Timestamp Based OTP and Enhanced RSA Key Exchange Scheme with SIT Encryption to Secure IoT Devices

V. N. Hemanth Kollipara[1], Sai Koushik Kalakota[1],
Sujith Chamarthi[1], S. Ramani[1], Preeti Malik[2]
and Marimuthu Karuppiah[3,*]

[1]*School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632 014, Tamil Nadu, India*
[2]*Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, India*
[3]*Department of Computer Science and Engineering & Information Science, Presidency University, Bengaluru 560064, Karnataka, India*
E-mail: marimuthume@gmail.com
Corresponding Author

Received 31 May 2022; Accepted 03 August 2022;
Publication 03 March 2023

Abstract

The Internet of Things (IoT) has become an emerging technology and is expected to connect billions of more devices to the internet in the near future. With time, more and more devices like wearables, intelligent home systems, and industrial automation devices are getting connected to the internet. IoT devices primarily transfer data using wireless communication networks, introducing more vulnerabilities like man-in-the-middle-attacks and eavesdropping. These security concerns are customary for any device communicating over the internet because of its intrinsic open nature. These problems are usually subdued by conventional cryptographic algorithms used in typical systems that are power-hungry and computationally intensive,

Journal of Cyber Security and Mobility, Vol. 12_1, 77–102.
doi: 10.13052/jcsm2245-1439.1214

making them infeasible to be used in IoT devices since they run on low-powered chips, limiting performance, memory, and bandwidth. Hence, there is a requirement to adopt lightweight cryptographic algorithms that can abate the security issues while using low computational resources, which is the constraint in the given scenario. Hence, we propose an end-to-end secured IoT system that ensures the system's integrity is never compromised using lightweight cryptographic algorithms. We propose a three-module system, where the first module handles user authentication using a time-based one-time password, the second secures communication using lightweight enhanced RSA, and the third performs data encryption using Feistel-based enhanced SIT. This kind of system is designed to deal with security challenges in IoT devices, ensuring adequate data security while reducing the computational footprint using lightweight cryptography.

Keywords: IoT, security, time-based authentication, lightweight RSA, device encryption.

1 Introduction

IoT is slowly becoming an integral part of our daily life. As people use more and more intelligent devices, which include smartwatches, fitness trackers that collect personal data to smart home products like smart refrigerators, locks, fire systems, and security systems that transmit critical data around the internet. As the cost of connectivity to the internet is getting cheaper every day, and with accessibility increasing allowing more and more people to connect to the internet along with their smart devices, contributing to the growth of IoT technology. These devices are often embedded systems housing a low-power processor chip that acts as the brains of the system and is connected to a variety of sensors that collect valuable data. This data can often be critical and thus raising the question of security threats [1,2]. In order to maintain the assurance of this technology, it is vital to ensure the security of such low-power devices. These devices are part of a complex network of similar devices, exchanging lots of information over the network. Each device in the network gathers data from their corresponding sensors, ranging from a temperature sensor that collects home temperature to camera sensors that monitor traffic. These sensors generate a humongous amount of data, and this data is communicated between other devices over the internet. Thus, IoT is changing the landscape of the conventional internet to the next level by connecting everything to the internet. So, security concerns related to data confidentiality, integrity, and authentication must be considered seriously. These devices must be able to safeguard the privacy of the user data and should not compromise the integrity of the system. Nevertheless, due to its nature of openness in terms of connectivity, IoT introduces new security challenges since it is inherently vulnerable to various threats like information leakage and unauthorized usages [3, 4]. One backdrop of IoT devices is that they are prone to physical attacks resulting in exposure of data stores in the components. Also, IoT devices are most commonly connected via wireless networking, making them vulnerable to security attacks and unauthorized access, resulting in data loss, data leakage, and damage to the entire network. Encryption would solve this problem of data security for regular devices like computers and smartphones, but using the same cryptographic algorithms for IoT devices which are embedded systems, is questionable since the hardware architecture in these devices is more diminutive and low-powered [5, 6].

Hence to address these security concerns, there is a need to use the appropriate cryptographic algorithms to maintain the integrity of the system. However, conventional cryptographic algorithms are not suitable for these smart devices since they are constrained by energy consumption, computational power, memory utilization, network bandwidth. Hence the cryptographic solutions must be appropriate for low-resource hindered devices, unlike conventional algorithms that use a lot of energy and computational power performing many rounds of encryption [7–9].

This paper presents a three-module system to meet the above requirements as efficiently and reliably as possible. The proposal is to implement various lightweight cryptographic encryption and security algorithms so that the data being transferred between these IoT devices is secure and not vulnerable to breaches. The main focus is to implement these algorithms into one shared platform for all IoT devices, from secure authentication to data encryption. This paper is organized as follows, Section 2 provides an incisive review on the related works on existing authentication, lightweight cryptographic algorithms centered around low powered devices. The proposed end-to-end system architecture overview, algorithms for the proposed systems are described in Section 3. Results and discussions are put forth in Section 4 with the conclusion followed up in Section 5.

2 Related Work

In this section, we discuss the existing authentication and lightweight cryptographic algorithms for user authentication, secure key-exchange over unsecure communication channels and encryption algorithms that are designed for low power embedded devices used in IoT systems. We also outline the various issues, drawbacks and challenges in the implementation of these algorithms.

Most of the secure authentication mechanisms in recent times often adopt a time based, one-time password system [10, 11]. In such a time-based One Time Password (OTP) authentication system as used in [12], the implementation is accomplished at the application level. Two applications, a server application that is running and a client application, are used in the entire process. The server application keeps accepting connection requests, and when a connection request from a client is sent to the server, it gets accepted and connects to the server. The client application first needs to go through a one-time registration process with the server application in order to get itself registered in the server database. The pre-shared secret key used to generate the TOTP is now shared between the client and server, enabling both to generate the same OTP independently. Once this process is done, the client application can use this secret key and the present timestamp at which the TOTP is triggered to generate the current TOTP that is used to establish an authenticated connection. The authors have used the time-based OTP for the secure authentication over the Transport Layer Security (TLS)/Secure Sockets Layer (SSL) tunnel using Secure Hash Algorithm (SHA) based hashing and a two-fish encryption algorithm to add a higher level of security [13]. The drawbacks are that secret keys at the registration process are exchanged using two fish encryption algorithms that are more vulnerable to brute force attacks. Also, the server does not take into consideration the device from which the connection is established, making it vulnerable to spoofing attackers made my attacker who could have managed to break into the network.

