Ad Hoc Networks 89 (2019) 97–106

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Light-weight hashing method for user authentication in Internet-of-Things

Vidya Rao, Prema K.V.*

Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, Karnataka, India

ARTICLE INFO

Article history:
Received 26 October 2018
Revised 1 February 2019
Accepted 5 March 2019
Available online 7 March 2019

Keywords:
Authentication
BLAKE2b
ECDSA
Hashing algorithms
Internet-of-Things
Raspberry Pi-3

ABSTRACT

The goal of Internet-of-Things (IoT) is that every object across the globe be interconnected under the Internet Infrastructure. IoT is expanding its application domain to range from environmental monitoring to industrial automation thereby leading to vast research challenges. Vast presence of devices in the Internet has increased the possible challenges faced by devices and data. The devices communicate on a public channel that is more likely to be accessed by unauthorized users and disturb the privacy of genuine users. The existing solutions that ensure data authenticity and user privacy, use MD5 and SHA-family of hashing algorithms under digital signature schemes. These algorithms create a trade-off between the security concern and energy consumption of IoT devices. To provide an energy efficient authentication method, we propose a customized BLAKE2b hashing algorithm with modified elliptic curve digital signature scheme (ECDSA). The parameters considered for the evaluation of the proposed methods are signature generation time, signature verification time and hashing time. The experiments are conducted under client server model using Raspberry Pi-3. The proposed method has shown about 0.7–1.91% improvement in the signature generation time and 7.67–9.13 % improvement in signature verification time when compared with BLAKE2b based signature generation/verification. The proposed method is resistant to Man-in-the-Middle attack, Distributed DoS attack (DDoS), pre-image resistance, second pre-image resistance and collision resistance. Based on the performance obtained by the experiments, it can be inferred that the proposed scheme is feasible for resource-constrained IoT devices.

1. Introduction

In an attempt to transform machine-to-machine communication to every object being communicated, the Internet-of-Things (IoT) has brought in the greatest revolution in the current digital era. Initially, the Internet was used only for online shopping, data exchange, playing games, but now the ability to connect every object around us to the Internet has made life much easier [1]. Accordingly, the IoT has formed the essential part of the socio-economic growth of the country by enabling various domain-specific applications ranging from smart homes to smart cities, food processing to smart farming retails to industrial automation and various environmental monitoring applications [2]. Hence in 2012, International Telecommunication Union (ITU-T) defined IoT as "a global infrastructure for the information society, enabling advanced services by interconnecting virtual and physical things based on existing and evolving interoperable information and communication technologies [3]".

As IoT is a world of self-organizing, intelligent and self-adaptive network, the wireless sensor network (WSN) acts as a digital skin that enables the sensor nodes to collect the physical environmental changes and communicate the processed data over the Internet. These sensor empowered devices are capable of sensing, actuating, data capturing, data storing and data processing [4,5]. These devices are classified based on the ability of communication as class-0 (devices too small to securely run on the Internet), class-1 (machines with more than 10KB of RAM and 100KB of code space)and class-2 (devices with about 50KB of RAM and 250 KB of code space) devices [6].

With the growth of connected devices under IoT, there is an increase in the potential vulnerability on security, privacy and governance [7]. Though IoT can make people's life convenient, it shall fail to ensure security and privacy of the user data and this may lead to a number of undesirable consequences. For example, in 2015 the IoT baby-monitors were hacked, through which the hackers were able to monitor the live feeds of the baby, change the

* Corresponding author.
E-mail addresses: vidyarao1988@yahoo.com (V. Rao), prema.kv@manipal.edu (P. K.V.).

https://doi.org/10.1016/j.adhoc.2019.03.003
1570-8705/© 2019 Elsevier B.V. All rights reserved.

camera settings and authorize other users remotely to view and control the baby-monitor [8]. One more such example was seen when intruders over-wrote the part of Ukraine power grid that caused the first cyber attack during 2017 [9]. Even the Internet-connected cars and wearable devices can also become a threat to the user's security and privacy. As these poorly secured IoT devices can serve as a means of an entry point for cyber attackers by allowing various malicious individuals to re-program them and cause malfunctioning. It becomes essential to provide the security and privacy at the device level. So to develop a safer IoT solution, it is essential to consider confidentiality, integrity and authentication properties (CIA-properties) [7].

Providing authentication for the user data and device has become a major challenge in IoT privacy and security system (IPS) [10]. IoT deployment depends on Low power Lossy Network (LLN) where the devices are having low power, limited memory and are highly dynamic. These LLNs also experience a high amount of data losses due to impersonation of nodes [11]. Traditionally, authentication is provided using Digital Signature Algorithms (DSA) based on Elliptic Curves [12]. Most of the DSAs use different hashing and public key crypto schemes. The commonly used hashing techniques are message digest (MD5) and variants of SHA family. These hashing techniques require more computations that can drain the resources of the devices. A stream cipher based hashing method called BLAKE [13,14] and BLAKE2 [15,16] have shown considerably less energy consumption and less computational cost in resource-constrained devices of IoT domain [17].

The proposed system contributes to a modified version of BLAKE2b called c-BLAKE2b (customized) along with elliptic curve cryptography (ECC) based digital signature algorithm for providing the authentication of the device to the server. A novel node registration phase is proposed by using the ECC based key generation algorithm. The proposed system has shown a considerable amount of good performance as compared to BLAKE2b.

This paper is organized as followed: Section 2 summaries the related work in the field of authentication and key generation with existing hashing algorithms. Section 3 explains the overall concept of elliptic curve cryptography and BLAKE2b. Section 4 generalizes the problem definition and system design. Section 5 provides a detailed explanation of cBLAKE2b and other authentication processes. Finally Sections 6 and 7 describe the performance analysis of the proposed system and the conclusion of the work respectively.

2. Related work

An uninterrupted and accurate functioning of IoT devices in smart city application is a crucial task. Such implementations have challenges to assure the authenticity of the devices to make better decisions. Hence to balance the performance between efficiency and communication cost in 2017, Nan Li et al. [18] designed a lightweight mutual authentication protocol based on public key encryption scheme for smart city applications. They have evaluated their work on a Contiki OS and CC2538 evaluation model. Even though their simulation showed a better performance against RSA and ECC, it is not realistically proved. This is because they used online and offline digital signatures that consume more time and create overhead on the node's resources.

In 2017, Kim et al. [19] designed an authentication and key management scheme (AKM) for IEEE 802.11ah based IoT communication. They used the existing security configurations of IEEE 802.11x. The design was able to delegate the burden of AKM processes to resourceful agents and these agents possessed powerful resources and performed cryptographic functions for IoT devices. They were able to show better performance in network overhead cost, memory usage and computation cost. As IoT is a dynamic network, their work could not show similar performance and also they failed to provide security at the device level storage.

In 2016, Jiye et al. [20] proposed a session key establishment based scheme for a clustered sensor network by using ECDH key exchange and hash chain. It showed resistance against session key attack, node impersonation attack, reply attack and node capture attack. It showed the main advantage of storing the past and future session keys in a repository. But on the real-time work model, these processes consumed device memory space and created overhead on the device.

In 2017, for a health care based IoT application Yuwen et al. [21] provided a solution to secure the patient's privacy. The devices placed over patient's body undergo many physiological movements and collect lot of data. Hence when a doctor wants to access this data through remote login, the patient has to authenticate the doctor. To enable the privacy of user identity, Yuwen et al. designed a scheme where a gateway knows the shared keys and these keys are shared using ECDHA to maintain key secrecy but they were unable to provide authentication at the gateway level that lead intruders into the network.

