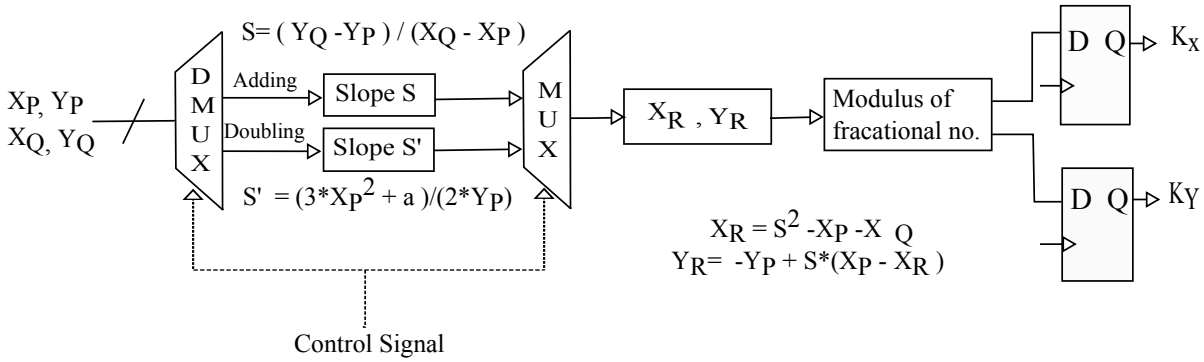


Figure 5: Architecture of ECDH

Table 3: Crypt-analysis for ECC illustrating Complexity of attack-algorithm [4, 26]

Algorithm	*Time Complexity*	*Space Complexity*
Brute-Force	O(n) =O(2^{192})	O(l)
Baby-Step, Giant-Step	O(l)	O($\sqrt{n}$)=O($2^{\frac{192}{2}}$)
Pollard's Rho	O($\sqrt{n}$)=O($2^{\frac{192}{2}}$)	O(l)

Algorithm 4 Modulus of a Fractional Number

1. Check value of D*i is just greater than P for i=1,2,3.....,P-1 If Yes go to step 2 and No check for higher i values.
2. Update the values N, D $N = N * i$ %.
3. If value of D is 1 then go to step 4 otherwise repeat from step 1.
4. Modulus of Fractional number is N

4. Hardware implementation

4.1 Architecture of ECDH

We have discussed about ECDH algorithm in the previous section. In this section, we present architectural implementation. A designer's concern is to trade-off area for improvement in performance to the extent possible. This architecture contains adders, shifters, multipliers, and hardware for finding modulus of a fractional number. This is shown in Figure 5. Adder (carry look ahead) and Wallace tree multiplier can be used according to target devices. Modulus of fractional number is calculated using Gauss algorithm [4].

4.1.1 Modulus of a Fractional Number

Modulus of an integer number is far easier to compute, compared to modulus of a fractional number. Some of the proposed methods in literature for computing modulus of fractional numbers, are complex and not efficient for hardware implementation. We propose an efficient method described in Algorithm 4 and illustrated in Figure 7. In this method, N denotes numerator, whereas D denotes denominator. Computation of modulus of fractional number with prime number P is described as $Mod = (N/D)\ \%\ P$

4.2 Architecture of PRESENT

4.2.1 Iterative Architecture of PRESENT

The primary concern during this design is to save area and power, and for the purpose iterative design architecture is chosen for 80-bit PRESENT cipher. This design architecture utilizes one S-Box layer and one permutation layer. One round of PRESENT is performed in one clock cycle. We require a 64-bit data-register and an 80-bit key-register to store the values of *plaintext* and the key. The data path has one *bitwise* XOR of 64 bits size, 16 S-Boxes working alongside, and one P-Layer for permutations shown in Figure 8. The key updation process requires an 80-bit key-register, a 61-bit left rotation usually wiring, one S-Box, and a 5-bit XOR operation. Initially, the *plaintext* and the secret key used are saved in the relevant registers. The value of data register and the key register is updated after each round. A 5-bit round counter is also required to process 32 rounds. At the end of 31 rounds, data in the data-register is finally XORed with the key of 32^{nd} round.