Shivraj et al. [14] proposed an end-to-end authentication in low-power IoT devices using an OTP-based algorithm that is lightweight yet reliable and scalable using the concept of Identity-based elliptic curve cryptography. In this proposed methodology, a client device first requests the respective server application to generate OTP. The server responds to the request by generating OTP and then shares this to the request remote device. For the client application to get authenticated, it must first submit its basic credentials, login, and password, and then the OTP to the server application. The server device verifies the received credentials and then proceeds to perform OTP validation. The algorithm first generates two dissimilar integral prime numbers p and q using super-singular elliptic curve, torsion point, and a secret key as given in [14]. The device with device id Dev-id obtains a public key, which is a torsion point on elliptic curve PDev-id, and a private key Dev-id, which is also a torsion point. In the validate phase, the client device communicates with the server application submitting to the OTP request, and the server then validates the OTP received from the client with the one generated by itself. Upon successful validation, it accomplishes the job of authentication between the client device and the remote server. The proposal also includes optimizations in the computation of various cryptographic parameters over the elliptic curve and makes the scheme more lightweight. Thus, using OTP in conjunction with existing authentication techniques is adequate and necessary to abate the security threats outlined. The drawbacks are that OTPs are generated on the server and sent to the user, thus relying on external infrastructure to get the OTP [15].

Lightweight RSA algorithm using three prime numbers proposed by Zaid et al. [16] utilizes three primes rather than two prime standard RSA algorithms. The first public key is generated and published from the intended client where the message is originated. The public key published is used to encrypt the plain text, and this cipher can only be decrypted by using the private only, which is not disclosed. The public key open to everyone cannot be used to decrypt the transmitted message making the transmission secure, enabling the exchange of messages in a secure way maintaining data privacy. The methodology proposed is an improvement to the existing RSA algorithm in terms of speed in key generation and decryption. This proposal [16] takes advantage of three prime numbers and the concept of the Chinese Reminder Theorem. A new prime number was added to the algorithm making the process of factorization a little bit easier, reducing the time taken for key generation. This added component increases the complexity of the algorithm, making the analysis of factorizing the Euler's totient much more complex than the traditional algorithm that uses two primes. This method not only reduces the time for generation, but it also has the added benefit of enhancing the algorithm by protecting it from a few attacks, namely timing attack, modulus attack, plaintext attack, ciphertext attack [17]. The shortcoming being, while the suggested modified RSA method is faster than the standard RSA algorithm, especially when considering key generation and decryption phases, but it does not speed up the process of encryption significantly enough, compromising its efficiency a little bit.

The Elliptic Curve based Diffie-Hellman (ECDH) Key Exchange scheme proposed in [18] focuses on encryption and decryption algorithms that are lightweight and low-power consuming algorithms yet at the same time robust and secure. Also, the paper proposed some enhancements in the computation process of the ECDH Key Exchange. This kind of key exchange on an unsecured channel using elliptic curves is widely used, and use usually uses the Diffie-Hellman Key agreement mechanism to exchange the keys. This public-key cryptography, when compared to others like RSA, offers the same security while using smaller keys lowering the power consumption and reducing the computational load on the system generating the keys. This allows calculations with much more speed and also uses low memory. Elliptic Curve Cryptography (ECC) utilizes an elliptic curve defined over a finite field F_q, which and contains the affine points (x, y) and $F_q \oplus F_q$ that satisfy the Weierstrass equation. Starting with a number q, which could be a prime or one that satisfies the $2m$ and two parameters for the elliptic curve, namely a and b. These values define the elliptic curve equation $Eq(a, b)$. After forming the equation, a base point (x_1, y_1) in $Eq(a, b)$. A small positive integer that satisfies $n \cdot G = 0$ of order n. The calculation of finding the modulus of a fractional number is a little harder, but an easy and simple method is proposed. Thus, the proposed methodology was found to be better than other comparisons made, consuming low power but robust. The main challenge in the proposed lightweight Elliptic Curve Diffie-Hellman Key Exchange is the complexity of the operations involved and the difficulty of implementing it securely, particularly the standard curves.

Symmetric encryption algorithms are usually based on block cipher or stream cipher. Bit-by-bit encryption is performed while using a stream cipher, whereas a group of bits (block) is considered before encryption. Block ciphers have higher computational throughput, hence preferred the most. The Tiny encryption algorithm [19] is a block cipher algorithm, and it is one of the most efficient and straightforward algorithms used for data encryption. Moreover, the algorithm is designed following the memory constraints in IoT devices and have Shannon's properties of complete confusion and discussion by implementing substitution and permutation boxes. Tiny Encryption Algorithm takes 64 data bits using a 128-bit key with 32 rounds. From the previous rounds, left and right inputs are derived, and from the 128-bit key, a subkey is derived. In each round, delta multiples are used where delta is a constant, which is 2654435769, and to ensure subkeys used in each round are different golden ratio is used. In the Tiny Encryption Algorithm, the plaintext is split into two halves, and the subkey will be applied to one half in the round function F. Then, the output of the F function will be XORed against another half before the two halves are swapped. This pattern is similar to all the rounds except the last one. The algorithm consists of dual bit shifting, which mixes data regularly, and XOR and ADD operations provide diffusion and confusion for the secure block cipher. TEA has a few weaknesses. It experiences identical keys attack that is each key is equal to three others. The related-key attack is also a danger for tiny encryption.

The Simon cipher proposed by Beaulieu et al. [20] is based on performing simple mathematical operations during the cryptographic computation that primarily uses a block cipher algorithm. Utilize the following procedures on the n-bit word: Bitwise Adding, Bitwise XOR, right circular shift, and Bitwise left for encryption and decryption. The block's upper and lower words are x_{i+1} and x_i, which is a bit word. The initial input consists of these two words which is called plain text. As bitwise XORing, bitwise ANDing and left circular shift operation are performed in round function. Circular shifting and bitwise AND operations are performed in every round on the upper word, and it is XORed with the lower word and the round key. As a result, its value is shifted to the lower word. Until the given number of rounds is reached, round functions continue to run repeatedly. Round functions continue to run until the specified number of rounds are reached [20]. Simon block cipher has a few weaknesses. Known-key distinguishing attack model is one major issue with Simon block cipher, and related-key attacks have a fair chance of breaking into the encryption of Simon block cipher.