In 2016, Lee et al. [22] designed an energy-efficient authentication scheme for IoT environment. Their scheme ensures lightweight mutual verification and key exchange mechanism based on hashing and XOR process. This scheme had no restriction on operational quantity, velocity and storage space. Though their scheme met international security standards, they were unable to show the same efficiency for a real-time IoT environment which is more resource constrained.

In 2015, Chung et al. [23] proposed a novel anonymous authentication scheme that uses the virtual identification for the IoT devices by ensuring anonymity and authentication. By keeping the uniqueness and virtual identities, they are providing untraceability of devices. This scheme has helped them to withstand replay attack, forgery and impersonation attacks, they are using common and specific keys that make the system more complicated. Even if the common key is leaked or forged, it is difficult to find the specific key. Hence by using collision free one-way hashing and XOR operations, they are able to provide considerable fast computation.

In 2016, Asha et al. [24] implemented mutual authentication for RFID based applications using hybrid ECC (HECC). RFID tags are likely to experience attacks like eavesdropping, impersonation, cloning, tag destruction, unauthorized tag reading, tag modification attacks [25]. So to defend the devices against these attacks an ECC based secured authentication mechanism was developed by Asha et al. The hardness of solving hyper-elliptic discrete logarithm problem (HCDLP) provides security over eavesdropping from breaking into cryptosystem. They have used a D-Quark hashing algorithm. HECC is used to exchange the symmetric keys between the communicating parties. But their scheme was not suitable for non-RFID based IoT applications.

In 2018, Md. Wazid et al. [26] designed a secure authentication scheme for hierarchical IoT network (HIoTN). HIoTN is made up of different nodes, gateway nodes, cluster head node and sensing nodes arranged in a hierarchical manner. For such a network they have proposed a three-factor remote user authentication scheme for HIoTNs called user authentication key management protocol (UAKMP). UAKMP uses a smart card, password and personal biometric entities to provide a three-tier user authentication. Even though UAKMP is proved to provide security against known attacks through simulation, they were unable to show the same performance on a real-time based scenario with resource-constrained devices.

In a cognitive IoT architecture, there are security concerns over the radios, hence in 2018, Lin et al. [27] proposed a two-tier device based authentication schemes. This setup has helped them to explore the tradeoff between detection of the malicious node and spectrum management. But by developing a joint spectrum allocation and topology control, their system could be extended to real-time sensing through which they could reduce the end-to-end delay and control the network access.

In 2017, based on the challenge-response phase of physically unclonable functions (PUF) of IoT devices, Aman et al. [28] proposed a mutual authentication scheme for communication between a device and server and between two devices. The challenge-response method was also used for session key establishment. Even though their system showed improvement in the performance, the latency of authentication was more as the number of messages exchanged between the entities were increasing with every new session.

The common methods to provide user authentication are password, tokens or biometrics. But these methods also possess security issues. Hence in 2017, Jangirala et al. [29] introduced a new method called bio-hashing. Biohashing eliminates false acceptance rate without an increase in the occurrence of the false rejection rate. This scheme supports user-friendly password reset and dynamic node addition. Under BAN-logic, their scheme has shown mutual authentication between the nodes and also under AVISPA tool they were able to test for man-in-the-middle(MITM) attack [30] and replay attack. But the scheme possessed limitations due to lack of dynamic identities for large growing IoT network.

In 2015, Shivraj et al. [31] proposed a one-time-password (OTP) based security scheme to ensure end-to-end authentication between various IoT devices. They have used lightweight identity-based ECC scheme and Lamport's OTP algorithm. With the argument of having two-factor authentication scheme, their experimental proof has shown significantly better performance than the existing standby OTP algorithms. But they are unable to provide the same efficiency for a widely spread IoT network as generating OTP simultaneously for many devices is infeasible.

In 2016, Pardeep et al. [32] presented ECC-based Access Control Protocol(ACP) solution to prevent malicious nodes from eavesdropping the network and also to protect the node privacy. During the comparative study of their scheme, the time taken by ACP in computing point multiplication is much lesser when compared with existing protocols and energy consumed during the multiplication is also less. They have concluded that their method is feasible for access control and privacy of the node. But they have failed to address the data integrity of the wireless sensor network based application. But providing integrity on sensed data is the major concern of secured IoT system.

In 2009, Xuan et al. [33] discussed the issues of mutual authentication problem in mission-critical applications related to WSN. They present an ENergy-efficient Access control scheme Based on ECC (ENABLE). The performance of ENABLE is compared with HBQ[Enable-4] and symmetric key based schemes on SENSE simulator under the AODV protocol that showed ENABLE provided better scalability with lesser memory requirement and no key predistribution. As they have used online KDC, if the Internet goes down providing access control would be difficult.

3. Background

3.1. Elliptic Curve Cryptography

Public key cryptography (PKC) enables a strong authentication over data and maintain the confidentiality of user's privacy. Some of the PKC algorithms are, Rivest-Shamir-Adleman (RSA) [16,34,35], digital signature algorithm (DSA) [36,37], elliptic curve cryptography (ECC) [38,39], ECC using Diffie–Helman Algorithm (ECDHA) [40] and ECC using Digital Signature Algorithm (ECDSA) [41]. Among these algorithms, ECC is considered a light-weight cryptographic solution for various resource-constrained devices. IEEE have published PKC standard specification P1363 which describes the implementation of elliptic curve operations [42]. Also, NIST has provided different types of elliptic curves that can be used for ECDSA [43]. From Table 1, we can see that ECC has a longer protection period with a key size of 128-bit which is equivalent to the 4096-bits key-size of RSA [44].

Table 1
Key size and security level comparison [44].

RSA key size (bits)	ECC key size (bits)	Security level (bits)	Protection
1024	160	73	Short term
1536	192	89	Between short and long term
2048	224	103	Between legacy standard level and medium
4096	256	128	Long term

Table 2
Elliptic Curve Domain Paramters [46].

Domain parameters	Description
p	Prime number p of field F_p
a, b	Elliptic curve parameters
G	Generator of the field
n	Order of G
H	Co-factor h = E(F_q)/n

ECC has shown higher performance with smaller key size [38,45] and is compatible with resource-constrained devices due to its low energy consumption. Basically, ECC is defined over elliptic curve E which is termed as Weierstrass Eq. (1), where a, b, c, d and e are real numbers [45].

$$E : y^2 + axy + by = x^3 + cx^2 + dx + e \tag{1}$$

The strength of ECC depends on Elliptic Curve Discrete Logarithm Problem (ECDLP) [45] i.e., defined by considering simplified cubic form of Eq. (1) as $E : y^2 = x^3 + ax + b$, where $4a^3 + 27b^2 \neq 0$, and for given two points S and T on the elliptic curve, it is easy to calculate $T = k * S$, where k is a integer value and $*$ is a scalar multiplication. But it is infeasible to compute value of k for the given values of T and S. These elliptic curves operate on finite field F_q, where q can take either primary field (F_p) or binary field (F_{2^m}). In neither case, each of the users are distributed with standard domain parameters (Eq. (2)) as provided in SEC2 [46] and is summarized in Table 2.

$$T = (p, a, b, G, n, h) \tag{2}$$

3.2. Elliptic Curve Digital Signature Algorithm (ECDSA)

Elliptic Curve Digital Signature Algorithm (ECDSA) is an elliptic curve based digital signature algorithm that depends on ECDLP property of ECC [40]. It was proposed by Scott Vanstone during 1992 but was accepted by NIST in 1998. ECDSA obtained recognition as ISO standard (IS)-148883, an ANSI standard in 1999 as ANSI X9.62 and in 2000 it was accepted as IEEE standard (IEEE 1360–2000) and NIST FIPS-186-2. ECDSA has three phases: key generation, signature generation and signature verification [47].

