Towards Light Weight Cryptography Schemes for Resource Constraint Devices in IoT

Santosh Pandurang Jadhav

Faculty of Telecommunications, Technical University of Sofia, Bulgaria
E-mail: spjadhav375@gmail.com

Received 28 August 2019; Accepted 07 November 2019;
Publication 29 January 2020

Abstract

The Internet of Things (IoT) is becoming the most relevant next Internet-related revolution in the world of Technology. It permits millions of devices to be connected and communicate with each other. Beside ensuring reliable connectivity their security is also a great challenge. Abounding IoT devices have a minimum of storage and processing capacity and they usually need to be able to operate on limited power consumption. Security paths that depend maximum on encryption are not good for these resource constrained devices, because they are not suited for performing complicated encryption and decryption tasks quickly to be able to transmit data securely in real-time. This paper contains an overview of some of the cryptographic-based schemes related to communication and computational costs for resource constrained devices and considers some approaches towards the development of highly secure and lightweight security mechanisms for IoT devices.

Keywords: Constrained devices, cryptographic algorithms, IoT, lightweight cryptography, signcryption.

Journal of Mobile Multimedia, Vol. 15_1&2, 91–110.
doi: 10.13052/jmm1550-4646.15125

1 Introduction

In near future, the Internet of Things (IoT) will be an essential element of our daily lives. Numerous energy constrained devices and sensors will continuously be communicating with each other, the security of which must not be compromised. Even nowadays IoT with its millions of unsecured devices is in no way immune to attacks. Such attacks can compromise gateways and deeper levels of IoT networks and disturb their performance, paralyze infrastructures, make systems fail and even put human lives in jeopardy. Such attacks could be classified as physical, network, software and encryption attacks [1, 2].

In physical attacks the system is accessed through proximity, for example inserting a USB drive. Such tampering with a system or network can enable the attacker to take over its control and extract data, or corrupt the system performance with a malicious code that opens a backdoor without being noticed, or force the system to get shut down by a distributed denial of service (DoS) attack. The Owlet WiFi Baby Heart Monitor [1] susceptibility is one case which shows how devices with the best of acceptation, such as adjusting parents when their babies experience heart troubles, can be dangerous if taken advantage of by a dishonest party. The connectivity element makes them vulnerable and manufacturers and developers should take extra steps to secure devices at the hardware layer. There are also physical attacks which target the energy consumption and in such battery drains even if the system is in sleep mode.

In the network type of attacks, the attackers attempt to listen to what is flowing through the network by inserting themselves between the user and the device which is also called “Man in Middle attack”. The goal is stealing passwords, creating fake identification and diverting the network packets to the desired locations for data analysis. Monitoring and eavesdropping, node subversion, node malfunctions, replication attack are different types of attacks on IoT Networks. The jeep hacking was a type of attack which exploits firmware update vulnerability. In this attack they hijacked the Jeep vehicle over the Sprint cellular network and were able to discover that they could make it speed up, slow down and even drive it as they wish. It was one of the proofs of the various emerging IoT attacks. Manufacturers and developers often ignore the security of peripheral devices or networks, and thus the consequences can be dangerous [1].

When some malware is installed into a network program or any malicious software sends a virus, or corrupts the data, or keeps watch on performing activities, these can be classified as software attacks. The mirai botnet was the largest Distributed DoS attack ever happened and was launched on the service provider using an IoT botnet. The IoT botnet was made by malware called Mirai. Once the devices and computers were infected with Mirai, they continuously search the internet for affected IoT devices and then use known default user identities and passwords to log in, affecting them with malware. Trojan horse, worm, logical bombs are types of software attacks [1].

Encryption attacks directly affect the heart of the algorithmic system. The attacker tries to find the encryption keys and if succeeds learns how the algorithms were developed. Usually in such cases attackers develop their own algorithms with the goal of installing them and taking control of the system.

Various security mechanisms are developed in order to handle such types of attacks. But many IoT devices have to be operated on minimum power and hence, with less storage capacity, memory, and processing capability, i.e. they are resource-constrained devices. In such cases the application of most of the standard security mechanisms can easily fail and other "light weight" cryptographic approaches should be introduced which use less memory and power and are highly secure.

Further this paper is organised as follows. Point 2 is an introduction to lightweight cryptography and its variants along with an overview of various lightweight ciphers. In Point 3 a general metric for the performance analysis of lightweight algorithms is presented. In Point 4 some aspects of future work are considered and the last point concludes the paper.

2 Lightweight Cryptography Algorithms

There are conventional methods of cryptography such as the Advanced Encryption Standard (AES), Secure Hash Algorithms (SHA-265), Rivest–Shamir–Adleman (RSA) and Elliptic curve cryptography (ECC) which work on systems that have abundant computational power and memory capacities, but they cannot perform well with embedded systems and sensor networks. Therefore, lightweight cryptography methods are proposed to overcome many of the problems of conventional cryptography when applied to systems having constraints related to physical size, computational requirements, limited memory, and energy consumption. Lightweight cryptography permits the application of secure encryption for devices with limited resources.

