Journal of Ambient Intelligence and Humanized Computing (2021) 12:2033–2051
https://doi.org/10.1007/s12652-020-02303-5

ORIGINAL RESEARCH

Improving the security of internet of things using cryptographic algorithms: a case of smart irrigation systems

Seyyed Keyvan Mousavi[1] · Ali Ghaffari[2] · Sina Besharat[3] · Hamed Afshari[4]

Received: 10 February 2020 / Accepted: 2 July 2020 / Published online: 8 July 2020

Abstract

Internet of Things (IoT) as a ubiquitous paradigm is a new concept in Information and Communications Technology (ICT) and has the ability to connect wireless and mobile embedded devices and things to the Internet. IoT is emerging as a key component of the Internet and a vital infrastructure for millions of smart and interconnected objects that are potentially vulnerable to different attacks. Thus, the security of resource-constrained devices in IoT is highly important. As an important solution, cryptographic algorithms are used to provide confidentiality and integrity of the transmitted data between the sender and receiver. Hence, this paper proposes a new hybrid cryptographic algorithm based on Rivest cipher (RC4), Elliptic-Curve Cryptography (ECC), and Secure Hash Algorithm (SHA-256) to protect sensitive information in IoT-based smart irrigation systems. In this paper, the RC4 key is encrypted by the ECC algorithm, and the output of this encryption process is transformed to SHA-256 for hashing and generating enigmatic data. SHA-256 algorithm encrypts RC4 based cipher text to improve data integrity. Comprehensive analysis and simulation results indicate that the proposed scheme is secure to various known attacks such as the Man-in-the-middle (MiM) attack, and has a better performance than other cryptographic algorithms. Also, the obtained results confirm the effectiveness of the proposed model and robustness in order to confidentiality based on analyzing secrecy.

Keywords IoT · Security · Cryptography · RC4 · ECC · SHA-256

1 Introduction

IoT is a new Internet-based technology that includes millions of interconnected embedded smart things. This technology integrates various smart devices with embedded sensors that interact with each other without human intervention (Alaba et al. 2017; Jazebi and Ghaffari 2020; Singh et al. 2017). The security of Wireless Sensor Networks (WSNs) (Azari and Ghaffari 2015; Ghaffari 2014; Ghaffari and Rahmani 2008; Ghaffari and Takanloo 2011; KeyKhosravi et al. 2010; Khabiri and Ghaffari 2018; Mohammadi and Ghaffari 2015) has become a critical challenge due to the widespread deployment of this technology in IoT (Liu et al. 2016). In modern farming, the watering process is one of the most important processes due to shortage of sweet water in most of the area of the world (Burton et al. 2018). Hence, security is the main challenge in IoT devices and the implementation of IoT services depends on protecting this technology against unwanted threats and security attacks (Sharma and Kalra 2018). Cryptography schemes prepare a fundamental security layer for data and various applications. Recently,

✉ Ali Ghaffari
a.ghaffari@iaut.ac.ir

Seyyed Keyvan Mousavi
mosavikeyvan90@gmail.com

Sina Besharat
s.besharat@urmia.ac.ir

Hamed Afshari
h.afshari@iaurmia.ac.ir

[1] Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

[2] Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran

[3] Department of Water Engineering, Faculty of Agricultural Sciences, Urmia University, Urmia, Iran

[4] Department of Mechanical and Bio Mechanical Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

with the rise of IoT, we need lightweight and efficient cryptographic schemes (Saha et al. 2019).

1.1 Motivation

Misuse of irrigation information, Distributed Denial of Service (DDoS), and Side-Channel Attacks (SCAs) are some common IoT threats (Agale and Gaikwad 2017). Smart irrigation systems use humidity-meter sensors to evaluate whether through soil moisture and the chance of rainfall (Gulati and Thakur 2018). They have a flexible design that allows farmers to appropriately determine irrigation time and plant moisture requirement or even delay irrigation when the chance of rain is high, which saves water and helps boost the harvest (Hendrawan et al. 2019). Despite the advantages offered by smart irrigation systems, there are security challenges that vary by performance and the environment (Babayiğit and Büyükpatpat 2019). IoT infrastructure facilitates the expansion of public spaces and offers a wide range of programmable services, but is also prone to many threats and security attacks.

To have successful access control on the IoT, several principles and features must be considered (Qiu et al. 2020b). The most important features that should be considered in access control are: confidentiality, data accuracy, and information access levels. Access control method can effectively monitor the access activities of resources, and ensure authorized users to access information resources under legitimate conditions (Li et al. 2019; Tian et al. 2020b).