3.3. Cryptographic hashing functions

Hash functions are being used in cryptology since many years for the applications like authentication, pseudo number generator for passwords, digital signatures and data integrity. A hash function is designed by constructing a fixed domain function called compression function and then iterating this compression function for numerous times to make a domain $\{0, 1\}^*$ [48]. These compression functions take an arbitrary length of the input string and convert it into a fixed length of output string after different iteration steps [38,49]. The iterative functions involve Merkle and Damgard (MD5) [50] used in SHA1, SHA-224, SHA-256, SHA-384 and SHA-512 as mentioned in [49], a wide pipe iterated hash design [49] used in SHA-224, SHA-348 and hash iterated framework (HAIFA) [51] used in LAKE [52], BLAKEx [13–15]. In our work, we are using the HAIFA model with the BLAKE2b algorithm.

3.3.1. HAsh Iterated FrAmework (HAIFA)

HAIFA was introduced by Biham and Dunklermann [51] to overcome the drawbacks of MD5 in-order to increase the flexibility and security. The compression function of HAIFA takes few more additional inputs like initial vector (*iv*), salt (*s*), number of bits hashed (*t*) and message block (m'), that can be represented as, f_{HAIFA}: $\{0, 1\}^{m_i}$ x $\{0, 1\}^{iv}$ x $\{0, 1\}^{s}$ x $\{0, 1\}^{t}$ using which a chain value (H_i) is derived as in Eq. (3):

$$H_i = f(H_{1-i}, M_i, t, s) \tag{3}$$

In HAIFA, if the length of the message is less than the block size, the message is padded with a string of zeros at the end. In other words, if we consider the length of the message as *M* and block size as *N*, then N-M number of zeros are added at the end of the message to make it a multiple of the block length. HAIFA hashing function hash shown that resistance against collision is indifferentiable from a random oracle. Also, it has shown good security resistance towards 2nd preimage attacks [53,54].

3.3.2. Customized BLAKE2b (c-BLAKE2b)

As BLAKE2b has shown improved performance here, we are using BLAKE2b as the basic algorithm and customizing it as c-BLAKE2b (CB2B) [13]. Primarily, the major difference between BLAKE and BLAKE2 is that, BLAKE2 uses few additional properties like parameter block, it has optional salt value and the rotation on *G* function which is optimized from 32, 25, 16 and 11 bits of BLAKE to 32, 24, 16 and 63 bits in BLAKE2. Also the padding of message is also removed [15]. Even though these changes improved the security concern, they have shown issues that are mentioned in [55]. First of all, the use of salt value at the IV rather than compression function has given way to multi-collision attacks. Secondly, use of parameter block has added to the computation of a large set of valid initial chaining values, this has increased collision attack as the attacker could generate different chain values for different messages and could map the values to break the message. Lastly, the removal of padding could not add any security concern as an attacker could infer the length of message using the counter value [55]. Due to these drawbacks, we are customizing BLAKE2b for resource-constrained devices by removing the parameter block which reduces 64 bytes computational load on the devices. Also, we are rearranging the *G*(*a*, *b*, *c*, *d*) function by using the diagonal elements through first four rotations and next four rotations are column elements which are reverse in BLAKE2b.

The BLAKE2b hash function uses the HAIFA [13] iteration mode which has compression function that depends on salt value (*s*) and a number of bits hashed so far, counter (*t*), to compress each message block with a distinct function. The structure of BLAKE and BLAKE2 versions are inherited by LAKE [13] as in Fig. 1: a larger inner state is generated from the initial value (*IV*), the counter (*t*) and the flag (*f*). Then the inner state is updated to message-depended rounds and finally, it is compressed to return the next chain value that is input for next message block to be compressed [15].

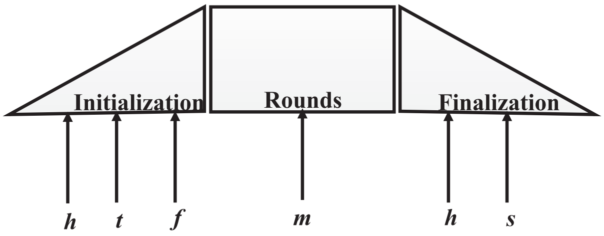

Fig. 1. Compression function of BLAKE2b.

Table 3
Notations table.

Symbols	Description
A_{ID}	Identity of node A
$DTime_i$	*i*th node deployment time
s_a	Random number between *n* and $DTime_A$ at node *A*
BS_{ID}	Identity of base station
P^A_{cur}	A_{th} node current available power
X_i	*i*th node private key
Y_i	*i*th node public key
X_i^{-1}	Inverse of *i*th node private key
M	Message
H	c-BLAKE2b hashing function
$H(M)$	Hash value of message *M*
K_a	Random per message key at A^{th} node
$(KG)_x$	x-component of *KG*
(sig_1, sig_2)	Signature pair

4. Problem definition and network model

4.1. Problem definition

In the IoT environment, the devices are connected over public channel called the Internet. Hence the communication is affected by security issues like user privacy and data authentication. To protect the network components and data, it is essential to provide a cryptographic solution. Hence we are proposing a secure data logging system using Raspberry Pi-3. The objectives of the proposed system are:

- To reduce the message hashing time.
- To design a light-weight digital signature scheme.
- To reduce the verification time of the signature.

4.2. Network model

In the proposed authentication method, Two Raspberry Pi 3 minicomputers are considered for the experiments. One Raspberry Pi 3 is made as client model and it is connected with DHT-11 and MQ-135 sensors to collect the environmental data. This data is signed and forwarded to another Raspberry Pi 3 model which acts as a server. At the server, the verification of the message is carried out. Table 3 explains the notations and symbols used in the proposed work.

5. Proposed method

Primarily, when the nodes are deployed, they need to generate cryptographic keys that can be used to register themselves with the server or base station. So at the time of deployment of Raspberry Pi-3 nodes certain security related global parameteres are stored. They are elliptic curve parameters E_i(*p*, *a*, *b*, *G*, *n*) (Where *i* represent the *i*th Koblitz curve (Secp160k1, Secp192k1, Secp224k1 and Secp256k1) [45,56])), hashing algorithm c-BLAKE2b parameters, node identity (A_{ID}), the base station identity (BS_{ID}), current node power(P^A_{cur}), deployment time ($DTime_A$) of the node