Many efforts are made to develop lightweight cryptography algorithms along with higher security [3]. In communication systems encryption is already deployed as standard on the data link layer. But in many cases, encryption in the application layer is more effective as it is giving an end to

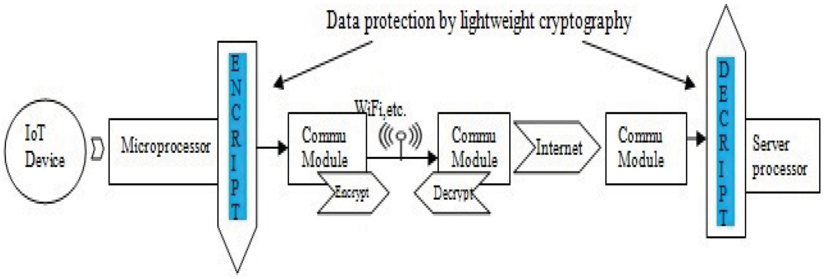

Figure 1 Implementation of lightweight cryptography at the application layer.

end data protection from the device to the server and allows for implementing security independently from the type and structure of the communication system. The encryption mechanism should be applied at the processor processing the application as shown in Figure 1 and on the available resources, therefore, it should be as lightweight as possible.

In resource constrained IoT devices having limited computing speed, low power backups and smaller size (circuit size, ROM/RAM sizes), the power is greatly dependent on the hardware such as the circuit size or the processor in use. Thus the size becomes the reference point for the lightness of the encryption method and also for the power consumption. The latter is dependent on the computing speed and execution time, so the number of computations that determines the processing speed usually is considered to be an index of the lightness of the algorithm. The throughput of any cryptographic system is calculated as the average of total plain text in a number of *k* bytes divided by the average encryption time and in the case of decryption, throughput is calculated as the average of total cipher text divided by the average decryption time.

Since encryption is a technology for securing the overall system the lightweight cryptography needs to adopt a method that is evaluated as ensuring sufficient security in the light of modern cryptography. The International Standards Organization (ISO) and the International Electrotechnical Commission (IEC) are maintaining standards about information and communication technology. According to these standards the algorithms to be considered as "light weight cryptography" should follow requirements related to [4]:

- **Security strength for lightweight cryptography.** The minimum is 80-bit, but is suggested that at least of 112-bit security should be applied for systems that will give security for maximum longer periods.

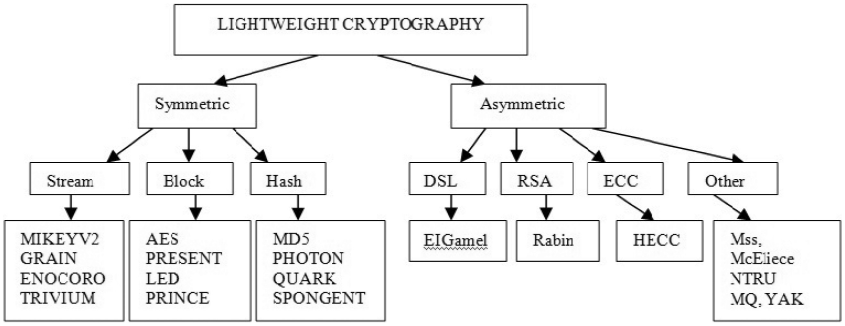

Figure 2 Types of lightweight cryptography.

- **Hardware implementation properties.** The chip area covered by the cryptographic mechanism and the energy consumption should be less compared to existing ISO standards.
- **Software implementation properties.** The code size and required RAM size should have fewer resource requirements than in existing standards for the same platform.
- The generality of the lightweight properties claimed for the cryptographic mechanism.

2.1 Classification of Lightweight Cryptography Algorithms

A classification of lightweight cryptographic algorithms is shown in Figure 2. There are two major types of cryptography algorithms: symmetric and asymmetric.

2.1.1 Symmetric encryption

Symmetric encryption uses the same key for both encryption and decryption of data. This method of encryption is secure and relatively faster. The major drawback of symmetric key encryption is the sharing of the key between the two communicating parties. An attacker can decrypt the data if he has access to the key. Symmetric key algorithms assure the confidentiality and integrity of data but do not guarantee authentication. This type of encryption uses three types of algorithms based on hashing, stream and block ciphers.

- **Hashing**: This method is based on producing a "hash" with a private key that can be verified with a public key. For lightweight cryptography, PHOTON [5] Spongent [6] and Lesamanta LW [7] are defined as standards for hashing methods within ISO/IEC 29192-5:2016. Conventional crypto hash functions for MD5 and SHA1 and other modern hash methods are not convenient for IoT devices. National Institute of Standards and Technology (NIST) has thus recommended new hashing methods such as SPONGENT, PHOTON, Quark, and Lesamnta-LW. These methods generate a much smaller memory footprint and have an input of just 256 characters (whereas conventional hash functions have up to 264 bits). Chaskey Cipher is a permutation based lightweight cryptography method for signing messages (MAC) using a 128-bit key. The Chaskey takes a 128-bit block using a 128-bit Addition-Rotation-XOR-based permutation [8].