In IoT environment, security of devices, communication protocols and different layers must be considered (Tian et al. 2020a). Unfortunately, a significant number of IoT devices have security vulnerabilities and are vulnerable, which can allow hackers and malicious individuals to damage and disrupt the operation of these devices and destroy users' privacy (Qiu et al. 2020a). Due to the nature and characteristics of the sensors used in IoT and the insecure nature of the Internet, the IoT is vulnerable to various attacks, especially internal routing attacks. The IoT infrastructure should support security of data, software, hardware, and physical devices (Tian et al. 2019). Ensuring data security is a very important factor in building trust on users and using the IoT platform. Users need to make sure that the IoT is secure enough to carry out security activities against threats. Therefore, considering the confidentiality of data means the formation of trust in IoT (Chen et al. 2019).

1.2 Main contributions

The aim of this paper is to develop a new model for protecting sensitive data of IoT based irrigation system. This paper proposes a novel model based on RC4 (Stinson 1995), ECC (Miller 1986), and SHA-256 (Gilbert and Handschuh 2004; Yoshida and Biryukov 2006) algorithms to preserve IoT security. In the proposed model, data are first encrypted by RC4 and ECC, and then transformed into a hash state using SHA-256.

The main contributions of this paper are as follows:

1) Design secure and efficient data transport scheme in the IoT environment.
2) Increasing security with encryption of the RC4 key by ECC.
3) Encrypting RC4 based cipher text using the SHA-256 algorithm to improve data integrity.
4) Improving encryption/decryption time, throughput and desirable confidentiality based on secrecy analysis.

The irrigation sensors, smartphone, data collection, public communication network and the IoT network are exposed to different security threats and most of the time the main reason was the vulnerabilities from the data manipulation. There are various vulnerabilities, threats and attacks in IoT-based smart irrigation system that proposed model prevent to their influence.

1.3 Organization of the paper

The rest of the paper is organized as follows: Sect. 2 provides a review of the literature. Section 3, describes the proposed model based on RC4, ECC, and SHA-256. Section 4, evaluates and compares the results of the proposed model. Finally, Sect. 5 concludes the paper and provides some future works.

2 Related works

To tackle security problems in IoT environment, researchers have presented various and numerous security solutions using cryptography schemes. This section describes previous and related works in the area of IoT security.

KP-ABE algorithm is used as an appropriate security mechanism for heterogeneous encryption, and is widely deployed for implementing access control solutions. Touati and Challal (2016) used this algorithm for IoT security through three phases: initialization and key generation, data encryption, data decryption and extraction. Encryption is vital for privacy in healthcare plans. IoT demands an efficient and low-energy cryptographic algorithm, Khader et al. (Khader et al. 2017) used modified AES algorithm to propose a low-energy cryptographic mechanism for IoT sensors. AES is a common method that uses a 128, or 192, or 256-bits key for encryption and decryption.

A hidden ciphertext policy Attribute-based Encryption (ABE) was proposed in (Belguith et al. 2018) that preserved privacy and had low processing overhead.

Similarly, an ABE-based model has proposed in (Yang et al. 2017) for health system to prevent unauthorized access and protected security. In (Yao et al. 2015), a cryptographic scheme based on ABE and ECC has proposed to deal with security and privacy issues in IoT. Results demonstrated high productivity and low computational costs of the proposed model. The ABE to prevent hidden access to IoT data was offered in (Han et al. 2018). A new CP-ABE scheme has proposed which can protect the user's attribute values against the attacks. A KP-based encryption model has proposed for access control in IoT (Lee et al. 2015). A biometric system to develop healthcare system based on IoT with high data accessibility was offered that identifies users by certain physiologic attribute vectors. The attributes vector is saved in database. This system features a high confidence coefficient (Hamidi 2019).

Privacy and security issues of IoT users were considered in (Wei and Zhou 2018). One essential problem is access to server to obtain information through mobile phones. To this end, homomorphic encryption and ABE were used. Homomorphic encryption allows for direct encryption of an infinite number of calculations without disclosing the secret keys. Diffie–Hellman (DH) encryption is also used for IoT security and privacy. It uses RSA heterogeneous encryption to generate keys between the application and the server, and then uses symmetric AES algorithm to encrypt communications between them by the generated key (Xu et al. 2019a). Diffie–Hellman encryption is used to deal with security and privacy challenges in IoT cloud. A fast encryption protocol has proposed in cloud servers (Wu et al. 2018). An AES-based encryption scheme with a 128-bit key for building a secure session between things was proposed with high computational efficiency, los costs and proved strong against different attacks like service denial, response attack, and physical manipulation attack (Jan et al. 2019). RSA algorithm is used for improving the security of IoT information. It is noticeably fast and is applied in many electronics (Hu 2011). RSA is also used for security infrastructure of IoT (Kothmayr et al. 2012). It is mounted on a hardware platform with low power for IoT.