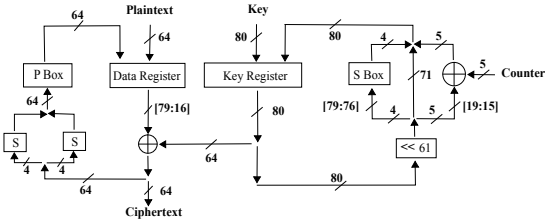


Figure 8: Iterative Architecture of PRESENT

4.2.2 Parallel Architecture of PRESENT

For designing the parallel architecture for the PRESENT, whole operation of encryption consisting 31 rounds is unrolled. Now, instead of using same data path for every round like in Iterative design, each round has its own data path. Round keys are calculated from the 80-bit key supplied by the user and are available at all times during the operation. Figure 6 explains the data path of the parallel architecture. The whole process consists of 32 XORs, 496 S-Boxes, and 31 P-Boxes to compute

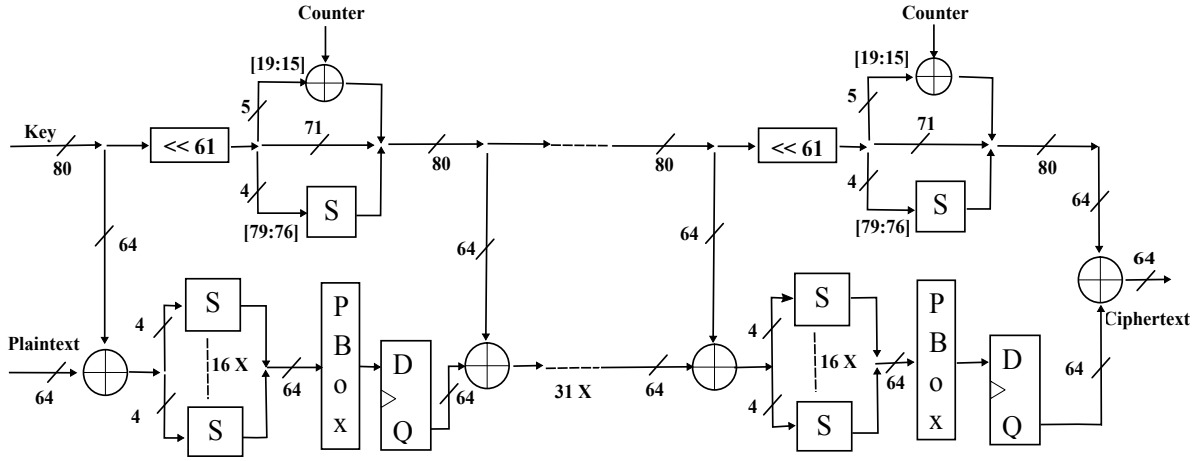


Figure 6: Parallel Architecture of PRESENT

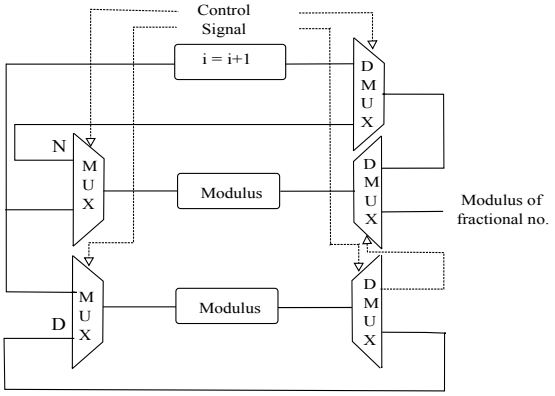


Figure 7: Architecture of modulus of fractional number

the *ciphertext*. The key updation process requires 31 S-Boxes and 31 XORs. To increase the maximum clock frequency, after the P-Box of each round, pipelined registers are added. Although, this design requires higher chip area but the throughput is significantly increased. Once the pipeline registers get full initially i.e. after 31 clock cycles, it encrypts the *plaintext* into *ciphertext* in every clock cycle.

4.2.3 Optimized S-Box Design for PRESENT

The standard design of PRESENT cipher makes use of LUTs for the implementation of the S-Box. An S-Box design based on LUT or BRAM occupies a large amount of space for its operation and due to their fix architecture there is not much room for optimization in their size. So to optimize the design performance, another option is considered. We decided to design the S-Box with Boolean expressions using logic gates. We used **Logic Friday**, a graphical user interface for espresso tool to generate these minimized expressions for the output bits of S-Box.