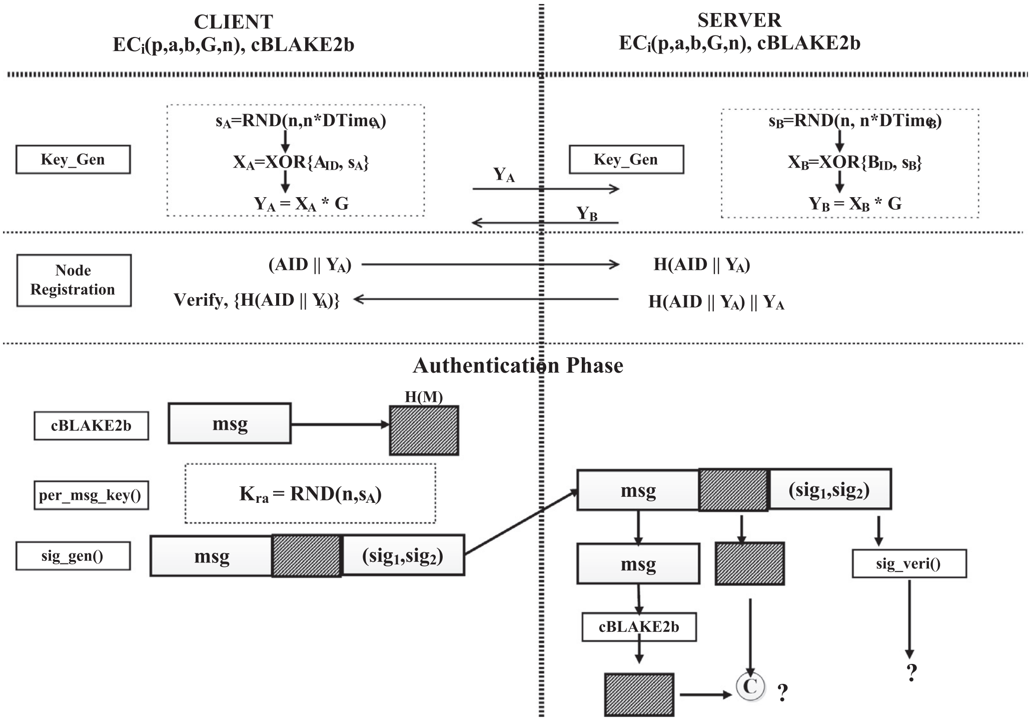

Fig. 2. Overall system design of proposed authentication scheme.

and respective cryptographic executable algorithms. In the proposed system, we are using ECDSA German version [47,57] with the customized BLAKE2b. The proposed system is divided into three phases: key generation phase, the registration phase and the authentication phase. Fig. (2) summarizes the overall system design of the proposed scheme.

5.1. Key generation phase

In this phase private and public keys of each participant are generated. At node A it generates a random number s_A which ranges between elliptic curve parameter n and $DTime_A$ as shown in Eq. (4).

$$s_A = RND(n, n * DTime_A) \tag{4}$$

Then by using s_A and the identity of node A (A_{ID}), a private key X_A is generated as shown in Eq. (5):

$$X_A = XOR(A_{ID}, s_A) \tag{5}$$

As per modular arithmetic, calculation of inverse function is more expensive, hence to reduce the cost of inverse function at the time of signature verification phase of ECDSA, we are calculating the inverse of private key as X_{A-inv} in the key generation phase and using it for generating the public key Y_A as shown in Eqs. (6) and (7) respectively.

$$X_{A-inv} = X_A^{-1} mod \quad n \tag{6}$$

$$Y_A = X_{A-inv} * G \tag{7}$$

5.2. Registration phase

Once all nodes have calculated their respective key pairs, the public keys of the nodes are shared with the BS through the registration phase as in Fig. 3. By considering the node A that wishes to register with BS, it forwards the public key to BS as $A \rightarrow BS$: $A_{ID}||Y_A$. Once the public key of node A is received, BS shall generate a hash value by concatenating the node ID and public key of A and encrypts it using the public key of node A and forwards it to A i.e., $BS \rightarrow A$: $Encrypt[H(A_{ID}||Y_A), Y_A]$. At the node A, a hash value is generated using A's id and public key as $H(A_{ID}||Y_A)$ and this hash value is compared with the received hash value and node A's registration is confirmed based on verification of hash value generated.

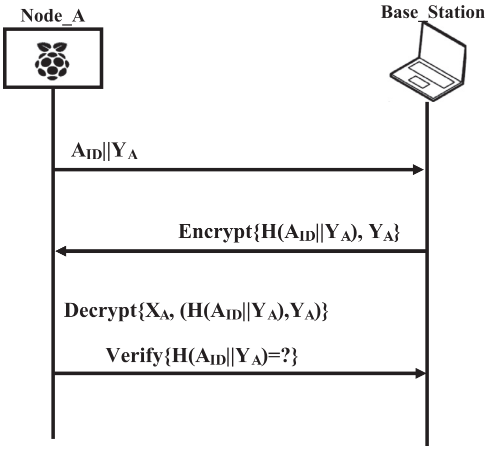

Fig. 3. Registration phase.

5.3. Authentication phase

5.3.1. Signature generation process

Using private key of node A, the message M is signed as shown in following steps:

1. Select K_a as a random number ranging between n and s_A.
2. Generate the hash value of message M using c-BLAKE2b hashing algorithm as $H(M)$.

3. By using point multiplication property of ECDSA, $KG = K_a\mathbf{.}G$ is generated as one-time signing value.
4. Using the x coordinator value of KG a signature component is generated as: $sig_1 = KG_x \ mod \ n$
5. The second signature component is generated as: $sig_2 = [(sig_1 \ \mathbf{.} \ K_a) - (H(M))] \ \mathbf{.} \ X_A \ mod \ \ n$
6. Node A attaches the message with it's hash code and signature pair and transmits it to the base station.

5.3.2. Signature verification process

The verification process is performed at the base station using the sig_1 and sig_2 values as following steps:

1. Obtains message M, hash code $H(M)$ and (sig_1, sig_2).
2. Checks for the range of sig_1 and sig_2 to be between [1, n−1]. If it fails, then reject the signatures.
3. Generate the hash value of message M using c-BLAKE2b hashing algorithm as $H(M)$ which is same as that at node A.
4. Then B generates following values:

$$\omega = sig_1^{-1} \ mod \ n$$
$$v_1 = H(M) \ \mathbf{.} \ \omega \ mod \ n$$
$$v_2 = sig_2 \ \mathbf{.} \ \omega \ mod \ n$$

5. Using these values B obtains the point $(x, y) = v_1 \ \times \ G \ + \ v_2 \times Y_A$ on the curve, if (x, y)=0, signature is rejected.
6. Otherwise, B convert the x-coordinate of (x, y) to an integer x and computes $t = x$ (mod n)
7. If the $t = sig_1$ accept the signature else reject it.

The proof of verification is given as: If (sig_1, sig_2) are signature pair generated by the user A on the message M, then $sig_2 = K_a^{-1}(H(M) + dsig_1) mod\, n$, by rearranging this we have:

$$\begin{aligned} K_a &= sig_2^{-1}(H(M) + d \quad sig_1) \quad mod \quad n \\ &= sig_2^{-1}H(M) mod \quad n + d \quad sig_2^{-1} sig_1 \quad mod \quad n \\ &= \omega \quad H(M) + d \quad \omega sig_1 \\ &= (v_1 + d \quad v_2) \quad mod \quad n. \end{aligned}$$

Thus $v_1 \ \times \ G \ + \ v_2 \times \ Y_A \ = \ (v_1 + d.v_2)G = K_a G$. So, $t = sig_1$ as required [40].

The complete Authentication phase is summarized in the Fig. (4) where the first six steps show the signature generation phase by Raspberry Pi 3 and next seven steps shows the signature verification phase at the base station sides.

6. Implementations and performance analysis

This section describes the experimental setup using the DHT-11 temperature sensor and MQ-135 harmful gas detection sensor with Raspberry Pi 3 model B as the processing unit as in Fig. (5). Also, the proposed method is evaluated on a real-time setup and results are compared with basic BLAKE2b.