Data security is also important in cloud computing. Some mechanisms like access control are used for this purpose. In (Pant et al. 2015), RSA was used for protecting data while sharing or storing data in cloud environment. Security of MQTT protocol was also provided by RSA. It is a binary and lightweight machine-to-machine protocol to transmit data with high confidence to resource-constrained clients. As a data-centric protocol, MQTT is better than other existing web protocols like HTTP because it has the least package overhead and is suitable for short message transfer while HTTP is document-centered and is used for sending video files, etc. (Mektoubi et al. 2016).

IoT implementation in healthcare centers is usually based on radio frequency. RFID authenticates RFID tags and readers. An authentication scheme based on ECC&RSA between RFID tags and readers and the server was proposed in (Jisha and Philip 2016) to promote data security. RSA, AES, and TDES have been proposed for IoT data encryption (Matsemela et al. 2017). A proper security algorithm for IoT is adopted in terms of time, memory, and processing. Data are encrypted, decrypted and encrypted once again, yielding a 168-bit key that is long enough for many sensitive data. Thus, Triple-DES is a stronger standard than DES. Results of testing the security algorithms show that AES has a better performance in terms of computational time, memory use and processing. The large key length in AES ensures higher protection levels. However, it has some deficits such as inability to authenticate and encrypt different data types including videos, photos, and audio files. Therefore, we used a combination of the above algorithms.

A security scheme based on RSA and ECC for IoT data has proposed that used RSA security blocks to promote security level (Chhabra and Arora 2017). A hybrid and secure algorithm for data storage and transmission in IoT cloud was proposed where the data are encrypted by AES before transmission. AES key is encrypted using RSA system. Moreover, RSA encryption key with authorized users is shared through email (Chandu et al. 2017). IoT are vulnerable to malware attacks such as buffer overflows, denial-of-service, and Trojan horse, worms, viruses and malicious codes. These attacks are modified by RSA and AES algorithms (Abinaya et al. 2018). TLS protocol was tested by RSA in terms of security measures, scalability, power consumption and data usage. The results were the compared to ECC. Key length in ECC provides a reasonable security level. TLS is mostly applied in transmission layer in wired and mobile networks and is used to provide a secure communication. Its specific mechanisms help establish data confidentiality, integrity and privacy (Suárez-Albela et al. 2018).

Network layer in IoT are vulnerable to probable attacks which disturb the connection between devices in the absence of encryption algorithms. To solve this, an RSA-based access protocol was designed that offers a safe interface in network layer. In such cases, the controller and the recipient device verify each other and generate a session key for next communications (Mao et al. 2018). RSA and DES encryption techniques are used for encrypting data frames. DES is a mathematics algorithm used for encrypting and decrypting coded information. It is computationally efficient and is executed by slow processors. 64-bits data are encrypted and decrypted by 56-bits keys. RSA is used to enhance cryptography and privacy processes (Hussain et al. 2017). CP-ABE-based RSA was proposed to guarantee a secure communication between IoT server and devices (Odelu et al. 2017). In (Xu et al.

2019b), an access control based on attributes on IoT cloud has used that allows the data owners to effectively manage the validity of data users and block unauthorized users.

Table 1 summarizes the proposed models of IoT security.

3 Proposed scheme

Security is a crucial challenge in IoT based irritation systems because they include databases, information files, and interconnected sensors and devices. This paper proposes a secure scheme for IoT environment in smart irrigation systems using RC4 and ECC algorithms. In the proposed scheme, RC4 and ECC algorithms are used for encryption and SHA-256 is used for hashing the irrigation data. Figure 1 demonstrates the IoT-Based smart irrigation system.