As discussed, the PRESENT S-Box is a 4-bit to 4-bit S-Box. Now, let us consider the input to this 4-bit S-Box be $x = (x_3\|x_2\|x_1\|x_0)$ and similarly the 4-bit output be $S(x) = (S_3(x)\|S_2(x)\|S_1(x)\|S_0(x))$. The Logic Friday tool provided the following minimal expressions for the S-Box output bits.

- $S_3[x] = \overline{x}_3x_2x_1\overline{x}_0+\overline{x}_3\overline{x}_2\overline{x}_1\overline{x}_0+\overline{x}_3x_2\overline{x}_1\overline{x}_0+x_3\overline{x}_2x_1+\overline{x}_3x_2x_1x_0+x_3\overline{x}_2\overline{x}_1x_0+\overline{x}_3\overline{x}_2x_1x_0$
- $S_2[x] = \overline{x}_3\overline{x}_2\overline{x}_1\overline{x}_0+\overline{x}_3\overline{x}_2\overline{x}_1x_0+x_3x_2\overline{x}_1+\overline{x}_3x_2x_1x_0+\overline{x}_2x_1\overline{x}_0+x_3\overline{x}_2\overline{x}_1x_0$
- $S_1[x] = \overline{x}_3x_2x_1\overline{x}_0+x_3x_2x_0+\overline{x}_2x_1\overline{x}_0+x_3\overline{x}_2\overline{x}_1x_0+\overline{x}_3\overline{x}_2x_1x_0+x_3\overline{x}_2x_0$
- $S_0[x] = x_3x_2\overline{x}_1x_0+\overline{x}_3x_2\overline{x}_1\overline{x}_0+\overline{x}_3\overline{x}_2\overline{x}_1x_0+x_3x_1\overline{x}_0+\overline{x}_3x_2x_1x_0+\overline{x}_3\overline{x}_2x_1x_0+x_3\overline{x}_2\overline{x}_0$

The benefit of this conversion is that the whole operation of S-Box can be expressed using Boolean expressions.

Table 4: Key Exchange between Alice and Bob

User	***Private Key***	***Base Point***	***Public Key***	***Share Key***
Alice	121	(2,2)	(115,48)	(161,69)
Bob	203	(2,2)	(130,203)	(161,69)

5. Results

5.1 Software Implementation

5.1.1 Elliptic Curve Diffie-Hellman

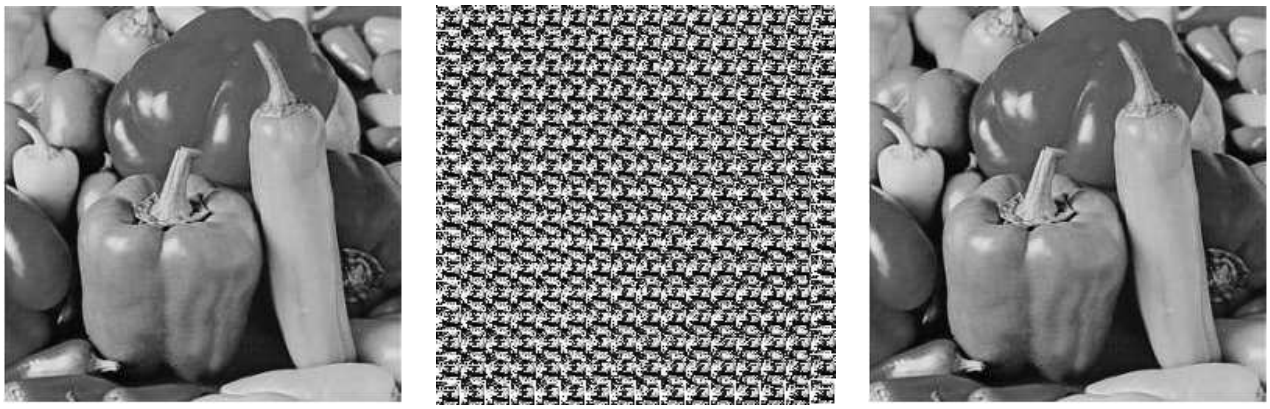
A general elliptic curve is taken that is represented by the following equation: E: $Y^2 = (X^3 + aX + b) \, mod \, P$ where X, Y are elements of GF(P) and a,b are the integers modulo P, satisfying: $4a^3 + 27b^2 \neq 0 (\, mod \, P)$

Generate elliptic curve values as following are chosen $a = 2$, $b = 0$ and $P = 37$. The private and public key pair for user A and B are generated as shown in Table 4

Encryption

ECDH is a key exchange scheme, as we wish to validate the larger number of elliptical points/ coordinates in an encryption scheme, let's choose an image message. For example illustration, let's have sample image of Figure 9 (a), read each of its pixel value and transformed these values into a ciphertext by XORing with the shared key. Message is shown in Figure 9.