Secure IoT (SIT) presented in [21, 22] is a lightweight encryption algorithm that takes 64-bit plain text as an input and outputs a 64-bit ciphertext. The algorithm is based on a symmetric key cryptography, and it consists of five encryption rounds. It has various mathematical functions in each round to achieve Shannon properties of both confusion and diffusion. The given algorithm consists of only five rounds. It is energy efficient, and the algorithm uses the feistel network of permutation and substitution functions to create enough randomness in the data to face the attacks. In symmetric key algorithms, the key generation is a necessary process. The key generation involves complex mathematical operations as the encryption algorithm consists of five rounds, and a separate key is used for each round. The proposed algorithm takes an input of 64 bits of data and a key for the data encryption. The key used to encrypt the data is taken from the user as input. Then expansion block will be given the user key as input. In the key expansion procedure, several operations create confusion and diffusion required to confront attacks and finally generate five different keys, each of 16 bits. The generated keys are used in each round of the encryption process, and these are good enough to stay the same through an attack. The key expansion structure consists of several F functions, ensuring sufficient shuffling of bits to ensure no dependency between input keys. Thus SIT employs a combination of feistel based and uniform substitution-permutation to provide enhanced security for data in IoT devices using lower computational resources.

3 Methodology

The proposed system architecture consists of three basic modules as shown in Figure 1 one handles authentication, another secures the communication channel, and another ensures that data privacy is not compromised through device encryption. The authentication module uses Time-based OTP as a multi-factor authentication method in addition to a traditional ID and password. The Data Encryption module implements a lightweight symmetric key cryptography used for data encryption for both storage and transmission. The symmetric key is exchanged using a lightweight RSA algorithm suitable for IoT devices. This kind of system is designed to deal with security challenges in IoT devices, ensuring adequate security to the data at the same time reducing the computational footprint with the use of lightweight cryptography.

3.1 User Authentication using Time-based One-time Password (OTP)

Password is considered to be the weakest in any information security system and is the most vulnerable and can be easily cracked. However, this problem has been around for a while, and this process of authentication using

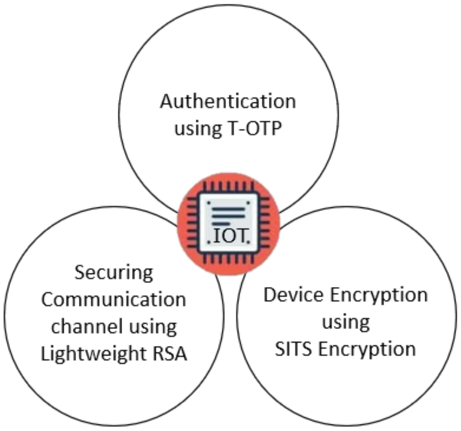
Authentication using T-OTP
IOT
Securing Communication channel using Lightweight RSA
Device Encryption using SITS Encryption

Figure 1 Proposed secure IoT system architecture.

passwords can be further strengthened by using multi-factor authentication such as one-time password based authentication. One such type of OTP-based authentication is time-based OTP. Here, authentication between the client and server depends on a pre-shared secret key that could be exchanged between the client and server using public-key cryptography if it was not done previously. Conventional OTP's are usually generated on the server and are then sent to the client utilizing SMS or email, but in this time-based one-time password, both the server and the client can generate the OTP using the pre-shared key and the timestamp at which the OTP was generated. These time-based OTPs are valid only for a specific interval of time and expire after that time interval for greater security. This process is represented as a flow diagram as shown in Figure 2 and Algorithm 1.

HMAC-based One-time Password algorithm (HOTP) is a token and a pre-shared symmetric key-based validation algorithm that generates a one-time password. Each time an OTP is requested, the token count increases

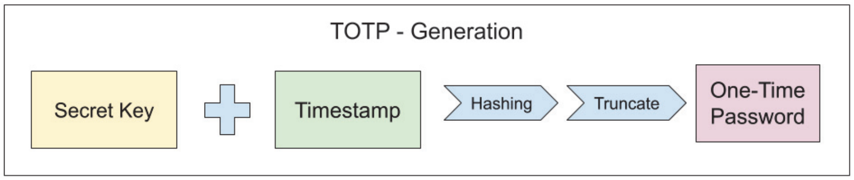

Figure 2 TOTP generation basic flow diagram.

```
$T_{unix}$ : number of seconds elapsed since Unix Epoch time
$T_c$ : number of time steps elapsed since Unix Epoch time
$ts$ : time step in seconds, default is 30sec
$P_{sk}, K$ : pre-shared secret key
$m$ : message to be hashed using key K
SHA-1 : cryptographic hash function
$n_d$ : number of digits required in the OTP
$ts \leftarrow 30$
$T_c = \lfloor T_{unix}/ts \rfloor$
TOTP $=$ HOTP$(P_{sk}, T_c)$
HOTP$(K, C) = Truncate($HMAC $(K, C))$
HMAC$(K, m) =$ SHA-1$((K \oplus opad) \parallel$ SHA-1$((K \oplus ipad) \parallel m))$
$opad \leftarrow$ 0x5c, $ipad \leftarrow$ 0x36
TOTP $=$ TOTP $\pmod{10^{n_d}}$
```

Algorithm 1 T-OTP generation

and HMAC SHA1 is performed, generating a 160-bit output which is then truncated and a 4-byte binary code is extracted from the truncated output.

Introducing an efficient, lightweight encryption algorithm can make the authentication process more secure. Furthermore, the International Mobile Equipment Identity (IMEI) number of the client device could be associated with the secret key by signing it with the IMEI number, thus prevents unauthorized access by any attackers who managed to break into the network as they need the physical device to authenticate. Thus, the proposed authentication system adds an extra layer of security to the existing password stack by leveraging the T-OTP algorithm to generate one-time passwords.

3.2 Key Exchange in Unsecured Channel using Lightweight RSA

This type of key exchange security mechanism is commonly known as public-key cryptography. This mechanism uses two different types of keys, one public key, which is published and can be seen by everyone, and another private, which is stored secretly by the publisher. First, the sender views the published public key, encrypts the plain text, and sends the ciphertext to the receiver who generated the keys. The receiver decrypts the ciphertext using the secret private key that is not disclosed anywhere into plain text. The ciphertext the sender sends can only be decrypted by using the private key to maintain the confidentiality of the transmitted message. Since the information disclosed by the public key is not enough is decrypt the ciphertext, this mechanism is widely used for key exchange in a protected manner over an unsecured channel. One such algorithm is RSA which uses public-key cryptography.