- **Streaming:** A stream cipher is a symmetric key cipher in which plaintext digits are combined with a pseudorandom cipher digit stream. In this type each plaintext digit is encrypted one at a time with the analogous digit of the keystream, to give a digit of the cipher text stream as a result. Mickey V2 is a lightweight stream cipher and was written by Steve Babbage and Matthew Dodd. It creates a key stream from an 80-bit key and a variable length initialization vector (of up to 80 bits). The keystream has a maximum length of 240 bits [9]. Trivium is also one lightweight stream cipher and it was developed by Christophe De Canniere and Bart Preneel and has a low footprint for hardware. It uses an 80-bit key and generates up to 264 bits of output, with an 80-bit IV [10]. Grain and Enocoro are the Light Weight Stream Ciphers which have 80 bit and 128-bit key respectively. Grain has relatively low power consumption and memory [11]. Ecarno is defined by Hitachi and is included in ISO/IEC 29192 International Standard for a lightweight stream cipher method.
- **Block:** A block cipher is an encryption method that applies a deterministic algorithm along with a symmetric key to encrypt a block of text, rather than encrypting one bit at a time as in stream ciphers. PRESENT and CLEFIA for block methods are defined as standards for lightweight cryptography within ISO/IEC 29192-2:2012. One of the first to show promise for a replacement for AES for lightweight cryptography is PRESENT [12] It operates on 64-bit blocks and uses a substitution-permutation method. CLEFIA is a well known lightweight block cipher was defined by Sony and has 128, 192 and 256-bit keys and 128-bit block sizes.

Many lightweight cryptography algorithms were developed among them several symmetric algorithms use AES (Advanced encryption standards) as a

Table 1 Comparison of various symmetric lightweight ciphers

Ref. No	Algorithm	Keysize	Block size	Merits
[13]	AES	128	128	Supports larger key sizes. Faster in both hardware and software.
[14]	PRESENT	80/128	64	Ultra lightweight cipher. Energy efficient.
[15]	RECTANGLE	128	64	Fast implementations using bit-slice techniques.
[16]	HIEGHT	128	64	Ultra-lightweight. Provides high security. Good for RFID tagging.
[17]	CAMELLIA	128	128	Resistance to brute force attack on keys. Security levels comparable to AES.
[18]	TWINE	80/128	64	Good for small hardware. Efficient software performance.
[19]	SIMON	128	128	Supports several key sizes. Performs well in hardware.
[19]	SPECK	128	128	Performs better in software.
[20]	XTEA	128	64	Works based on network
[21]	KTANTAN	80	32/48/64	Very efficient hardware-oriented block cipher algorithm.
[22]	TREYFER	64	64	Aimed at smart card applications. Extremely simple algorithm.
[23]	Lilliput	80	64	Reduces the delay and increases the speed of operation.
[24]	PRINCE	128	64	Overhead for decryption on top of encryption is negligible.

standard. Table 1 shows some of the different symmetric algorithms and their merits. Examples of lightweight algorithms with some industry applications are given in Table 2.

2.1.2 Asymmetric encryption

Asymmetric cryptography is a cryptographic system that utilizes two types of keys; public keys that may be distributed widely and private keys which are known only to the owner. The generation of the public keys depends on cryptographic algorithms based on one way mathematical functions. Thus the public key can be openly distributed without compromising security as

Table 2 Various lightweight ciphers used for industry applications

Ref. No	Algorithm	Applications
[25]	A5/1	2G GSM protocol still uses this algorithm.
[26]	Atmel Ciphers.	Stream ciphers used by the secure memory, crypto Memory and CryptoRF families of products from Atmel.
[27]	Crypto-1.	Stream cipher used by the Mifare classic line of smartcards.
[28, 29]	Css	Content of DVD discs is encrypted by the content scrambling system.
[30]	Dsc.	Stream cipher used to encrypt the communications of cordless phones.
[31]	Hitag2 Megamos.	Stream ciphers used in the car immobilizers implemented by different car manufacturers.
[32]	Kindle Cipher (PC1).	Amazon used it at least up until 2012 for the DRM scheme protecting its e-book using the Mobi file format.
[33]	Oryx.	Stream cipher Oryx was chosen by the telecom industry association standard (tia) to secure phone communications in North America.
[34]	CMEA	This block cipher was used by the telecom industry association standard to secure the transmission of phone numbers across telephone lines

for achieving effective of security the requirement is keeping the private key private [35]. In such this type of systems, any person can encrypt a message using the receiver's public key, but the encrypted message can only be decrypted with the receiver's private key. Asymmetric lightweight cryptography algorithms are highly recommended for devices with resource limitations. Asymmetric ciphers are computationally far more demanding than their symmetric counterparts. There are conventional asymmetric algorithms such as Rabin/RSA which is based on integer factorization problem, ECC/HECC which are based on Elliptic Curve Discrete Logarithm Problem.