Values of pixel (M) will vary between 0 to 255 (Black to White). Encryption operation will be as follows: Plain text (M_i)= 144. Cipher text (C_i) = M_i XOR (K_{a_x} $*K_{a_y}$). Encrypted image is shown in Figure 9 (b). We



(a) Plain text image (b) Encrypted image (c) Decrypted image

Figure 9: Validating encryption and decryption for ECDH with Image message

use XOR operation on during processing, which bleaches CBC. In order to mitigate the impact of CBC, we have used round operations, which renders image like having regular patterns.

Decryption

Now, Bob will receive C_i and calculate the plain text M_i using his private key K_b. Decryption process will be as follows: M_i=C_i XOR (K_{b_x} *K_{b_y}). Decrypted image is shown in Figure 9 (c).

5.1.2 PRESENT SCHEME

Encryption Cipher designs are implemented using Verilog and simulations are performed in Xilinx ISE 14.7. For Encryption routine we have taken 64-bit hexadecimal plaintext *2e4780e27f27f830* and a 80-bit key *2f0c5f6a1-cb4de011239*. After successful implementation of the algorithm we got the ciphertext value as *ed929c2635d40836*.

Decryption The decryption module works like the encryption module. The 32^{nd} key of the encryption module is required for the first round of decryption module. For compact design, we have assumed that this key is already computed and available at the beginning itself. So we provided values of ciphertext *ed929c2635d40836* and 32^{nd} key *e2f62a9eca945eb76026* and successfully retrieved our plaintext back.

5.2 Hardware Implementation

5.2.1 PRESENT implementation

After successfully implementation of the PRESENT (Standard and optimized S-box) scheme, the results are shown in Table 5 and its variants on FPGA. The results reflect an area difference (more than 50 LUTs). For experimentation, we synthesized this PRESENT-extend version using Synopsis Design Compiler with 90 nm UMC FARADAY library.

Both Iterative and Parallel Architecture designs have been synthesized, in order to analyze their overall performance. This synthesis provided us with area, power and performance details of our design. Table **??** shows the Area comparison results of the two architectures. It shows the design area in $\mu m^2$and corresponding Gate Equivalents (GE), calculated by dividing the total design area with area of 2-input NAND Gate. The GE of our Iterative design is 1920.7 only, satisfying the 2000 GE condition of RFIDs. While the Parallel Architecture has 29575.2 GEs suitable for applications where area is not a constraint. The iterative design reuses same components repeatedly to achieve reduction in the chip area, but it increases the latency to 32 cycles. The design takes the next input after 32 cycles and thus both latency and throughput are equal i.e. 32 cycles. While for the parallel design the output corresponding to any input, comes after 32 cycle but once the pipeline is filled we get output every cycle. Here, the latency is 32 cycles, while throughput is one cycle only. Table 6 shows the Power consumption comparison between both the architectures. The Iterative design has Total Dynamic Power in few mW range while the power of Parallel design is nearly 10 times higher.

5.2.2 ECDH implementation

In Section 4., we have presented the architecture of ECDH. A hardware implementation of ECDH was too much costly area-wise, compared to software implementation. In ECDH architecture, a lot of area (million number of gates) is consumed by multiplier and modulus of fractional number. According to requirements of low area and high throughput, efficient multiplier could be used like Booth's, Wallace tree, systolic array multiplier. Similarly, we have tried to get an efficient solution for modulus of fractional number. However, our design wasn't as efficient as PRESENT scheme. Nevertheless, ECDH was was found to be fast and efficient compared to other asymmetric algorithms viz. RSA and DH. Diffie-Hellman key exchange algorithm, Rivest-Shamir-Adleman (RSA) algorithm, and Elliptic curve Diffie-Hellman (ECDH) algorithm were implemented. The dynamic power dissipated is generally classified as a sum of the internal power and switching power. Using PrimeTime (Synopsys), the computed total dynamic power [4] for three of these algorithms is shown in Figure 10.