The proposed Algorithm 2 is a modification of the standard algorithm, which usually uses only two primes, but here we use three prime numbers. This proposal takes advantage of three prime numbers and the concept of the Chinese Reminder Theorem to provide speedup enhancement on on key generation and decryption part of the algorithm. The key idea is that factorizing the large prime number N used in the algorithm and getting back the original numbers p, q, r is very hard since there exist no algorithm that could perform this task in reasonable amount of time, increasing the strength of the algorithm. Figure 3 below shows the block diagram of key generation, encryption and decryption, all the steps involved in the algorithm.

Also, this enhancement prevents the algorithm from a few known attacks along the way. A new prime number added to the algorithm increases its complexity but reduce the computational time, increasing the algorithm's security as it is much harder to analyze the patterns and perform attacks.

```
$p, q, r$ : prime numbers chosen at random
$\phi$ : Euler's Totient
GCD : Greatest Common Divisor
$K_{public}$ : Public Key
$K_{private}$ : Private Key
$M$ : Plain text
$C_m$ : Cipher Text

Key Generation
    $N = p \times q \times r$
    $\phi(N) = (p-1) \times (q-1) \times (r-1)$
    Choose $e$ such that $1 < e < \phi(N)$, GCD$(e, \phi(N)) = 1$
    Compute $d$ where $e \times d = 1 \bmod \phi(N)$
    Compute $d_p$ where $e \times d_p = 1 \bmod (p-1)$
            $d_q$ where $e \times d_q = 1 \bmod (q-1)$
    Compute $Q_{in}$ where $q \times Q_{in} = 1 \bmod p$  if $p > q$
                           $p \times Q_{in} = 1 \bmod q$  if $q > p$
    Public Key, $K_{public} = (e, N)$
    Private Key, $K_{private} = (Q_{in}, d_p, d_q, p, q)$
Encryption
    Cipher Text, $C_m = M^e \bmod N$
Decryption
    $M_a = C^{d_p} \bmod p$
    $M_b = C^{d_q} \bmod q$
    $h = (Q_{in} \times (M_a - M_b)) \bmod p$
    $M = M_b + (h \times q)$
```

Algorithm 2 Lightweight RSA

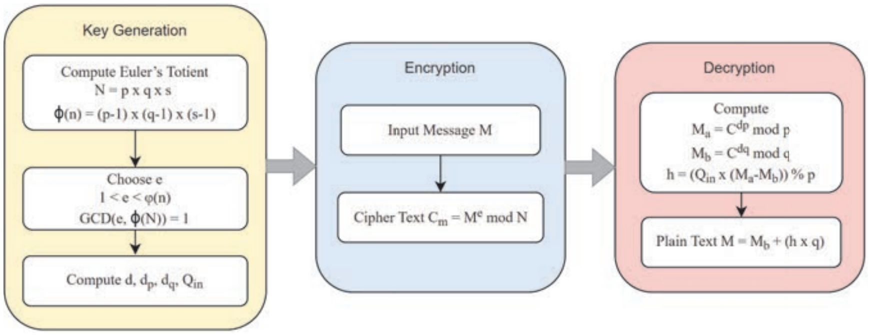

Figure 3 Lightweight RSA key generation, encryption, decryption.

3.3 Hybrid User Authentication using T-OTP and Lightweight Enhanced RSA Key Exchange

Authentication in an insecure environment is quite a challenging task. Here we take a hybrid approach of using both lightweight RSA and T-OTP algorithms (Algorithm 3). Basically, the RSA algorithm is used to generate $public$ and $private$ keys, where the public keys are used to encrypt the client credentials and one-time password, and the private key is used to decrypt the cipher text back to plain text on the server thus protecting the data in transit. The detailed process that takes place to achieve this is mentioned below and depicted in Figure 4.

3.4 Feistel Based Enhanced SIT for Data Encryption

This SIT algorithm is a type of encryption algorithm that is considered lightweight cryptography. These algorithms are designed for implementation in a constrained environment as the application involves an IoT device.

```
$K_{public}$ : Public keys generated using RSA used to encrypt data on client-side
$K_{private}$ : Private keys used to decrypt cipher text received on server-side
$C_{id}, C_{password}$ : Basic credentials from client
$C_{ct}$ : Cipher text generated from plain text on client-side
2FA : Two-Factor Authentication
$C_{otp}$ : One-time password generated by the client
$C_{psk}$: Pre-shared secret key used to generate the T-OTP

1: Client initiates Authentication request
2: Server generates $(K_{public}, K_{private})$ using RSA to secure the channel
   Server publishes $K_{public}$
3: Client encrypts the user credentials $(C_{id}, C_{password})$ using $K_{public}$
   Client submits the cipher text $C_{ct}$ back to the server
4: Server decrypts $C_{ct}$ using $K_{private}$ and verifies the credentials and requests for 2FA
   using T-OTP.
5: Client generates one-time password $C_{otp}$ using T-OTP algorithm, and encrypts using
   $K_{public}$ again. Then, Client submits encrypted T-OTP back to the server.
6: Server also generates T-OTP using $C_{psk}$ simultaneously
7: Server decrypts $C_{otp}$ using $K_{private}$ received from client
8: If $C_{otp}$ is valid
       Client is authenticated successfully and connection is established
9: Else
       Client is denied authentication after certain number of incorrect OTP
```

Algorithm 3 Secure Authentication in insecure communication channel

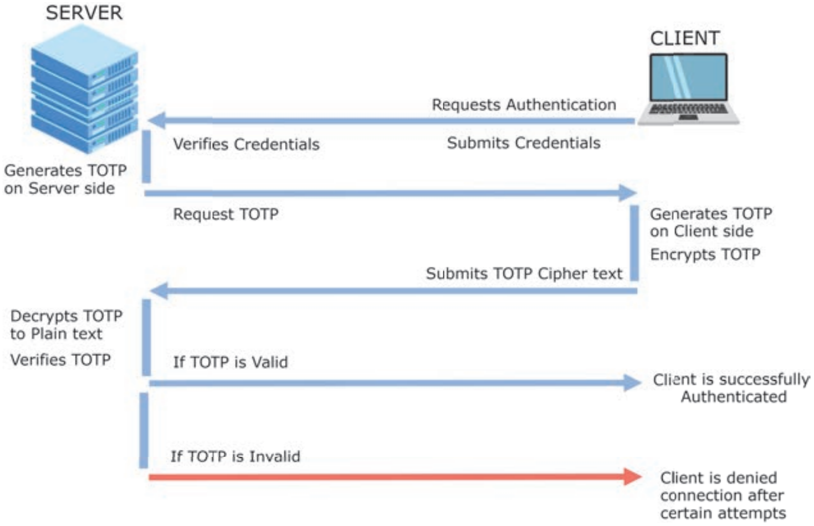

Figure 4 User Authentication using T-OTP and Lightweight RSA.