The Rivest–Shamir–Adleman (RSA) is the most popular algorithm for asymmetric cryptography which supports key sizes from 1024 to 4096 bits. As such, it is well known for the various public key cryptosystems that researchers propose. But, its large hardware footprint and its resource demanding implementations guide researchers go for other algorithms which

Table 3 Comparisons of key sizes in bits of ECC and HECC

Security level	Elliptic curve	HECC Genus 2	HECC Genus 3
256	94	47	32
512	128	64	43
1024	174	87	58
2048	234	117	78
4096	313	157	105
8192	417	209	139

are more convenient for applications in constrained devices [36]. Rabin is somewhat similar to RSA. There is one main difference in the complexity of the factorization problems that they depend upon. The encryption for Rabin is faster than RSA. But, the decryption is less efficient. BluJay [37] is a hybrid Rabin-based scheme that is suitable for lightweight platforms and is based on WIPR and Hummingbird-2. The encryption by BluJay is significantly faster and more lightweight than RSA and ECC for the same level of security.

Elliptic Curve Cryptography (ECC) and its counterpart HECC (Hyper Elliptic Curve Cryptosystem) HECC have been considered as one of the best suits for power constraint devices in embedded systems. It is being observed that as security increases the key size of the conventional asymmetric algorithm such as RSA grows much faster as compared to ECC. ECC and HECC are more suitable for devices which need lightweight cryptography due to their lesser resources such as memory, computational power, and energy. HECC is a generalization of elliptic curves. As the genus increases, the arithmetic of encryption gets more and more complex, but it needs fewer bits than ECC for the same level of security. As HECC's operand size is smaller, it is considered to have better performance than ECC and to be more attractive for applications in resource-constrained devices. Example of an application requiring smaller operand and less computational power is E-commerce where lightweight cryptography is needed in order to make faster transactions [38]. In Table 3a short comparison of ECC and HECC algorithms is presented [39, 40].

An example of algorithm based on the Discrete Logarithm Problem (DLP) in Finite Fields is ElGamal [41]. This algorithm is of less interest for application in resource constrained platforms because its computation is more intensive than RSA and the result of the encryption from plaintext results in an increase of two times in the size of the ciphertext. It is also considered vulnerable to some types of attacks, like chosen encryption attacks.

There are some application specific asymmetric algorithms which are known for their performance and their resistance to quantum computer based decryption approaches, such as MSS, NTRU, McEliece, MQ, YAK. Merkle Signature Scheme (MSS) is a hash based cryptography and uses typical AES based hash functions. It is popular because of its smaller code size and faster verification process them RSA and ECC [42]. NTRU is one of the open source public key cryptosystems that utilizes lattice-based cryptography for encryption and decryption of data. NTRU was patented but was placed in the public domain in 2017. It has two algorithms NTRUEncrypt, for encryption, and NTRUSign, for digital signatures. Like other prominent public key cryptosystems, it is also resistant to attacks using Shor's algorithm and its performance is significantly better. Regarding the performance of NTRU in equivalent cryptographic strength, it should be noted that NTRU executes costly private key operations much faster than RSA. Performance time of RSA private operation increases as the cube of the key size, while as that of an NTRU operation increases quadratically [43]. NTRU provides the same level of security comparable to RSA and ECC and therefore is highly efficient and suitable for embedded system. RSA is 200 times slower in key generation and almost 3 times slower in encryption and about 30 times slower in decryption as compared with NTRU. The drawback of NTRU is that it produces larger output, which may lower the performance of the cryptosystem if the number of transmitted messages is complex and crucial but it is safe when it is implemented when the recommended parameters are used [44].

McEliece is an asymmetric algorithm which was not largely accepted because of its larger public and private key matrices as compared to RSA. The encryption and decryption are faster than RSA. McEliece was not used to produce signatures, but the signature has been constructed based on Niederreiter scheme which is a variant of McEliece. From a security point of view, Niederreiter provides the same security level as McEliece [45]. MQ requires 9690 bytes for the public key and 879 bytes for the private key and is based on the problem of solving multivariable quadratic equations over finite fields. It is commonly accepted that multivariate cryptography is more successful for building signature schemes basically because multivariate schemes give the shortest signature among quantum resistant algorithms [46].

In 2010 Feng Hao proposed the public key authenticated key agreement protocol YAK. Like other protocols, YAK normally requires a public key infrastructure to distribute authentic public keys to the parties involved in the communication. YAK provides different security properties. One is the private key security in which an attacker cannot learn the user's static private key even though if all session-specific secrets in any compromised session is known. Another is full forward secrecy in which session keys that were securely settled in the past uncorrupted sessions will remain incomputable in the future even when both users' static private keys are not closed. In such types of key security the attacker cannot compute the session key even if he acts like a user but has no permission to the user's private key [47].

2.2 Signcryption

There are various mechanisms for lightweight cryptography algorithms effectively to address the issues with resource constraint devices. Signcryption is one of them. This is a new public key cryptographic primitive that executes the functions of digital signature and encryption in a single logical step and with a cost lower than that required by the traditional approach of signature followed by encryption [48]. The biggest advantage of signcryption over the application of signature followed by encryption is the reduction of computational cost and communication overhead which can be shown by:

$$\textbf{\textit{Cost (signcryption)}} < \textbf{\textit{Cost(signature) + Cost(encrypt)}}$$

Many signcryption schemes that have been proposed are based on HECC. The Authors in [49] propose a signcryption scheme which greatly improves efficiencies for software and hardware applications but could not address forward secrecy and public verification. In [50] an efficient HECC based encryption scheme is developed which saves computational time and communication overhead up to 40% due to small key size, but is missing to show forward secrecy, authentication and availability properties.