3.1 ECC scheme

ECC is an algebraic structure of elliptic curves scheme on finite fields. The basic advantage of ECC to other asymmetric algorithms is the small key length that improves processing time. The security of ECC is based on an exponential discrete logarithm that is hard to break. ECC is applied in finite fields. Assume p is a prime number and Fp. as a set of integers smaller than p, the two-dimensional coordinate of elliptical bend E is defined by Eq. (1) as follows (Miller 1986):

$$\mathbf{y^2 = x^3 + ax + b}. \tag{1}$$

where $a, b \in F_p$. and $4a^3 + 27b^2 \neq 0(modp)$.. If a point in (x, y) is te in Eq. (1), it belongs to elliptical bend. Moreover, $E(F_p)$. is a set of all points on the elliptical bend, and Q is a point on E. in ECC encryption, a random number x in the interval [1, n-1] from Fp. field is selected as the private key. Then, public key H is calculated as $H = x.Q$. In ECC,

Table 1 Proposed models for IoT security

References	Model	Encryption algorithm	confidentiality	Trust	Authentication	Access control
Touati and Challal (2016)	Cloud IoT	ABE	✓	✓	–	–
Khader et al. (2017)	IoT in healthcare system	AES	✓	✓	–	–
Belguith et al. (2018)	IoT data encryption based on cloud computations	ABE	✓	✓	✓	✓
Yang et al. (2017)	IoT in healthcare system	ABE	✓	✓	–	–
Yao et al. (2015)	Data encryption in IoT	ABE&ECC	✓	✓	–	–
Han et al. (2018)	Data encryption in IoT	ABE	✓	✓	✓	✓
Lee et al. (2015)	IoT data encryption based on cloud computations	ABE	✓	✓	–	✓
Hamidi (2019)	IoT in healthcare system	biometric	✓	✓	✓	✓
Wei and Zhou (2018)	Data encryption in IoT	ABE + Homomorphic	✓	✓	✓	✓
Xu et al. (2019a)	Data encryption in IoT	DIFFIE-Hellman	✓	✓	✓	–
Wu et al. (2018)	Cloud IoT	DIFFIE-Hellman	✓	✓	–	–
Jan et al. (2019)	Data encryption in IoT	AES	✓	✓	✓	✓
Hu (2011)	Data encryption in IoT	RSA	✓	✓	–	–
Kothmayr et al. (2012)	Cloud IoT	RSA	✓	✓	✓	✓
Pant et al. (2015)	Cloud IoT	RSA	✓	✓	–	–
Mektoubi et al. (2016)	Data encryption in IoT	RSA	✓	✓	✓	✓
Jisha and Philip (2016)	Data encryption in IoT	ECC + RSA	✓	✓	–	–
Matsemela et al. (2017)	Data encryption in IoT	RSA&AES&EDES	✓	✓	✓	✓
Chhabra and Arora (2017)	Data encryption in IoT	ECC & RSA	✓	✓	–	–
Chandu et al. (2017)	Data encryption in IoT	AES + RSA	✓	✓	–	–
Abinaya et al. (2018)	Data encryption in IoT	RSA	✓	✓	–	–
Suárez-Albela et al. (2018)	Data encryption in IoT	RSA	✓	✓	–	–
Mao et al. (2018)	Data encryption in IoT	RSA	✓	✓	✓	–
Hussain et al. (2017)	Data encryption in IoT	RSA&DES	✓	✓	–	–
Odelu et al. (2017)	Cloud IoT	ABE	✓	✓	–	-
Xu et al. (2019b)	Cloud IoT	ABE	✓	✓	–	