6.1. Experimental setup

To replicate a real-life scenario the study is made use of DHT-11 (Fig. (6)) sensor and MQ-135 (Fig. (7)) sensors to collect the temperature, humidity and carbon dioxide (CO_2) content in the experimental area. The Table (4) shows the pin connection made between the sensors and Raspberry Pi 3 board.

The DHT-11 sensor has four pins among which VCC pin take 5V input power and GND is grounded. The not connected pin(NC pin) is left unused. Finally, the last DATA pin is used to obtain the output with a reading of 5 bytes data. Where the first 2 bytes are used to represent relative humidity in percentage and next 2 bytes represent the temperature value in degree Celsius. The last byte is used as checksum that helps in evaluation of sensed values.

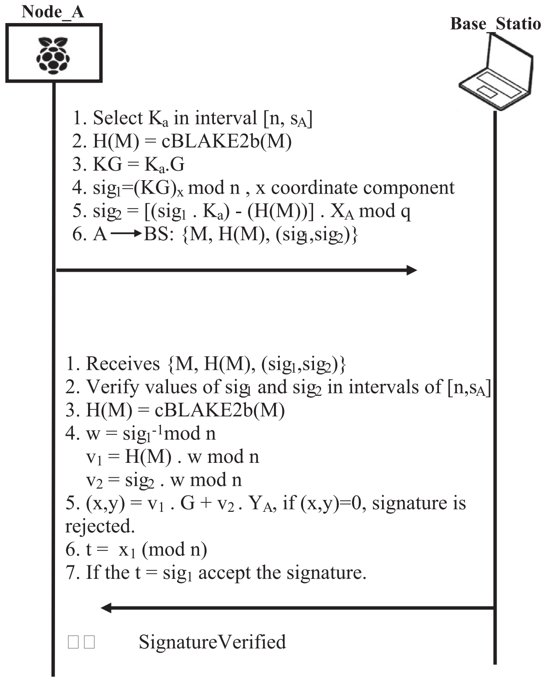
Node_A
Base_Station
1. Select K_a in interval [n, s_A]
2. H(M) = cBLAKE2b(M)
3. KG = K_a.G
4. sig_1=$(KG)_x$ mod n , x coordinate component
5. sig_2 = [(sig_1 . K_a) - (H(M))] . X_A mod q
6. A ⟶ BS: {M, H(M), (sig_1,sig_2)}
1. Receives {M, H(M), (sig_1,sig_2)}
2. Verify values of sig_1 and sig_2 in intervals of [n,s_A]
3. H(M) = cBLAKE2b(M)
4. w = sig_1^{-1}mod n
v_1 = H(M) . w mod n
v_2 = sig_2 . w mod n
5. (x,y) = v_1 . G + v_2 . Y_A, if (x,y)=0, signature is rejected.
6. t = x_1 (mod n)
7. If the t = sig_1 accept the signature.
□□ SignatureVerified

Fig. 4. Signature generation and verification process of authentication phase.

Table 4
PIN connection between sensors and test board.

	Sensor PIN	Raspberry Pi PIN
DHT-11	VCC	5 V
	GND	GND
	Vout	GPIO-4
MQ-135	VCC	5 V
	GND	GND
	Aout	GPIO-26

Table 5
Raspberry Pi 3 model B specification.

Board	Raspberry Pi 3	Architecture	ARMv8-A
SoC	Broadcom	CPU	1.2 GHz quad-core
Core frequency	250 MHz frequency	ARM	600 MHz
On-board network	Ethernet, WiFi 802.11, Bluetooth 4.1	Core OS	1.3062 V

Similarly, MQ-135 has four pins: VCC and GND are same as of DHT-11. The D-out pin gives digital output and A-out pin provides the analog output. To convert the analog output to CO_2 value in parts-per-million (ppm) we used the following Eq. (8) [58].

$$\begin{gathered} Rs = (5.0 * RL) - (RL * V_{out}) \\ \text{Ratio} = Rs/Ro \\ CO_2 = 146.15 * (2.868 - \text{Ratio}) + 10 \end{gathered} \tag{8}$$

Where Ro is internal sensor resistance, RL is sensor pulldown resistance and Rs is sensing resistance according to data sheet [58]. These sensors are coupled with a palm-size processor called Raspberry Pi 3 model B and the specification of Raspberry Pi 3 is shown in Table (5).

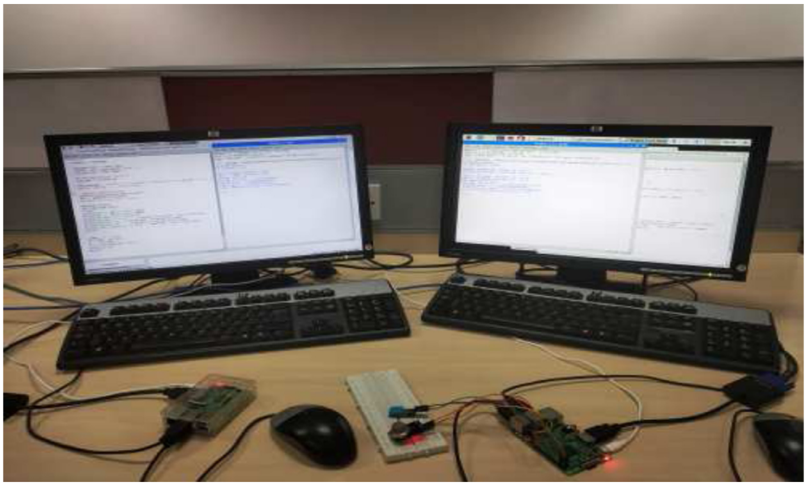

Fig. 5. Proposed setup at laboratory conditions.

Fig. 6. DHT-11 temperature sensor.

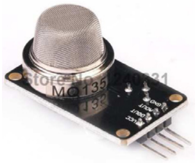

Fig. 7. MQ-135 gas sensor.

Table 6
Message size generated w.r.t time period.

Iteration	Duration (s)	Message size (B)
1	60	156
2	120	312
3	180	468
4	240	624
5	300	780
6	600	936

6.2. Evaluation of proposed method

The proposed method is evaluated under lab condition to check the performance and execution time on a Raspberry Pi 3 based IoT application. The hardware setup is done as described in the previous section. The setup used is related to environmental air quality monitoring for agricultural-based application hence the sensed data are not transmitted to the base station every second. Instead, the program output 16 bytes of data for every 10s and these byte strings are combined based on the different period and then forwarded to the server. The data is collected for the period of the 60s, 120s, 180s, 300s and 600s which is shown in Table 6 to compare the cBLAKE2b execution time with BLAKE2b and mBLAKE2b.

The proposed system is evaluated based on the time efficiency of the execution of BLAKE2b and cBLAKE2b. By considering the time taken by BLAKE2b as T_{b2b} and by cBLAKE2b as T_{c2b} the time efficiency is calculated as $(T_{b2b} - T_{c2b})/T_{b2b}$ and multiplying it with 100 to obtain the performance improvement of cBLAKE2b over BLAKE2b.