Table 5: Standard and Optimized Iterative design of PRESENT

S Box Style	***LUTs***	***FFs***	***Total Eq. Slices***	***Frequency*** (MHz)	***Clock cycles***	***Through-put*** $(Mbps)$	***Area Efficiency*** $(Mbps/Slices)$
Standard	347	149	224	226	32	452	2.01
Optimized (Proposed Boolean)	287	149	194	233	32	466	2.40

Table 6: Area and Power comparison results between different designs

Algorithms	***Architecture***	***Area*** (μm^2)	***Area*** (GEs)	***Total Dynamic Power*** (mW)
PRESENT-80	Iterative	6023.47	1920.7	0.834
PRESENT-80	Parallel	92747.98	29575.2	8.441
PRESENT-128	Iterative	7124.60	2271.6	0.918
PRESENT-128	Parallel	95508.44	30455.4	8.578

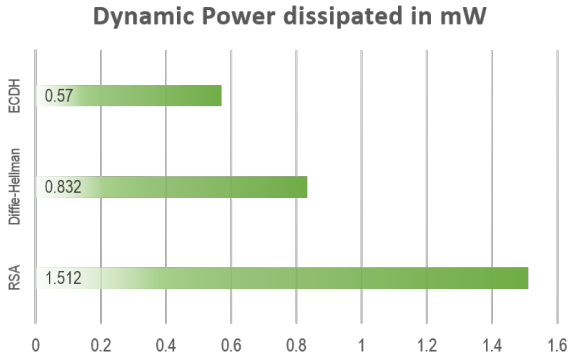


Figure 10: Comparative evaluation of dynamic power for three algorithms- RSA, DH, and ECDH [4]

6. Conclusions

We have proposed improved PRESENT block cipher and ECDH, which have light-weight properties and are suitable for constrained devices. This paper explores different implementations of PRESENT cipher on various platforms ranging from FPGAs to ASICs. The augmented designs have shown lightweight properties and are suitable for low resource devices. We have optimized and augmented the PRESENT by improving the S-Box design. The results have shown that there is a significant decrease in the total slices used by the design. The devices with lesser resources and power capability, aim at a design with low chip area, where throughput may not be of much concern. Moreover, with low chip area, the proposed PRESENT cipher also provides adequate security along with.

The second parallel implementation is designed for devices, where area and power are not constraints but higher throughput is desired. Table 7 shows the Performance comparison between PRESENT-80/128, AES and ECDH architectures. These results prove to be good solution for IoT devices. Developers are trying to create low-cost IoT devices. However, it is a fact that they don't have too much room for security. That's why several people have raised a concern about security. This Table 7 and Table 8 show the answer to both the problems.

Table 8 summaries the robustness for various algorithms. Assuming key size to be N, this is described as what is the hardness in algorithm. These comparisons are based on time complexity needed by a breaking-attack. A row in this table illustrates the required key length for an equivalent security strength. Actual key sizes, used are 128, 192, 256 for AES, and 80, 128 for PRESENT.

PRESENT-128 algorithm is superior as compared to other algorithms in term of security strength, area, power and throughput. The hardware design of PRESENT possesses following features.

- **Adequate Security:** For 128 bit key, time complexity for PRESENT- 2^{127}, AES- 2^{126}, & ECDH- 2^{64} . PRESENT offers high security with equivalent number of bits.
- **Low Area:** S Box Optimization saves area, hence lesser gate equivalent. We have got adequate security with 2272 GEs. The S-Box is of 4 bits as compared to AES which has S-box of 8 bits.
- **Low Power Design:** power consumption is (0.98 μW $/MHz$). This is lowest among all.
- **Throughput** is the highest (1.86 $Gbps$) as compare to AES and ECDH without paying any area penalty.

We infer that proposed modified PRESENT is better, more suitable than AES in terms of low power, satisfies area constraints, as well possesses better security (2^{128} versus 2^{127} for 128-bit version). The work is continuing for comparative exploration of many of recently proposed block cipher schemes [20, 21, 22, 23] for resource constrained applications.

Acknowledgments

The authors gratefully acknowledge *SMDP-C2SD project (ODRC No. 1000110086)* project funded by *Ministry of Electronics & IT, Government of India* for technical support.