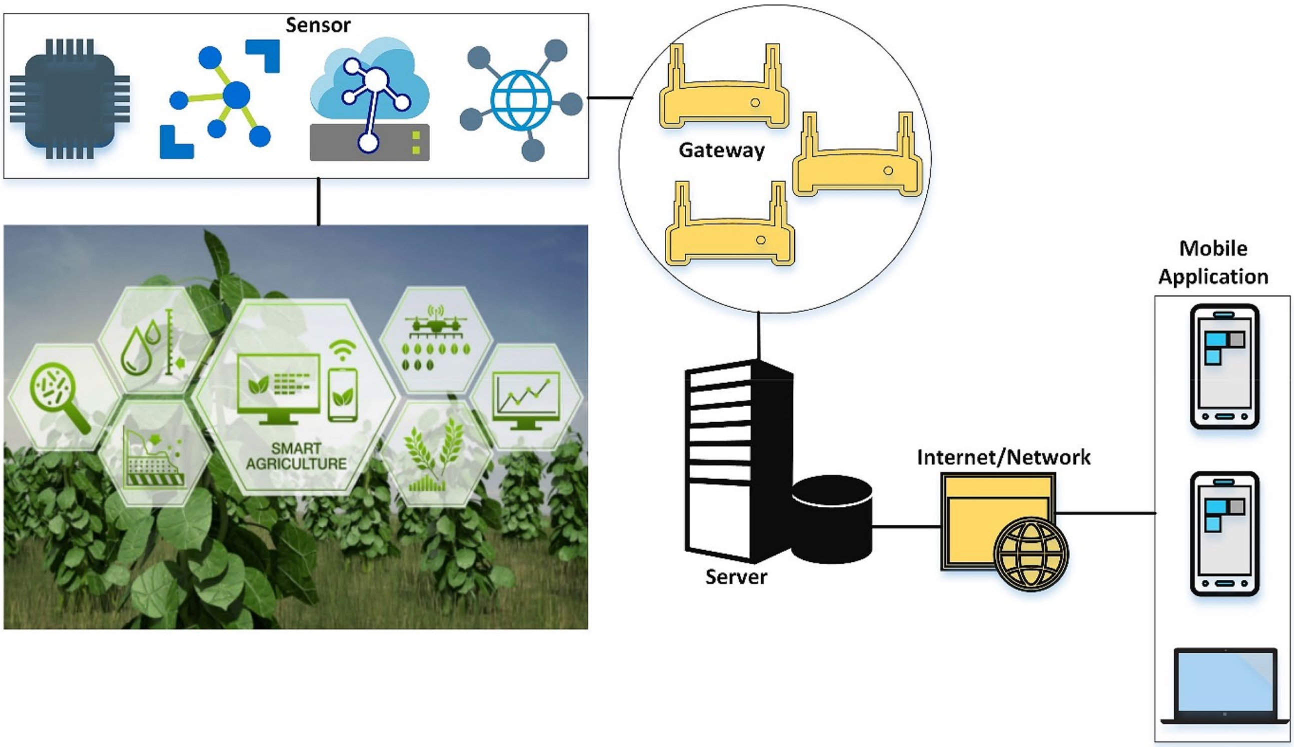


Fig. 1 IoT-Based smart irrigation system

a character is converted to bites that are then converted to (x, y) bites. These pnts are encrypted on an elliptical bend which is finally converted to bites. Encryption of an elliptical bend is performed as the following:

(1) Initialization: Sides of the elliptical bend E and generator Q of order p agree with each other.
(2) Pubic key generation: The public key is generated as $H = x.Q$. and H is shared as the public key between the sender and receiver. x is the private key that the sender uses for decryption.
(3) Encryption: To encrypt the message $m \in EQ$., random number r is selected and encrypted by Eq. (2). The data owner sends C to the receiver in order to deliver message m (Miller 1986).

$$\boldsymbol{C} = \mathbf{Enc}(\boldsymbol{m}) = \begin{cases} \boldsymbol{c}_1 = \boldsymbol{rQ} \\ \boldsymbol{c}_2 = \boldsymbol{m} + \boldsymbol{rH} \end{cases}. \tag{2}$$

(4) Decryption: The receiver uses C and the private key(x) for the decryption phase tough E (3) (Miller 1986).

$$\mathbf{Dec}(\boldsymbol{C}) = \boldsymbol{c}_2 - \boldsymbol{x}.\boldsymbol{c}_1 = \boldsymbol{m} + \boldsymbol{rH} - \boldsymbol{xrQ} = \boldsymbol{m} + \boldsymbol{rxQ} - \boldsymbol{xrQ} = \boldsymbol{m} \tag{3}$$

3.2 RC4 and SHA-256 algorithms

RC4 algorithm includes two phases: (1) the Key Scheduling Algorithm (KSA) phase and (2) the Pseudo Random number Generation Algorithm (PRGA) phase. KSA phase extends the S-box to 256 bytes. Finally, PRGA phase produces a pseudo-random key stream and XOR encryption with the plain text to form a cipher text. Algorithm-1 and Algorithm-2 define the KSA and PRGA respectively.

Algorithm-1. Key Scheduling Algorithm (KSA)

```
Input: Secret key K
K: key length
Output: Internal state S
j = 0;
  // State Initialization
for i = 0 to N – 1 do
    S [i] = i;
  // State Randomization
for i = 0 to N - 1 do
j= (j + S [i] + K [i mod k]) mod N
Swap (S [i], S [j])
```

Algorithm-2. Pseudo-Random Generation Algorithm (PRGA)

```
Input: Internal state S, generated by KSA
Output: keystream Z
i=0
j= 0
for each new message byte do
    i = (i + 1) mod N
    j = (j + S [i]) mod N
    Swap (S [i], S [j])
    Z= S [(S [i] + S [j]) mod N]
    output Z
```

The SHA-256 scheme includes seven logical functions that work on 32-bits words represented by x, y, and z. SHA-256 input may be a string of 2^{64} with a block size of 512-bits divided to 16 words of 32-bits. Messages are divided into blocks. A, B, C, D, E, F, G, and H variables are used as initial states of hashing. SHR operator moves data bits to the right. It shifts all target operand bits to the right. ROTR rotates its target bit operands to the right. A bit exported from the right enters the operand from the left.