Memory size and energy consumption are important factors to be considered. The proposed algorithm takes 128-bit plain text and uses five-round encryption to encrypt the data, and each round uses a unique key generated from the key expansion block. As the algorithm consists only of five rounds and 128-bit data encryption, it is ideally suited for IoT devices' security.

3.4.1 Key generation

Key generation is one of the critical steps in the SIT algorithm. The generation of keys is done through various complex mathematical operations and the shuffling of bits. SIT algorithm is a Feistel based encryption algorithm, and in Feistel structure, each round requires a separate key. The encryption and decryption of the above algorithm have five rounds. Therefore, it requires five different keys. The initial input of a 64-bit cipher key is taken as input, and this key can be used for further key expansion.

- The given input 128-bit key is divided into 16 parts, each of 8-bit
- Then, an initial substitution is performed on the parts of the cipher key to send these bits to the f function, which operates on 32-bit data.

  1. $Kb_1f = Kc_0 + Ko_4 + Kc_8 + Kc_{12}$
  2. $Kb_2f = Kc_1 + Kc_5 + Kc_9 + Ke_{13}$
  3. $Kb_3f = Kc_2 + Kc_6 + Kc_{10} + Ke_{14}$
  4. $Kb_4f = Kc_3 + Kc_7 + Kc_{11} + Kc_{15}$

Table 1 Notations

Notation	Function
$\oplus$	XOR
$\odot$	XNOR
$++$	Concatenation

Table 2 P table

K_{ci}	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$P(K_{ci})$	3	F	E	0	5	4	B	C	D	A	9	6	7	8	2	1

Table 3 Q table

K_{ci}	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$Q(K_{ci})$	9	E	5	6	A	2	3	C	F	0	4	D	7	B	1	8

- The f-function in the key expansion architecture consists of P and Q tables, which results in Shannon properties of confusion and diffusion.
- The output of F-function Kaif, which is 16-bit, is written as a 4x4 matrix utilizing splitting into multiple rows. These four matrices are Km_1, Km_2, Km_3, Km_4.
- The round keys, $K1, K2, K3, K4$ are obtained from these matrices where 16-bits are transformed, and the arrangements of these bits can be observed from the block diagram.

1. $K1 = a_4 + +a_3 + +a_2 + +a_1 + +a_5 + +a_6 + +a_7 + +a_8 + a_{12} + +a_{11} + +a_{10} + +a_9 + +a_{13} + +a_{14} + +a_{15} + +a_{16}$
2. $K2 = b_1 + +b_5 + +b_9 + +b_{13} + +b_{14} + +b_{10} + +b_6 + +b_2 + +b_3 + +b_7 + +b_{11} + +b_{15} + +b_{16} + +b_{12} + +b_8 + +b_4$
3. $K3 = c_1 + +c_2 + +c_3 + +c_4 + +c_8 + +c_7 + +c_6 + +c_5 + +c_9 + +c_{10} + +c_{11} + +c_{12} + +c_{16} + +c_{15} + +c_{14} + +c_{13}$
4. $K4 = d_{13} + +d_9 + +d_5 + +d_1 + +d_2 + +d_6 + +d_{10} + +d_{14} + +d_{15} + +d_{11} + +d_7 + +d_3 + +d_4 + +d_8 + +d_{12} + +d_{16}$
5. $K5 = K1 \oplus K2 \oplus K3 \oplus K4$

3.4.2 Encryption

After generating five-round keys, the encryption algorithm consists of left-shifting of bits and swapping groups of 32 bits and substitution.

- The given input 128-bit key is divided into four segments, each of 32 bits.

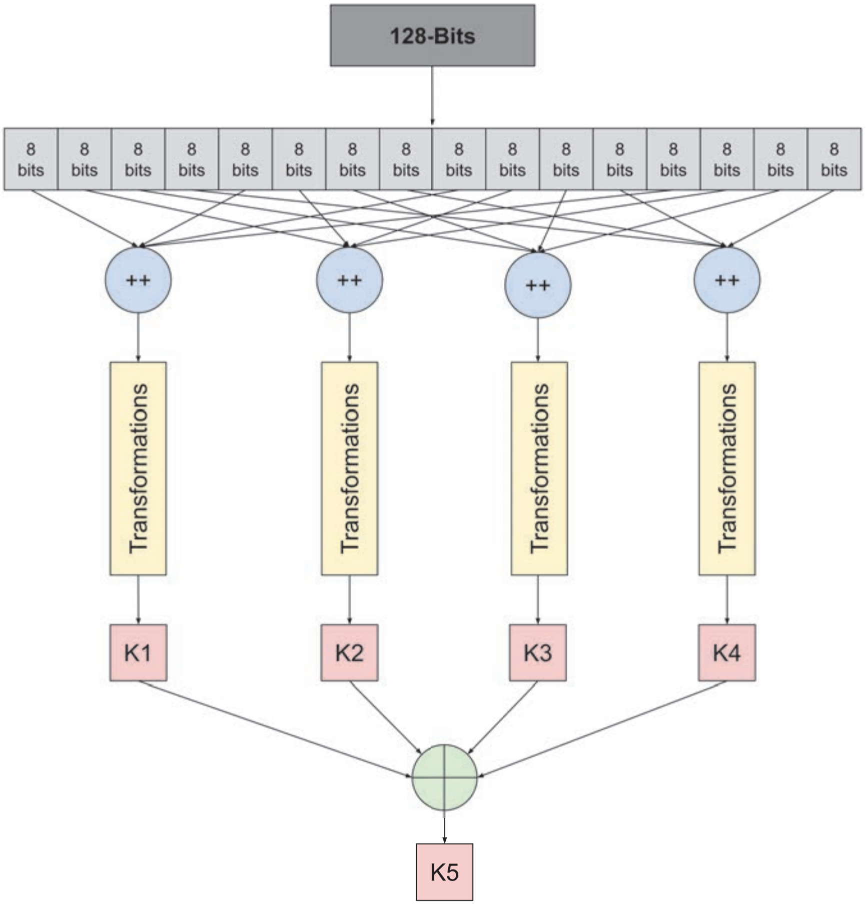

Figure 5 Key expansion flow diagram.