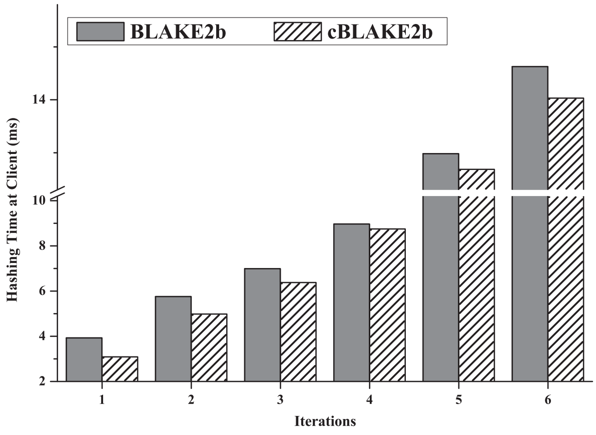

Fig. 8. Time taken to hash the data of various sizes at client.

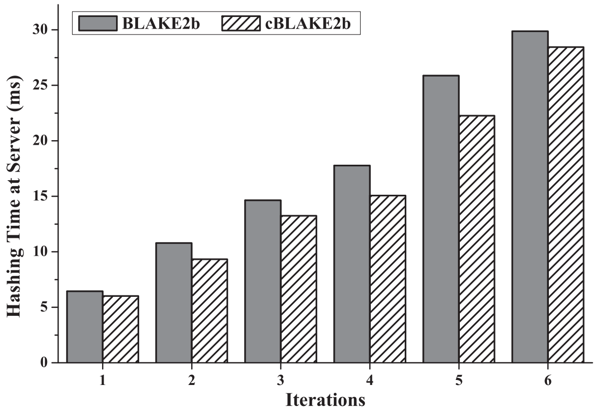

Fig. 9. Time taken to hash the data of various sizes at server.

Figs. (8) and (9) describes the hashing time taken by client and server machines at various iterations as mentioned in Table 6. From Fig. (8) it is shown that the customized BLAKE2b (cBLAKE2b) function takes considerably less time when compared with the BLAKE2b function [13,15]. This is because the computation done by the parameter block of BLAKE2b is omitted in the cBLAKE2b and also the change the *G* function rotation has resulted in a reduction

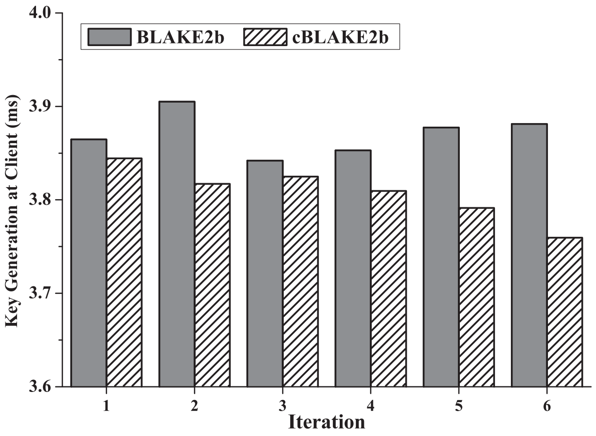

Fig. 10. Key generation time at client machines.

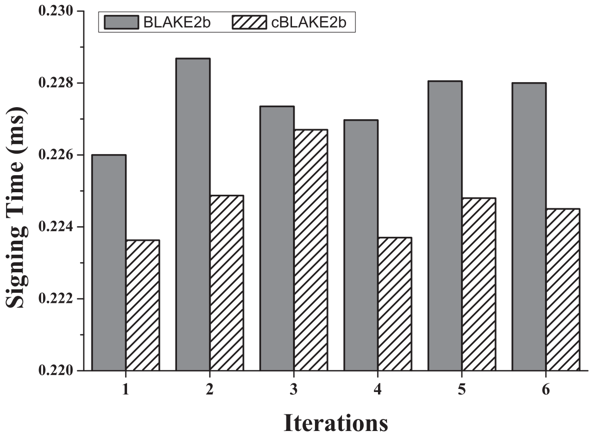

Fig. 12. Signature Generation time is milliseconds.

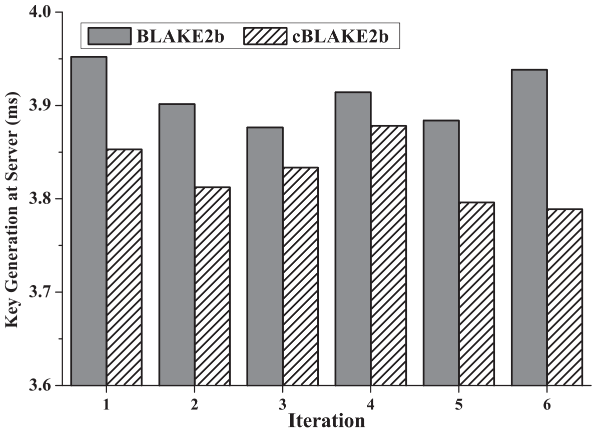

Fig. 11. Key generation time at server machines.

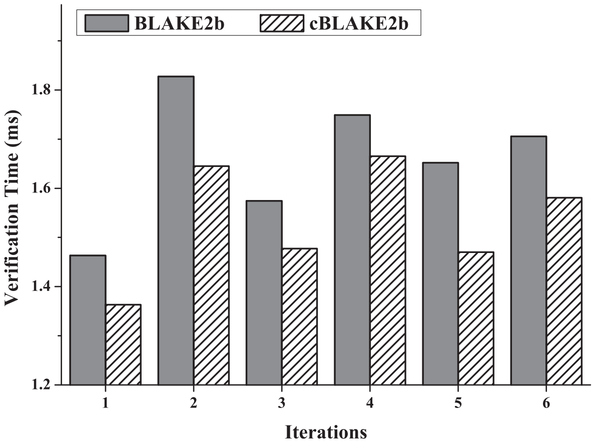

Fig. 13. Verification time is milliseconds.

of time consumed during hashing function. Similarly, at the server side, the hashing done by using cBLAKE2b uses less time than that of BLAKE2b. Overall, the cBLAKE2b has shown an average time efficiency of 4.075–8.70 % for various data sizes mentioned in Table 6.

Fig. (10) describes the key generation time at the client machine for various data size mentioned in Table 6. From the graph, it is observed that the key generation time taken by the proposed authentication time with BLAKE2b is noticeably more than that of cBLAKE2b hashing method. On an average, cBLAKE2b has shown almost 1.11–3.13 % of time efficiency than that of BLAKE2b.

Similarly, when the time taken to generate a key is checked at the server machine, the cBLAKE2b took less time than to BLAKE2b as shown in the Fig. (11). It can be observed that cBLAKE2b showed a time efficiency of 1.10–3.79% improvement than BLAKE2b.

Fig. (12) illustrates the time taken for signing the data sensed by the sensors at the client side. The signing time depends on the time taken for hashing the data, hence the when hashing time reduces due to cBLAKE2b, there is a decrease in ECDSA signature time process too. From the Fig. (12) it can be inferred that the signature time for various data sizes is considerably less during cBLAKE2b hashing based signature. On the whole, the cBLAKE2b showed about 0.7–1.91 time efficiency for different size data.

Fig. (13) represents the time taken for signature verification at the server end for various iterations. The time taken for verifying the signature using cBLAKE2b is much less when compared with the signature verification time using BLAKE2b hashing algorithm with the ECDSA. It is shown that the time efficiency of cBLAKE2b with ECDSA is almost 7.67–9.13% more than that of BLAKE2b with ECDSA.

6.3. Security analysis of the proposed method

In this section, a theoretical study on the security strength of the proposed schemes are discussed. The security strengths of cBLAKE2b is considered under the commonly observed security attacks of hashing functions as mentioned in [17].