3.3 Proposed secure scheme

The proposed scheme uses ECC to encrypt the key of RC4 algorithm. Then, the encrypted key of RC4 is transformed to SHA-256 scheme for hashing purpose and generating an enigmatic data. SHA-256 algorithm hashes the RC4 based cipher text to improve data integrity. Figure 2 depicts the flowchart of the proposed scheme based on hybrid of RC4, ECC, and SHA-256.

Due to small key size of ECC, this algorithm is appropriate for encrypting information of IoT sensors. In the proposed scheme, a combination of RC4 and ECC is used for high security levels. Figure 3 shows the encryption/decryption steps of the proposed model.

Weak Key Scheduling Algorithm (KSA) will make the encrypted data under risk. To secure data transmission, a secure channel must be guaranteed between the user and the server of IoT. In this regard, lightweight ECC is a critical component for constructing the security system of IoT (Liu et al. 2016). Encryption is done on a data file $D=(M_1,\ldots, M_n)$ where M is the text. The data must be encrypted using the encryption RC4 with a key K' (Encrypted key by ECC) where $K \neq K'$. To preserve the RC4 key, the proposed scheme uses ECC encryption. Finally, SHA-256 scheme use for hashing the encrypted data.

4 Performance evaluation

The simulation experiments will compare the confidentiality of the proposed encryption algorithm, the encryption and decryption time, the encryption and decryption throughput, the average secrecy value and the amount of encrypted data.

The hardware facilities of the simulation experiments are Intel Core i7(2.0 GHZ), 8G memory, equipped with 64-bit Windows 8 operating system, the programming language is C#.NET 2017. Table 2 shows simulation parameters value.

In this paper, small file size, the key size has not important impact on the encryption/decryption time. But it is important for the level of security. In this paper, AES-128 algorithm (128 bits key size) is used.

4.1 Cipher text Size

Figure 4 shows a comparison of the plaintext size and ciphertext size based on different models. The X-axis represents the plaintext size and the Y-axis represents the cipher text size. In the proposed model, the ciphertext size is smaller than other models, which indicates an important improvement for the proposed model. The size of the data is changed from 20 to 1000 KB and ciphertext is calculated.

From Fig. 4, it is clear that the cipher text file size for the proposed model is 1370 KB for 1000 KB, 3DES&ECC&SHA-256 takes 2436 KB, RC4&3DES&SHA-256 takes 1827 KB, AES&RC4&SHA-256 takes 1827 KB, AES&3DES &SHA-256 takes 2436 KB, RC4&AES&SHA-256 takes 1827 KB.

4.2 Encryption/decryption time

Table 3 compares encryption time of the proposed model and other models. It is clear the proposed model is the most time-efficient model, and AES&3DES&SHA-256 has an average shorter encryption time than other models.

Figure 5 shows the chart of encryption time of the proposed model and other models. It is concluded that encryption time is directly related to file size, i.e., larger file demand longer encryption time.

Table 4 compares decryption time of the proposed model and other models. As can be seen, decryption time in the proposed model is shorter than other models. A 1 MB file is decrypted at 97 ms, which is shorter than other models.

Figure 6 shows decryption time of the proposed model and other models based on file size. The proposed model has a better performance than others.