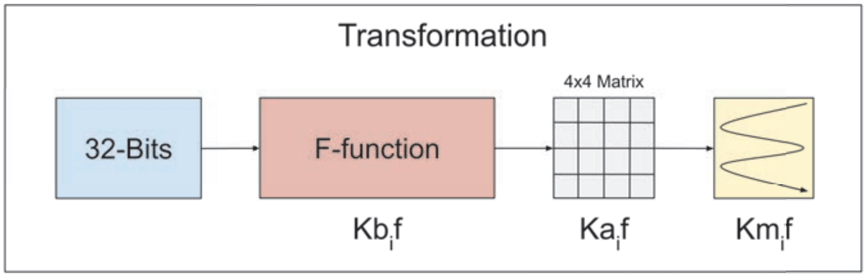

Figure 6 Key expansion Transformation function.

- To remove the data originality, the operation is applied at the end of each round swapping.
- Bitwise XNOR is performed between the key generated for each round and Px_{0-31}, and the same is applied on the respective round key and Px_{96-127} resulting in Ro_{11} and Ro_{14}. The output of the above operations is then passed into f-function resulting in Ef_{l1} and Ef_{r1}, and then Bitwise XOR function is applied between Ef_{l1} & Px_{64-95} to obtain Ro_{12} and Ef_{r1} & Px_{32-63} to obtain Ro_{13}.
- Finally, shuffling of adjacent 32 bits are done

$$Ro_{i,j} = \begin{cases} Px_{i,j} \odot K_i; & j = 1\&4 \\ Px_{i,j+1} \oplus Ef_{li}; & j = 2 \\ Px_{i,j-1} \oplus Ef_{ri}; & j = 3 \end{cases} \tag{1}$$

- The above steps are repeated for the other rounds. Results of the last round are combined to get cipher text. $Ciphertext = R_{51} + +R_{52} + +R + 53 + +R_{54}$

4 Results and Discussion

Passwords are weak and can be cracked or obtained by some or other attacks like phishing attacks. One-time password-based systems can increase security by acting as a two-factor authentication system that validates the OTP user enters with the server, generated after validating the password. The added advantage of time-based OTP is that they are not dependent on any external infrastructure like SMS or email, which requires internet connectivity which might compromise the integrity of the system due to the fault of the service on which the OTP is reliant, like sim spoofing. Since T-OTPs are based on timestamps and expire after a certain period. Thus making it almost impossible for an intruder to gain unauthorized access. The time interval can be narrowed down to increase the network's security.

The enhanced key exchange security mechanism using RSA increases the speed of key generation, decryption of ciphertext, and very marginal improvements in encryption of plain text. This algorithm provides the same security mechanism of key exchange but for a low computational cost making it ideal for low-powered IoT devices. Even though being a lightweight algorithm, it does not compromise the security in any way, allowing for passing keys over an unsecured communication channel. With the utilization of three integral primes and the Chinese remainder theorem, more speed of key generation

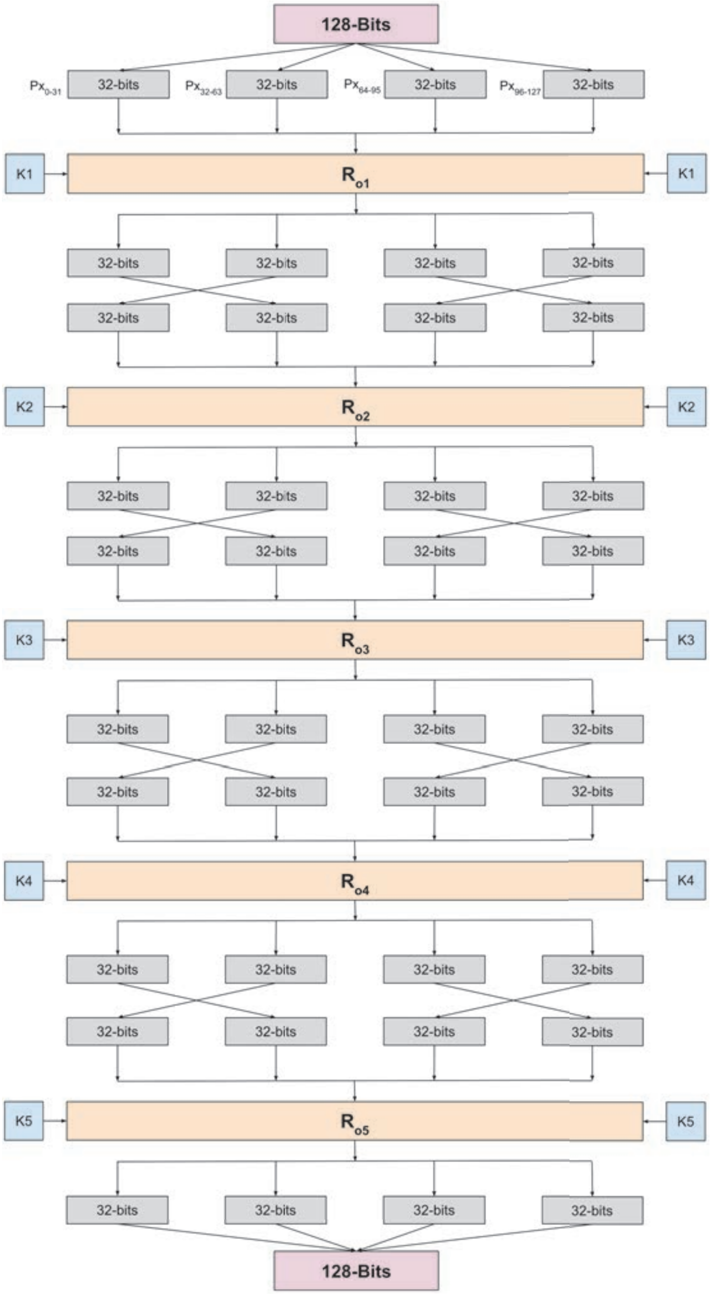

Figure 7 Flow diagram of each round in encryption.