A signcryption scheme with forwarding secrecy based on HECC that provides functionality and public verifiability which is more suitable for resource constraint devices is proposed in [51], but it uses zero-knowledge protocol for public verifiability. In [52] a public verifiable signcryption scheme with forwarding secrecy based on HECC is considered. There in public verifiability, a third party can verify the authenticity of the sender without cracking the confidentiality and knowing the private key of the receiver. In this case, the third party just needs the signcrypted text and some additional parameters. A mathematical model of public verifiability property is not given the paper.

Approach to smart card resistance as well as to offline password guessing attack in proposed in [53], based on secure signcryption on HECC with sensor-based random number. Limitations related to the generation of random

Table 4 Coefficients for general metric

Platform	Performance metric	α	B	λ	μ	τ
Hardware	Area/bit	1	0	0	0	0
	Th = N_B/T_B	0	−1	0	−1	0
	Th/A	−1	−1	0	−1	0
	FOM= Th/GE^2	−2	−1	0	−1	0
	Th /(A * E_b)	−1	−1	−1	−2	0
	Power	0	−1	1	0	0
	Energy per bit	0	0	1	1	0
Software	Cycles/block	0	0	0	0	1
	Through output	0	−1	0	−1	0
	Code size * cycle count/block size	1	0	0	1	1

numbers is one drawback of this scheme. In [54] an implementation of HECC based signcryption approach is described, but it is not implemented for resource constrained devices.

3 General Performance Metric for Lightweight Cryptography Algorithms

Considering different works in the literature related to the performance analysis of different lightweight cryptography algorithms the authors in [55] come up with a generalized metric that attempts to allow performance comparison of lightweight algorithms by presenting a uniform formula to represent current and future performance metrics. Such a general metric for hardware design is given by:

$$\textbf{\textit{General Metric}}(\boldsymbol{A^{\alpha}, T_B^{\beta}, E_B^{\lambda}, C_B^{\tau}, N_B^{\mu}}) = \frac{\boldsymbol{A^{\alpha}, T_B^{\beta}, E_B^{\lambda}, C_B^{\tau}}}{\boldsymbol{N_B^{\mu}}},$$

where: *A* is the area; T_B the time to encrypt one block; *E* is the energy; C_B is the number of cycles to encrypt one block; N_B is the block size; α β λ τ and μ are power coefficients. This general metric includes respective coefficients representing different performance metrics. By stating the appropriate value of these coefficients all the different metrics related to the performance of the various cryptographic ciphers could be covered. In Table 3 the values of the different coefficients used to derive different performance metrics by using the general metric are given.

The retrieving of different performance metrics using the values of the coefficient given in Table 4 can be illustrated by the determination of the so-called Figure of Merit (FOM).

$$\textbf{\textit{FOM}} = \textbf{\textit{General metric}} = \boldsymbol{A^{-2}, T_B^{-1}, E_B^{-1}, C_B^{0}, N_B^{-1}}$$

This shows that any performance metric can be derived without any confusion and ambiguity. As an example of performance efficiency the throughput per area (Th/A) can be derived as:

$$\textbf{\textit{Th/A}} = \textbf{\textit{General metric}} = \boldsymbol{A^{-1}, T_B^{-1}, E_B^{0}, C_B^{0}, N_B^{-1}}.$$

For a HECC based signcryption scheme that has various major operations such as hyperelliptic curve divisor multiplication (HECDM), Encryption and Decryption, inverse division, addition, subtraction, and key hash operations, the computational cost is calculated by the time required for the execution of these major operations. For example using an implementation platform with a pc running on c# with 1.70 GHz processing speed and intel i3 CPU, 4 GB RAM capacity and OS WINDOWS 7, the time for the encryption followed by signature approach for a 5 kb and 10 kb text file is noted to be 276 ms and 293 ms respectively. For signcryption using HECC and SHA-256 it is found to be 246 ms and 269 ms respectively. This shows that the cost of signcryption is less than the cost of the approach of signature followed by encryption. In this sample case considering the general metric for software, the calculated throughput is:

$$\text{Throughput} = \frac{T_B^{\beta} C_B^{\tau}}{N_B^{\mu}} = \frac{256^{-1}}{5^{-1}} = 0.019 \text{ block/ms}$$

In this way it is possible to find the efficiency and performance of different lightweight cryptography algorithms and their lightness in respect to application for securing resource constraint devices to be evaluated.