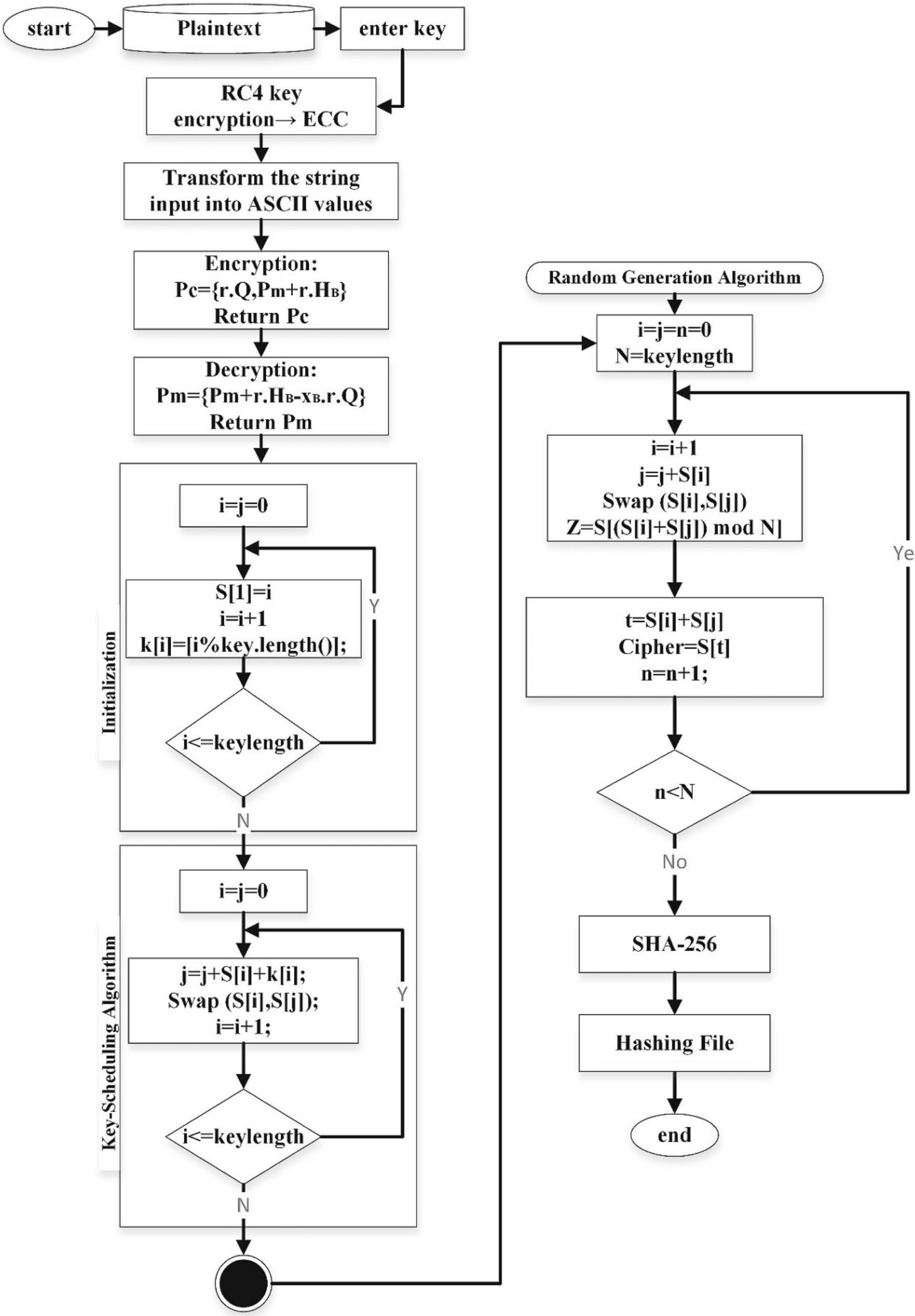
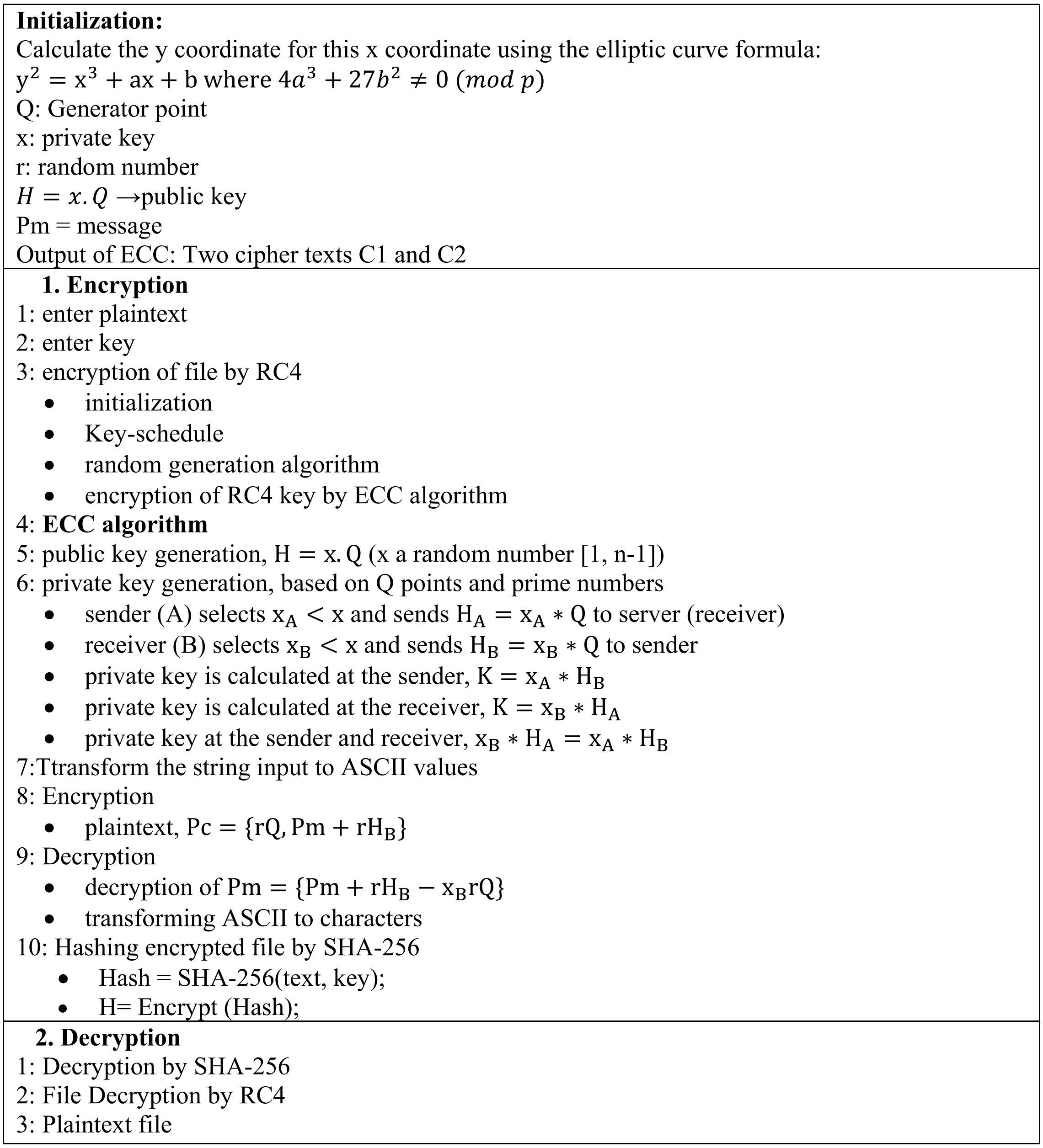