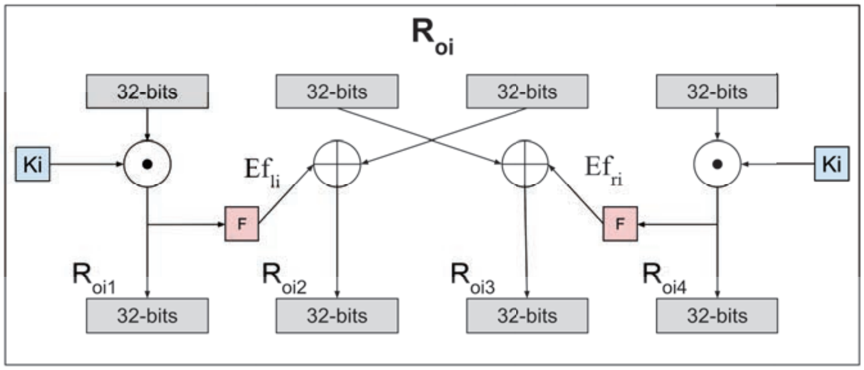

Figure 8 Encryption flow diagram.

Table 4 Encryption and decryption time

Filesize	Encryption Time (ms)	Decryption Time (ms)
8KB	179 ± 1.2	177 ± 1.11
12KB	267 ± 1.77	263 ± 1.62
24KB	551 ± 2.91	534 ± 2.55
64KB	1450 ± 3.51	1420 ± 3.29
128KB	2890 ± 21.6	2810 ± 18.87
256KB	5770 ± 28.4	5730 ± 24.59

and decryption than standard RSA is achieved. Also another advantage is the prevention of attacks involving mathematical factorization since the addition of the third prime adds complexity to the algorithm making it much more challenging to factorize and get back the originally considered primes. It also prevents various other attacks, namely timing attack, modulus attack, plaintext attack, and ciphertext attack.

This SIT encryption algorithm is well suited for the data security of IoT devices as the proposed algorithm relates to the category of lightweight cryptography. It ensures minimum memory and energy consumption. This algorithm provides the same security level as other encryption algorithms but with a low computational cost suited for IoT devices. Even though the algorithm only consists of only five rounds of encryption, it ensures sufficient confusion and diffusion of Shannon's properties for ideal security. The SIT algorithm overcomes weak keys as the algorithm does not use the actual key in the encryption process. Instead, it is first XOR-ed and then passed into an f-function. In the f-function, sufficient confusion and diffusion are created to diminish the bias in selecting a key. It also overcomes the problem of related keys. The given algorithm's key expansion is fast and non-linear diffusion of cipher key difference of that rounds. The presented algorithm can also stand against some well-known attacks known as Interpolation and square attacks.

5 Conclusion

The proposed three-module system implements user authentication using time-based OTP, exchanges keys in unsecured communication channel using a lightweight RSA algorithm using three prime numbers, and finally, data encryption using SIT algorithm. These modules form an end-to-end secured IoT system that ensures the security of the system is never compromised using lightweight cryptographic algorithms. This end-to-end security solution can be adapted to any low-power IoT device to communicate over the internet securely. Thus, the proposed lightweight algorithms are suitable solutions to the current security challenges faced by IoT devices. The proposed system combines multiple lightweight cryptographic solutions into one shared platform, from secure authentication to data encryption for transmission and storage, ensuring end-to-end security for IoT devices.

References

[1] H. Suo, J. Wan, C. Zou and J. Liu. Security in the internet of things: a review. In *Proceedings of the International conference on computer science and electronics engineering, IEEE*, 3: 648–651, 2012.

[2] A. M. Mohamad Al-Aboosi, S. Kamil, S. N. H. Sheikh Abdullah and K. A. Zainol Ariffin. Lightweight Cryptography for Resource Constraint Devices: Challenges and Recommendation. In *Proceedings of the 3rd International Cyber Resilience Conference(CRC), IEEE*, 1–6, 2021.

[3] S. Misra, M. Maheswaran, S. Hashmi. *Security challenges and approaches in internet of things*. Cham: Springer International Publishing, 2017.

[4] I. K. Dutta, B. Ghosh and M. Bayoumi. Lightweight cryptography for internet of insecure things: A survey. In *Proceedings of the 9th Annual Computing and Communication Workshop and Conference(CCWC), IEEE*, 0475–0481, 2019.

[5] P. Shah, M. Arora and K. Adhvaryu. Lightweight Cryptography Algorithms in IoT-A Study. In *Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), IEEE*, 332–336, 2020.

[6] S. A. Kumar, T. Vealey and H. Srivastava. Security in internet of things: Challenges, solutions and future directions. In *Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS), IEEE*, 5772–5781, 2016.

[7] N. A. Gunathilake, A. Al-Dubai and W. J. Buchana. Recent Advances and Trends in Lightweight Cryptography for IoT Security. In *Proceedings of the 16th International Conference on Network and Service Management(CNSM), IEEE*, 1–5, 2020.

[8] M. Katagi and S. Moriai. "Lightweight cryptography for the internet of things." Sony Corporation, 7–10, 2008.

[9] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann and L. Uhsadel. A survey of lightweight-cryptography implementations. *IEEE Design & Test of Computers*, 24(6):522—533, 2007.

[10] D. M'Raihi, David, S. Machani, M. Pei, and J. Rydell. Totp: Time-based one-time password algorithm, *Internet Engineering Task Force*, RFC: 6238, 2011.

[11] M'Raihi, David, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen. Hotp: An hmac-based one-time password algorithm. In *The Internet Society, Network Working Group*. RFC4226, 2005.

[12] M. L. T. Uymatiao and W. E. S. Yu. Time-based OTP authentication via secure tunnel (TOAST): A mobile TOTP scheme using TLS seed exchange and encrypted offline keystore. In *Proceedings of the 4th IEEE International Conference on Information Science and Technology, IEEE*, 225–229, 2014.

[13] D. Kumar, A. Agrawal and P. Goyal. Efficiently improving the security of OTP. In *Proceedings of the International Conference on Advances in Computer Engineering and Applications, IEEE*, 912–915, 2015.