4 Future Work

Recently many lightweight cryptographic algorithms had evolved in securing the resource constraint devices in IoT. ECC and HECC combined with signcryption had shown the remarkable results in computational cost and energy consumption. There is a generalized signcryption algorithm for low computing devices which is proposed in [56]. This general approach which uses

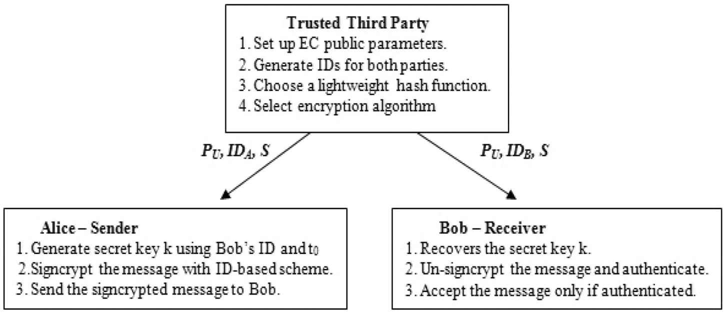
Trusted Third Party
1. Set up EC public parameters.
2. Generate IDs for both parties.
3. Choose a lightweight hash function.
4. Select encryption algorithm
P_U, ID_A, S
P_U, ID_B, S
Alice – Sender
1. Generate secret key k using Bob's ID and t_0
2.Signcrypt the message with ID-based scheme.
3. Send the signcrypted message to Bob.
Bob – Receiver
1. Recovers the secret key k.
2. Un-signcrypt the message and authenticate.
3. Accept the message only if authenticated.

Figure 3 General approach for lightweight cryptography.
(P_u – Public parameters, ID_A – Identity of Alice, ID_B – Identity of Bob, S – Selected Algorithms for hash and encryption, t_0 – Timestamp)

all the security features can be used in developing lightweight cryptographic mechanisms for devices with little computing power.

In Figure 3 the general approach towards lightweight cryptography is illustrated, where three parties are involved in the system during communication. The sender and receiver need to share their public parameter (Pu), their identities (ID) and choose among several available algorithms (S) for hash generation, encryption, and decryption. Along with this signcryption helps in reducing the time required for the total cryptographic process which makes the scheme more efficient.

In future, scope for researchers after referring the general approach is to implement zero knowledge protocol for verification purpose for signcryption using lightweight cryptography algorithm HECC. Which can improve the security level of the system while the use of lightweight encryption algorithm and signcryption scheme will make the system having resource constraint devices more efficient and secure.

5 Conclusion

From the presented in this paper survey and analysis of lightweight cryptography algorithms, it can be concluded that there is a need for more secure efficient lightweight cryptography mechanisms that could be suitable for implementation in resource constraint devices for IoT. Such mechanisms should have lower computational cost, lower power consumption and should provide same or higher level of security. On solution is the use of different schemes such as signcryption, zero knowledge protocol along with secure and lightweight cryptography protocol which can solve some of the major issues of security as well as allow the development of highly efficient security mechanism for low resource constraint devices in IoT.

References

[1] https://www.iotforall.com/5-worst-iot-hacking-vulnerabilities/

[2] Sattar B. Sadkhan, Akbal O. Salman. "A Survey on Lightweight-Cryptography" 2018 International Conference on Advances in Sustainable Engineering and Applications (ICASEA). pp. 105–108. 2018.

[3] Shamsher Ullah, Xiang-Yang Li, Lan Zhang. "A Review of Signcryption Schemes Based on HyperElliptic Curve" 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM). pp. 51–58. 2017.

[4] Biryukov, Alex and Leo Perrin. "State of the Art in Lightweight Symmetric Cryptography." IACR Cryptology ePrint Archive P. 511. 2017.

[5] Guo J, Peyrin T, Poschmann A. "The PHOTON family of lightweight hash functions Springer, vol 6841. pp. 222–239. 2011.

[6] Bogdanov A, Knezevic M, Leander G, et al. "{SPONGENT}: the design space of lightweight cryptographic hashing" IACR Cryptology ePrint Archive, 2011.

[7] Hirose S, Ideguchi K, Kuwakado H, et al. A lightweight 256-bit hash function for hardware and low end devices Lesamnta-LW. Berlin, Heidelberg. pp. 151–168. Springer 2011.

[8] Buchanan WJ, "Chaskey Cipher." [Internet]. Available from: http://asecuritysite.com/encryption/chas.

[9] Buchanan WJ, "Mickey V2 lightweight stream cipher." [Internet]. Available from: http://asecuritysite.com/encryption/mickey.

[10] Buchanan WJ, "Trivium lightweight stream cipher." [Internet]. Available from: http://asecuritysite.com/encryption/trivium.

[11] 9WJ, "Grain lightweight stream cipher." [Internet]. Available from: http://asecuritysite.com/encryption/grain.

[12] Bogdanov A, Knudsen LR, Leander G, et al. "PRESENT: An ultra-lightweight block cipher" vol 4727. pp. 450–466. 2007.

[13] A. Moradi et al., "Pushing the Limits: A Very Compact and a Threshold Implementation of AES," in Advances in Cryptology – EUROCRYPT 2011 Lecture Notes in Computer Science, vol. 6632, pp. 69–88. Springer, 2011.

[14] A. Bogdanov et al., “PRESENT: An Ultra-Lightweight Block Cipher, in Cryptographic Hardware and Embedded Systems - CHES 2007 Lecture Notes in Computer Science, pp. 450–466. Springer, 2007.

[15] W. Zhang et al., RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms,” in Science China Information Sciences, vol. 58(12), pp. 1–15. 2015.