Table 7: Performance Comparison of different algorithms

Algorithms	***Technology*** *(nm)*	***Area*** *(GEs)*	***Max. Freq.*** *(MHz)*	***Max. T'put*** *(Gbps)*	***Area Eff.*** *($Mbps/\mu m^2$)*	***Power*** *($\mu W/MHz$)*
PRESENT-80	90	1921	934.57	1.86	0.31	0.89
PRESENT-128	90	2272	934.57	1.86	0.26	0.98
AES-128[6]	130	3900	290.00	0.23	-	62
ECDH-10[4]	90	10204	44.64	0.446	0.014	12.76

Table 8: Robustness Comparison: Key size for equivalent security strength [4]

Key Size (No. of Bits)			
AES	***PRESENT***	***RSA***	***ECDH***
56	56	512	112
80	80	1024	160
112	112	2048	224
128	128	3072	256
256	256	15360	512

REFERENCES

[1] J. A. Stankovic, "Research directions for the internet of things," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 3–9, Feb 2014.

[2] K. Akpinar, K. A. Hua, and K. Li, "Thingstore: A platform for internet-of-things application development and deployment," in *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems*, ser. DEBS '15. New York, NY, USA: ACM, 2015, pp. 162–173. [Online]. Available: http://doi.acm.org/10.1145/2675743.2771833

[3] D. Bonino, M. T. D. Alizo, A. Alapetite, T. Gilbert, M. Axling, H. U. andJose Angel Carvajal Soto, and M. Spirito, "Almanac: Internet of things for smart cities," in *Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on*, Aug 2015, pp. 309–316.

[4] T. K. Goyal and V. Sahula, "Lightweight security algorithm for low power iot devices," *International Conference on Computing, Communications and Informatics*, pp. 1740–1744, sept 2016.

[5] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "Aes implementation on a grain of sand," *IEE Proceedings - Information Security*, vol. 152, no. 1, pp. 13–20, Oct 2005.

[6] P. Hamalainen, T. Alho, M. Hannikainen, and T. D. Hamalainen, "Design and implementation of low-area and low-power aes encryption hardware core," in *9th EUROMICRO Conference on Digital System Design (DSD'06)*, 2006, pp. 577–583.

[7] W. Zhao, Y. Ha, and M. Alioto, "Aes architectures for minimum-energy operation and silicon demonstration in 65nm with lowest energy per encryption," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 2349–2352.

[8] V. L. Dao, V. P. Hoang, A. T. Nguyen, and Q. M. Le, "A compact, low power aes core on 180nm cmos process," in *2016 International Conference on IC Design and Technology (ICICDT)*, June 2016, pp. 1–5.

[9] T. Good and M. Benaissa, "692-nw advanced encryption standard (aes) on a 0.13 um cmos," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, pp. 1753–1757, Dec. 2010. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5256141

[10] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, *PRESENT: An Ultra-Lightweight Block Cipher*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 450–466. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-74735-2_31

[11] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.

[12] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, *Pushing the Limits: A Very Compact and a Threshold Implementation of AES*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 69–88. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20465-4_6

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

[14] A. J. Menezes, P. C. v. Oorschot, and S. A. Vanstone, *PRESENT: An Ultra-Lightweight Block Cipher*. Boca Raton, Florida, USA: CRC Press, 1996.

[15] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, Nov 1976.

[16] M.-K. Lee, K. T. Kim, H. Kim, and D. K. Kim, *Efficient Hardware Implementation of Elliptic Curve Cryptography over GF(p m)*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 207–217. [Online]. Available: http://dx.doi.org/10.1007/11604938_16

[17] M. U. Sharif, R. Shahid, K. Gaj, and M. Rogawski, "Hardware-software codesign of rsa for optimal performance vs. flexibility trade-off," in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, Aug 2016, pp. 1–4.

[18] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978. [Online]. Available: http://doi.acm.org/10.1145/359340.359342

[19] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 5th ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2010.