6.3.1. Security of authentication scheme

In the proposed authentication scheme, the client registers with the server by using its identity and the public key. During this phase, the mutual authentication is provided by using the German based ECDSA algorithm. The hash value generated is the output of cBLAKE2b (a one-way hash function) and it is impossible for the adversary to forge the content on either of the sides. The following standard attacks are addressed in the remaining part of this section.

- Man-in-the-middle attack (MITM): In the registration phase of the proposed scheme, the hash value of the node identity is generated by the server as $H(A_{ID}||Y_A)$ and encrypted using the public key of the client and finally, this encrypted hash is broadcast. During this process, it is easy for the server to know the intruders who are not registered with it. Hence in the registration phase, the MITM attack is handled.
- Distributed DoS (DDoS) attack: In the proposed scheme, the server is assumed to be resourceful and secured without duplication. Thereby, in the registration phase, when the server receives the node ID, it generates H(ID) and verifies with the already generated and stored H(ID) and the broadcast the mes-

sage. Hence, DDoS is prevented. When considering the authentication phase, at the node level, the hash values received by the server is compared with the hash value generated by the node, the signature verification starts. Thereby, DDoS is prevented.

- Replay attack: In the proposed method, a unique hash value is generated for each client. This process ensures that the adversary cannot generate replay attack. Also the randomness of s_A prohibits the adversary to identify the K_a that protects from replay attack.

6.3.2. Security of cBLAKE2b

The major three security requirements of any hash functions are pre-image resistance, second pre-image resistance and collision resistance. The common design approach of any hash function is that it should have pre- and suffix free padding. cBLAKE2b follows the similar structure that of HAIFA construction but omits the use of the salt and the counter values. Thereby like HAIFA, cBLAKE2b protects the function against collision resistance and second pre-image attack. And by considering the attack on ECDSA, the strength of ECDSA depends on hash function and ECDLP. If the hash function is not resistance to pre-image and second pre-image attacks, it would be easier to attack ECDSA. Hence, cBLAKE2b is a better choice among the other existing cryptographic hash functions.

7. Conclusion

IoT being a universal network enables connection with every object around us through the Internet. As these objects are communicating under the public domain, IoT is facing major issues about security and privacy of the user data. The measures that have been taken to provide a secure and authentic scheme for the security of the data have always shown a resource-constraint issue. This paper projects a new version of hashing function for providing authentication among resource-constrained devices of IoT. The proposed authentication scheme uses a light-weight elliptic curve digital signature and customized BLAKE2b (cBLAKE2b) hashing algorithm. Performance of cBLAKE2b is compared with traditional BLAKE2b on a real-time platform of Raspberry-Pi 3. The performance parameters like hashing time, data size, signature generation time and verification time are considered in the proposed method. The experiments on various data sizes using cBLAKE2b have shown a better performance than the BLAKE2b. There is improvement in efficiency under cBLAKE2b during the signature generation and verification, thereby consuming less power of the IoT devices. More rigorous security analysis needs to be carried out. Our work is evaluated against MITM and replay attacks, other passive and active attacks may also be considered.

References

[1] K.A.M. Zeinab, S.A.A. Elmustafa, Internet of Things applications, challenges and related future technologies, World Scientific News 2 (67) (2017) 126–148.

[2] M.J. McGrath, C.N. Scanaill, Sensor Technologies: Healthcare, Wellness and Environmental Applications, Apress, 2013.

[3] T. Standardization Sector of ITU Series Y, Global Information Infrastructure, Internet Protocol Aspects and Next-Generation Networks, ITU, Geneva, 2012, pp. 1–6.

[4] M.A. Jazayeri, S.H. Liang, C.-Y. Huang, Implementation and evaluation of four interoperable open standards for the Internet of Things, Sensors 15 (9) (2015) 24343–24373.

[5] M.-S. Hwang, C.-C. Lee, J.-Z. Lee, C.-C. Yang, A secure protocol for bluetooth piconets using elliptic curve cryptography, Telecommunication Systems 29 (3) (2005) 165–180.

[6] P. Ray, A survey on Internet of Things architectures, Journal of King Saud University Computer and Information Sciences (2016).

[7] F. Hu, Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations, CRC Press, 2016.

[8] R. Khan, P. Maynard, K. McLaughlin, D. Laverty, S. Sezer, Threat analysis of blackenergy malware for synchrophasor based real-time control and monitoring in smart grid., ICS-CSR, 2016.

[9] O. Vermesan, P. Friess, Internet of Things-from research and innovation to market deployment, 29, River Publishers Aalborg, 2014.

[10] C.-T. Li, T.-Y. Wu, C.-L. Chen, C.-C. Lee, C.-M. Chen, An efficient user authentication and user anonymity scheme with provably security for iot-based medical care system, Sensors 17 (7) (2017) 1482.

[11] F.A. Alaba, M. Othman, I.A.T. Hashem, F. Alotaibi, Internet of Things security: A survey, Journal of Network and Computer Applications 88 (2017) 10–28.

[12] C.-C. Lee, C.-T. Chen, P.-H. Wu, T.-Y. Chen, Three-factor control protocol based on elliptic curve cryptosystem for universal serial bus mass storage devices, IET Computers & Digital Techniques 7 (1) (2013) 48–56.

[13] J.-P. Aumasson, L. Henzen, W. Meier, R.C.-W. Phan, SHA-3 proposal BLAKE, Submission to NIST. Finalized in 2012.

[14] D. Chang, M. Nandi, M. Yung, Indifferentiability of the hash algorithm BLAKE., IACR Cryptology ePrint Archive 2011 (2011) 623.

[15] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, C. Winnerlein, BLAKE2: simpler, smaller, fast as MD5, in: International Conference on Applied Cryptography and Network Security, Springer, 2013, pp. 119–135.

[16] B. Preneel, The first 30 years of cryptographic hash functions and the NIST SHA-3 competition, in: Cryptographers Track at the RSA Conference, Springer, 2010, pp. 1–14.

[17] M. Lavanya, V. Natarajan, LWDSA: Light-Weight Digital Signature Algorithm for wireless sensor networks, Sādhanā 42 (10) (2017) 1629–1643.

[18] N. Li, D. Liu, S. Nepal, Lightweight mutual authentication for IoT and its applications, IEEE Transactions on Sustainable Computing 2 (4) (2017) 359–370.

[19] K.-W. Kim, Y.-H. Han, S.-G. Min, An authentication and key management mechanism for resource constrained devices in IEEE 802.11-based IoT access networks, Sensors 17 (10) (2017) 2170.

[20] J. Kim, J. Moon, J. Jung, D. Won, Security analysis and improvements of session key establishment for clustered sensor networks, Journal of Sensors 2016 (2016).

[21] Y. Chen, J.-F. Martínez, P. Castillejo, L. López , A privacy protection user authentication and key agreement scheme tailored for the Internet of Things environment: PriAuth, Wireless Communications and Mobile Computing 2017 (2017).

[22] J. Lee, Y. Sung, J.H. Park, Lightweight sensor authentication scheme for energy efficiency in ubiquitous computing environments, Sensors 16 (12) (2016) 2044.

[23] Y. Chung, S. Choi, D. Won, Anonymous authentication scheme for intercommunication in the Internet of Things environments, International Journal of Distributed Sensor Networks 11 (11) (2015) 305785.

[24] A.L. John, S.M. Thampi, Mutual authentication based on HECC for RFID implant systems, in: International Symposium on Security in Computing and Communication, Springer, 2016, pp. 18–29.