[16] D. Hong et al., “HIGHT: A New Block Cipher Suitable for Low-Resource Device,” in Cryptographic Hardware and Embedded Systems – CHES 2006 Lecture Notes in Computer Science, pp. 46–59. 2006.

[17] A. Satoh and S. Morioka, “Hardware Focused Performance Comparison for the Standard Block Ciphers AES, Camellia, and Triple-DES,” in Lecture Notes in Computer Science Information Security, pp. 252–266. Springer, 2003.

[18] T. Suzaki et al., “TWINE: A Lightweight Block Cipher for Multiple Platforms,” in *Selected Areas in Cryptography Lecture Notes in Computer Science*, vol. 7707, pp. 339–354. Springer, 2013.

[19] R. Beaulieu et al., “The SIMON and SPECK lightweight block ciphers, in Proceedings of the 52nd Annual Design Automation Conference, pp. 1–6. 2015.

[20] R. M. Needham and D. J. Wheeler. Tea extensions. Technical report, Cambridge University, Cambridge, UK, October 1997.

[21] Christophe De Cannière, Orr Dunkelman, and Miroslav Knežević. KATAN and KTANTAN – a family of small and efficient hardware-oriented block ciphers. In Christophe Clavier and Kris Gaj, editors, Cryptographic Hardware and Embedded Systems – CHES 2009, volume 5747 of Lecture Notes in Computer Science, pp. 272–288. Springer, Heidelberg, September 2009.

[22] Gideon Yuval. Reinventing the Travois: Encryption/MAC in 30 ROM bytes. In Biham [Bih97], pp. 205–209, 1997.

[23] Thierry Pierre Berger, Julien Francq, Marine Minier, and Gaël Thomas. Extended generalized Feistel networks using matrix representation to propose a new lightweight block cipher: Lilliput. IEEE Transactions on Computers, pp. 99, August 2015.

[24] Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezecic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçin. “PRINCE – A low-latency block cipher for pervasive computing applications” vol 7658, pp. 208–225. 2012.

[25] Ross Anderson. A5 (Was: HACKING DIGITAL PHONES). uk. elecom (Usenet), https://groups.google.com/forum/?msg/uk.telecom/TkdCaytoeU4/Mroy719hdroJ#!msg/uk.telecom/TkdCaytoeU4/Mroy719hdroJ, June 1994.

[26] Flavio D. Garcia, Peter van Rossum, Roel Verdult, and Ronny Wichers Schreur. Dismantling Secure Memory, Crypto Memory, and CryptoRF. In Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS '10 pp. 250–259, New York, NY, USA, 2010. ACM.

[27] Karsten Nohl, David Evans, Starbug Starbug, and Henryk Plötz. "Reverse engineering a cryptographic RFID tag', In USENIX security symposium, volume 28, 2008.

[28] M. Becker and A. Desoky. "A study of the DVD content scrambling system (CSS) algorithm", In Proceedings of the Fourth IEEE International Symposium on Signal Processing and Information Technology, pp. 353–356, Dec 2004.

[29] Lea Troels Møller Pedersen, Carsten Valdemar Munk, and Lisbet Møller Andersen. "Cryptography – the rise and fall of DVD encryption", Available online at http://citeseerx.ist.psu.edu/viewdoc/download;sessionid=3672D97255B2446765DA47DA97960CDF?doi=10.1.1.118.6103&rep=rep1&type=pdf. 2007.

[30] Stefan Lucks, Andreas Schuler, Erik Tews, Ralf-Philipp Weinmann, and Matthias Wenzel. "Attacks on the DECT authentication mechanisms", In Marc Fischlin, editor, Topics in Cryptology – CT-RSA 2009, volume 5473 of Lecture Notes in Computer Science, pp. 48–65. Springer, Heidelberg, April 2009.

[31] Roel Verdult, Flavio D Garcia, and Baris Ege. "Dismantling Megamos Crypto: Wirelessly lockpicking a vehicle immobilizer", In Supplement to the 22nd USENIX Security Symposium (USENIX Security 13), pp. 703–718. USENIX Association, August 2013.

[32] Alex Biryukov, Gaetan Leurent, and Arnab Roy. "Cryptanalysis of the "kindle" cipher". In Knudsen and Wu [KW13], pp. 86–103, August 2012.

[33] David Wagner, Leone Simpson, Ed Dawson, John Kelsey, William Millan, and Bruce Schneier. Cryptanalysis of ORYX. In Stafford E. Tavares and Henk Meijer, editors, SAC 1998: 5th Annual International Workshop on Selected Areas in Cryptography, volume 1556 of Lecture Notes in Computer Science, pp. 296–305. Springer, Heidelberg, August 1999.

[34] David Wagner, Bruce Schneier, and John Kelsey. Cryptanalysis of the cellular encryption algorithm. In Burton S. Kaliski Jr., editor, Advances in Cryptology – CRYPTO'97, volume 1294 of Lecture Notes in Computer Science, pp. 526–537. Springer, Heidelberg, August 1997.

[35] https://en.wikipedia.org/wiki/Public-key_cryptography.

[36] Oren, Y., Feldhofer, M.: WIPR - a low-resource public-key identification scheme for RFID tags and sensor nodes. In: Basin, D.A., Capkun, S., Lee, W. (eds.) WISEC, pp. 59–68. ACM 2009.