[20] S. Banik, S. Kumar Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "GIFT: A Small Present Towards Reaching the Limit of Lightweight Encryption (Full version)," Tech. Rep., 2017. [Online]. Available: https://infoscience.epfl.ch/record/232021/files/622.pdf

[21] J. Ge, Y. Xu, R. Liu, E. Si, N. Shang, and A. Wang, "Power Attack and Protected Implementation on Lightweight Block Cipher SKINNY," in *2018 13th Asia Joint Conference on Information Security (AsiaJCIS)*. IEEE, aug 2018, pp. 69–74. [Online]. Available: https://ieeexplore.ieee.org/document/8453764/

[22] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "FPGA-Based Assessment of Midori and Gift Lightweight Block Ciphers." Springer, Cham, oct 2018, pp. 745–755. [Online]. Available: http://link.springer.com/10.1007/978-3-030-01950-1{_}45

[23] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, "The simon and speck lightweight block ciphers," in *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2015, pp. 1–6.

[24] J. Stern, D. Pointcheval, J. Malone-lee, and N. P. Smart, "Flaws in applying proof methodologies to signature schemes," in *In Advances in Cryptology crypto'02, Santa Barbara, Lectures Notes in Computer Science 2442*. Springer-Verlag, 2002, pp. 93–110.

[25] T. Kunz, S. Okunick, and U. Pordesch, "Data structure for the security suitability of cryptographic algorithms (dssc)," RFC_5698, Nov. 2009. [Online]. Available: https://tools.ietf.org/html/rfc5698

[26] J. M. Pollard, "Monte carlo methods for index computation," *Mathematics of Computation*, vol. 32, pp. 918–924, 1978.

[27] K. Jeong, H. Kang, C. Lee, J. Sung, and S. Hong, "Biclique cryptanalysis of lightweight block ciphers present, piccolo and led," 2012.

[28] C. Lee, "Biclique cryptanalysis of present-80 and present-128," *The Journal of Supercomputing*, vol. 70, no. 1, pp. 95–103, 2014.

[29] F. Abed, C. Forler, E. List, S. Lucks, and J. Wenzel, "Biclique cryptanalysis of the present and led lightweight ciphers," 2012.

[30] A. Bogdanov, D. Khovratovich, and C. Rechberger, *Biclique Cryptanalysis of the Full AES*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 344–371. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-25385-0_19

[31] B. Tao and H. Wu, *Improving the Biclique Cryptanalysis of AES*. Cham: Springer International Publishing, 2015, pp. 39–56. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-19962-7_3

[32] F. Abed, C. Forler, E. List, S. Lucks, and J. Wenzel, "Biclique cryptanalysis of present, led, and klein revision 2013-05-20," 2012.

 Tarun Goyal obtained a Bachelor of Engineering in Electronics & Communication Engineering from University College of Engineering, Rajasthan Technical University, Kota INDIA and Master of Technology in Embedded Systems from MNIT Jaipur in 2016. He is currently working in Western Digital at Bengaluru INDIA. He has earlier served in Aricent Inc. and Mirafra Technologies, both in Bengaluru. His research interests are in the areas of VLSI Design especially design for test(DFT), Cryptography.

 V. Sahula (M'92-SM'04, IEEE) obtained his Bachelors in Electronics (honors) from Malaviya National Institute of Technology, Jaipur, India in 1987 and Masters in Integrated Electronics & Circuits as well as Ph.D. Degree from Department of Electrical engineering, Indian Institute of Technology, Delhi (IITD) in 1989 and 2001, respectively. In 1990, he joined Malaviya National Institute of Technology, Jaipur as faculty member, where he is currently Professor in the Department of Electronics and Communications Engineering. He has 100+ research papers in reputed journals and conference proceedings to his credit. His research interests are into Trust, Integrity & Resilience in Hardware; machine learning & cognition modeling.

Dr. Sahula has served on the Technical programme committees of the VLSI Conference and VLSI Design and Test Symposium, India from 1998 to 2019 and as reviewer of many journals from ACM and IEEE. He has also served on organizing committee as fellowship-chair of Embedded Systems Week in 2014 & of 22nd IEEE International Conference on VLSI Design, India in 2009. He is a Senior member of IEEE, Fellow of IETE and IE, and member of ACM SIGDA, IMAPS and IET.

 Deepak Kumawat obtained a Bachelor of Technology in Electronics & Communication Engineering from Rajasthan Technical University, Kota INDIA and Master of Technology in VLSI Design from MNIT Jaipur in 2016. He is currently working as an entrepreneur running a Wireless Home Automation business. His research interests are in the areas of Lightweight Cryptography, system level design and modeling.