[14] V. L. Shivraj, M. A. Rajan, M. Singh and P. Balamuralidhar. One time password authentication scheme based on elliptic curves for Internet of Things (IoT). In *Proceedings of the 5th National Symposium on Information Technology: Towards New Smart World, IEEE*, 1–6, 2015.

[15] K. S. Roy and H. K. Kalita. A survey on authentication schemes in IoT. In *Proceedings of the International Conference on Information Technology(ICIT), IEEE*, 202–207, 2017.

[16] M. Abd Zaid, Mustafa, and S. Hassan. Lightweight RSA Algorithm Using Three Prime Numbers. *Journal of Engineering and Applied Sciences*, 14(5): 9032–9035, 2019.

[17] J. Sahu, V. Singh, V. Sahu, and A. Chopra. An enhanced version of RSA to increase the security. *Journal of Network Communication and Emerging Technologies*, 7(4), 1–4, 2017.

[18] T. K. Goyal and V. Sahula. Lightweight security algorithm for low power IoT devices. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE*, 1725–1729, 2016.

[19] V.G. Kumar Kiran, S.J. Mascarenhas, S. Kumar, J. Pais Viven Rakesh. Design and implementation of Tiny encryption algorithm. *International Journal of Engineering Research and Applications*, 5(6): 94–97, 2015.

[20] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. SIMON and SPECK: Block Ciphers for the Internet of Things. *NIST Lightweight Cryptography Workshop.*, 1–15, 2015.

[21] M. Usman, I. Ahmed, M.I. Aslam, S. Khan, and U.A. Shah. SIT: a lightweight encryption algorithm for secure internet of things. *International Journal of Advanced Computer Science and Applications*, 8(1): 1–10, 2017.

[22] F. Thabit, S. Alhomdy, A. H. Al-Ahdal and S. Jagtap. A new lightweight cryptographic algorithm for enhancing data security in cloud computing. *Global Transitions Proceedings*, 2(1):91–99, 2021.

[23] C. A. Lara-Nino, A. Diaz-Perez and M. Morales-Sandoval. Elliptic Curve Lightweight Cryptography: A Survey. *IEEE Access*, 6: 72514–72550, 2018.

[24] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu & N. Kumar. A secure authentication scheme based on elliptic curve cryptography for IoT and cloud servers. *The Journal of Supercomputing*, 74(12), 6428–6453, 2018.

[25] A. Karati, S. H. Islam & M. Karuppiah. Provably secure and lightweight certificateless signature scheme for IIoT environments. *IEEE Transactions on Industrial Informatics*, 14(8), 3701–3711, 2018.

[26] S. Basu, M. Karuppiah, K. Selvakumar, K. C. Li, S. H. Islam, M. M. Hassan & M. Z. A. Bhuiyan. An intelligent/cognitive model of task scheduling for IoT applications in cloud computing environment. *Future Generation Computer Systems*, 88, 254–261, 2018.

[27] P. Punithavathi, S. Geetha, M. Karuppiah, S. H. Islam, M. M. Hassan & K. K. R. Choo. A lightweight machine learning-based authentication framework for smart IoT devices. *Information Sciences*, 484, 255–268, 2019.

[28] M. Naeem, S. A. Chaudhry, K. Mahmood, M. Karuppiah & S. Kumari. A scalable and secure RFID mutual authentication protocol using ECC for Internet of Things. *International Journal of Communication Systems*, 33(13), e3906, 2020.

[29] M. Karuppiah, A. K. Das, X. Li, S. Kumari, F. Wu, S. A. Chaudhry & R. Niranchana. Secure remote user mutual authentication scheme with key agreement for cloud environment. *Mobile Networks and Applications*, 24(3), 1046–1062, 2019.

Biographies

V. N. Hemanth Kollipara is a senior year student pursuing his Bachelor's in Computer Science at Vellore Institute of Technology, Vellore. He is passionate about working with emerging technologies, research, and building beneficial applications. His research interests include IoT, Machine Learning, Deep Learning, Computer Vision, and Intelligent systems.

Sai Koushik Kalakota is a senior year student pursuing his Bachelor's in Computer Science at Vellore Institute of Technology, Vellore. His research interests include Machine Learning, Deep Learning, IoT, and Natural Language Processing.

Sujith Chamarthi is a senior year student pursuing his Bachelor's in Computer Science at Vellore Institute of Technology, Vellore. His research interests include IoT, Embedded Systems, and Cyber security.

S. Ramani is an Assistant Professor (Senior) in the School of Computing Science and Engineering at Vellore Institute of Technology (VIT), Vellore, India. He received his B.E in Computer Science and Engineering from MNM Jain Engineering College, Madras University and M. Tech Computer Science and Engineering from Bharathidasan University, and Ph.D. in Computer Science and Engineering from Vellore Institute of Technology (VIT), Vellore. He has 13+ years of experience in teaching and 2 years of experience in the Consultancy and Software Industry. He has published about 40 research papers in International Journals on Machine Learning, Nature-Inspired Algorithms, Cyber Security and Health Care. His research interest includes Data Mining, Machine learning, Database Systems, Optimization Techniques, and Cyber Security. He is a life member of the Computer Society of India (CSI), IEEE, and other technical societies.

Preeti Malik is working as an Assistant Professor in Department of Computer Science and Engineering in Graphic Era (Deemed to be University), Dehradun. She received her doctorate from Gurukul Kangri University in 2017 and Master of Computer Applications in 2011. She has more than 5 years of research and teaching experience. She has authored the book entitled "Algorithms" and edited books. She has published more than 15 research papers in National and International Journals/conferences. Her research interests include Mobile Agents, Cyber security, Software Testing and Reliability.

Marimuthu Karuppiah received the B.E. degree in computer science and engineering from Madurai Kamaraj University, Madurai, India, in 2003, the M.E. degree in computer science and engineering from Anna University, Chennai, India, in 2005, and the Ph.D. degree in computer science and engineering from VIT University, Vellore, India, in 2015. He was an Associate Professor with VIT University, Vellore, India. He is currently a Professor with the Department of Computing Science and Engineering, SRM Institute of Science and Technology, Delhi-NCR Campus, India. He has published more than 50 research articles in SCI indexed journals. Also, he has published more than 30 research articles in SCOPUS indexed journals and international conferences. His current research interests include cryptography and wireless network security, in particular, authentication and encryption schemes. He is a Life Member of the Cryptology Research Society of India (CRSI) and the Computer Society of India (CSI), and a member of ACM.