[37] Saarinen, M.-J.O.: The BlueJay ultra-lightweight hybrid cryptosystem. In: 2012 IEEE Symposium on Security and Privacy Workshops (SPW), 24–25 May 2012, pp. 27–32. 2012.

[38] Javed R. Shaikh et al (2017) 'Enhancing E-Commerce Security Using Elliptic Curve Cryptography', International Journal of Current Advanced Research, 06(08), pp. 5338–5342. DOI: http://dx.doi.org/10.24327/ijcar.2017.5342.0701

[39] Reza Alimoradi, "A Study of Hyperelliptic Curves in Cryptography" I. J. Computer Network and Information Security, 2016, 8, 67–72.

[40] Roman, R., Alcaraz, C., Lopez, J.: A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes. J. Mob. Netw. Appl. 12(4), 231–244, 2007.

[41] Gamal, T.E.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theor. 31(4), 469–472 (1985).

[42] Rohde, S., Eisenbarth, T., Dahmen, E., Buchmann, J., Paar, C.: Fast hash-based signatures on constrained devices. In: Grimaud, G., Standaert, F.-X. (eds.) CARDIS 2008. LNCS, vol. 5189, pp. 104–117. Springer, Heidelberg 2008.

[43] https://en.wikipedia.org/wiki/NTRU#Performance.

[44] Howgrave-Graham, N., Silverman, J.H., Whyte, W.: Choosing parameter sets for NTRUEncrypt with NAEP and SVES - 3. In: Menezes, A. (ed.) CT-RSA 2005. LNCS, vol. 3376, pp. 118–135. Springer, Heidelberg 2005.

[45] Shoufan, A., Wink, T., Molter, G., Huss, S., Strentzke, F.: A novel processor architecture for McEliece cryptosystem and FPGA platforms. In: Proceedings of the 20th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP 2009), pp. 98–105. 2009.

[46] Yang, B.-Y., Cheng, C.-M., Chen, B.-R., Chen, J.M. Implementing minimized multivariate PKC on low-resource embedded systems. In:

Brooke, P.J., Clark, J.A., Paige, R.F., Polack, F.A.C. (eds.) SPC 2006. LNCS, vol. 3934, pp. 73–88. Springer, Heidelberg (2006).

[47] https://en.wikipedia.org/wiki/YAK_(cryptography).

[48] https://en.wikipedia.org/wiki/Signcryption.

[49] X. W. Zhou, ”Improved Signcryption Schemes Based on Hyper-elliptic Curves Cryptosystem,” in Applied Mechanics and Materials, pp. 546–552, 2010.

[50] Nizamuddin, S. A. Chaudhry, W. Nasar, and Q. Javaid, “Efficient Signcryption Schemes based on Hyperelliptic Curve Cryptosystem,” IEEE International Conference on Emerging Technologies (ICET 2011), pp. 84–87, September 2011.

[51] A. S. Ch, Nizamuddin. M. Sher, G. Anwar, N. Husnain, and I. Azeem, “An efficient signcryption scheme with forwarding secrecy and public verifiability based on hyper elliptic curve cryptography,” Multimedia Tools and Applications, vol. 74, pp. 1711–1723, 2015.

[52] C. Ashraf, and M. Sher, “Public verifiable signcryption schemes with forward secrecy based on hyperelliptic curve cryptosystem,” in International Conference on Information Systems, Technology, and Management, pp. 135–142, 2012.

[53] J. Premalatha, K. Sathya, and V. Rajasekar, “Secure signcryption on hyperelliptic curve with sensor-based random number”, pp. 95–98.

[54] P. Kumar, A. Singh, and A. D. Tyagi, “Implementation of Hyperelliptic Curve Based Signcryption Approach”. International Journal of Scientific and Engineering Research, Vol. 4, Issue 7, 2013.

[55] Bassam J. Mohd, Thaier Hayajneh, Athanasios Vasilakos. “A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues. Journal of Network and Computer Applications, 58. pp. 73–93. 2015.

[56] Anuj Kumar Singh, B.D.K.Patro, “Performance Comparison of Signcryption Schemes – A Step towards Designing Lightweight Cryptographic Mechanism”. International Journal of Engineering and Technology (IJET) ISSN (Online) : 0975-4024. Apr–May 2017.

[57] Shamsher Ullah, Xiang-Yang Li, Lan Zhang. “A Review of Signcryption Schemes Based on Hyper Elliptic Curve” 2017 3rd International Conference on Big Data Computing and Communications. 978-1-5386-3349-6/17. 2017 IEEE.

Biography

Santosh Pandurang Jadhav is a Ph.D. student at the Technical University of Sofia at Sofia, Bulgaria since 2017. He received his B.E. in Information Technology Engineering from North Maharashtra University, India in 2007 and M.E. in Computer Science & Engineering from the Savitribai phule, Pune University of Maharashtra, India in 2012. As an Assistant Professor in NDMVPS's KBT college of engineering, Nashik, Maharashtra, India he has acquired a solid experience about 11 years of teaching in Information Technology Engineering.