[25] C.-T. Li, C.-Y. Weng, C.-C. Lee, A secure rfid tag authentication protocol with privacy preserving in telecare medicine information system, Journal of Medical Systems 39 (8) (2015) 77.

[26] M. Wazid, A.K. Das, V. Odelu, N. Kumar, M. Conti, M. Jo, Design of secure user authenticated key management protocol for generic IoT networks, IEEE Internet of Things Journal 5 (1) (2018) 269–282.

[27] S.-C. Lin, C.-Y. Wen, W.A. Sethares, Two-tier device-based authentication protocol against PUEA attacks for IoT applications, IEEE Transactions on Signal and Information Processing over Networks 4 (1) (2018) 33–47.

[28] M.N. Aman, K.C. Chua, B. Sikdar, Mutual authentication in IoT systems using physical unclonable functions, IEEE Internet of Things Journal 4 (5) (2017) 1327–1340.

[29] J. Srinivas, S. Mukhopadhyay, D. Mishra, Secure and efficient user authentication scheme for multi-gateway wireless sensor networks, Ad Hoc Networks 54 (2017) 147–169.

[30] C.-Y. Yang, C.-C. Lee, S.-Y. Hsiao, Man-in-the-middle attack on the authentication of the user from the remote autonomous object, IJ Network Security 1 (2) (2005) 81–83.

[31] V. Shivraj, M. Rajan, M. Singh, P. Balamuralidhar, One time password authentication scheme based on elliptic curves for Internet-of-Things (IoT), in: 5th National Symposium on Information Technology: Towards New Smart World (NSITNSW), IEEE, 2015, pp. 1–6.

[32] P. Kumar, A. Gurtov, J. Iinatti, M. Sain, P.H. Ha, Access control protocol with node privacy in Wireless Sensor Networks, IEEE Sensors Journal 16 (22) (2016) 8142–8150.

[33] X.H. Le, S. Lee, I. Butun, M. Khalid, R. Sankar, M. Kim, M. Han, Y.-K. Lee, H. Lee, An energy-efficient access control scheme for wireless sensor networks based on Elliptic Curve Cryptography, Journal of Communications and Networks 11 (6) (2009) 599–606.

[34] X. Zhou, X. Tang, Research and implementation of RSA algorithm for encryption and decryption, in: Proceedings of 2011 6th International Forum on Strategic Technology, 2, 2011, pp. 1118–1121.

[35] M. Preetha, M. Nithya, A study and performance analysis of RSA algorithm, IJCSMC 2 (2013) 126–139.

[36] Y. Zheng, Digital signcryption or how to achieve cost (signature and encryption) cost (signature) plus cost (encryption), in: Springer Annual International Cryptology Conference, Springer, 1997, pp. 165–179.

[37] A. Roy, S. Karforma, A survey on digital signatures and its applications, Journal of Computer and Information Technology 3 (1) (2012) 45–69.

[38] W. Stallings, Cryptography and Network Security: Principles and Practice, Pearson Education India, 2003.

[39] D. Hankerson, A.J. Menezes, S. Vanstone, Guide to Elliptic Curve Cryptography, Springer Science & Business Media, 2006.

[40] D. Johnson, A. Menezes, S. Vanstone, The Elliptic Curve Digital Signature Algorithm (ECDSA), International Journal of Information Security 1 (1) (2001) 36–63.
[41] A.K. Lenstra, E.R. Verheul, Selecting cryptographic key sizes, Springer Journal of Cryptology 14 (4) (2001) 255–293.
[42] B. Kaliski, IEEE p1363: Standard specifications for public-key cryptography, 1999, p. 45. Aug 17.
[43] K. Lauter, The advantages of elliptic curve cryptography for wireless security, IEEE Wireless Communications 11 (1) (2004) 62–67.
[44] I. ECRYPT, Yearly Report on Algorithms and Keysizes, ECRYPT II Network of Excellence (NoE), funded within the Information Societies Technology (IST) Programme of the European Commissions Seventh Framework Programme (FP7), 2012.
[45] A. Bundesamt fur Sicherheit in der Informationstechnik, Technical Guideline TR-03111, Elliptic Curve Cryptography (2005). https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TR03111/BSI-TR-03111.pdf.pdf?blob=publicationFile.
[46] S. SEC, 2: Recommended elliptic curve domain parameters, Standards for Efficient Cryptography Group, Certicom Corp, 2000.
[47] H.-Z. Liao, Y.-Y. Shen, On the Elliptic Curve Digital Signature Algorithm, Tunghai Science 8 (2006) 109–126.
[48] M. Nandi, Design of Iteration on Hash Functions and its Cryptanalysis, Indian Statistical Institute, Kolkata, 2005 Ph.D. thesis.
[49] R. Sobti, G. Geetha, Cryptographic hash functions: A Rreview, International Journal of Computer Science Issues (IJCSI) 9 (2) (2012) 461.
[50] R. Rivest, The MD5 Message-Digest Algorithm, Technical Report, 1992.
[51] E. Biham, O. Dunkelman, A Framework for Iterative Hash Functions—HAIFA, Technical Report, Computer Science Department, Technion, 2007.
[52] J.-P. Aumasson, W. Meier, R.C.-W. Phan, The hash function family LAKE, in: International Workshop on Fast Software Encryption, Springer, 2008, pp. 36–53.
[53] B. Denton, R. Adhami, Modern hash function construction, in: Proceedings of the International Conference on Security and Management (SAM), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2011, p. 1.
[54] E. Andreeva, C. Bouillaguet, O. Dunkelman, P.-A. Fouque, J. Hoch, J. Kelsey, A. Shamir, S. Zimmer, New second-preimage attacks on hash functions, Journal of Cryptology 29 (4) (2016) 657–696.
[55] J.-P. Aumasson, W. Meier, R.C.-W. Phan, L. Henzen, The Hash Function BLAKE, Springer, 2014.
[56] T. Oliveira, J. López, D. Cervantes-Vázquez, F. Rodríguez-Henríquez, Koblitz curves over quadratic fields, Journal of Cryptology (2018) 1–28.
[57] G. Sarath, D.C. Jinwala, S. Patel, A survey on elliptic curve digital signature algorithm and its variants, in: Second International Conference on Computational Science and Engineering (CSE-2014) Dubai, UAE, 2014, pp. 121–136.
[58] Technical Data on MQ-135, https://www.olimex.com/Products/Components/Sensors/MQ135/SNS-MQ135.pdf.

Vidya Rao is pursuing her Ph.D. in the Department of Computer Science and Engineering at Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE - recognized as “Institute of Eminence” by Union HRD Ministry of India), Manipal. She obtained her B.E. in Computer Science and Engineering in 2010 and M.Tech degree in Computer Network and Engineering in 2013 from Visvesvaraya Technological University, Karnataka, India. Her research interest are in the area of Network security, Sensors and actuators, Wireless Sensor Networks, and Internet-of-Things.

Prema K.V. is working as Professor at Department of Computer Science and Engineering at Manipal Institute of Technology, Manipal Academy of Higher Education (MAHE - recognized as “Institute of Eminence” by Union HRD Ministry of India), Manipal. She has completed her Ph.D under Mysore University, Mysore, Karnataka and M.Tech from University Visvesvaraya College of Engineering, Bangalore. She is having more than 26 years of teaching and 16 years of research experience. She has published various research paper in National and International Journals and conferences. Her areas of interests are Cryptography and Network Security, Ad Hoc Networks, Soft-computing and Body Area Network.