Fig. 2 Flowchart of the proposed scheme



```
Initialization:
Calculate the y coordinate for this x coordinate using the elliptic curve formula:
$y^2 = x^3 + ax + b$ where $4a^3 + 27b^2 \neq 0\ (mod\ p)$
Q: Generator point
x: private key
r: random number
$H = x.Q$ →public key
Pm = message
Output of ECC: Two cipher texts C1 and C2

  1. Encryption
1: enter plaintext
2: enter key
3: encryption of file by RC4
    • initialization
    • Key-schedule
    • random generation algorithm
    • encryption of RC4 key by ECC algorithm
4: ECC algorithm
5: public key generation, $H = x.Q$ (x a random number [1, n-1])
6: private key generation, based on Q points and prime numbers
    • sender (A) selects $x_A < x$ and sends $H_A = x_A * Q$ to server (receiver)
    • receiver (B) selects $x_B < x$ and sends $H_B = x_B * Q$ to sender
    • private key is calculated at the sender, $K = x_A * H_B$
    • private key is calculated at the receiver, $K = x_B * H_A$
    • private key at the sender and receiver, $x_B * H_A = x_A * H_B$
7:Ttransform the string input to ASCII values
8: Encryption
    • plaintext, $Pc = \{rQ, Pm + rH_B\}$
9: Decryption
    • decryption of $Pm = \{Pm + rH_B - x_B rQ\}$
    • transforming ASCII to characters
10: Hashing encrypted file by SHA-256
    • Hash = SHA-256(text, key);
    • H= Encrypt (Hash);

  2. Decryption
1: Decryption by SHA-256
2: File Decryption by RC4
3: Plaintext file
```

Fig. 3 Encryption/ Decryption steps for the proposed scheme

Table 2 Simulation parameters value

	CPU	Intel Core i7(2.00 GHZ)
System Information	RAM	8 GB
	Operating System	Windows 8
	System Type	64 bits
Configuration	C#.NET 2017	Cryptography Class
Models	Key Size (bits)	Block Size (bits)
3DES	128	64
ECC	128	–
RC4	128	S-box (256 bytes)
AES	128	128

Figure 7 compares average encryption/decryption time of the proposed model and other models for different file sizes, such as (20–1000 KB based on average).

4.3 Encryption/decryption throughput

Encryption throughput is calculated based on plaintext divided by total encryption time. Higher throughput indicates algorithm strength and efficiency. Encryption throughput of the proposed model is higher than other models. Table 5 shows encryption throughput of the proposed model. Decryption throughput is calculated based on plaintext

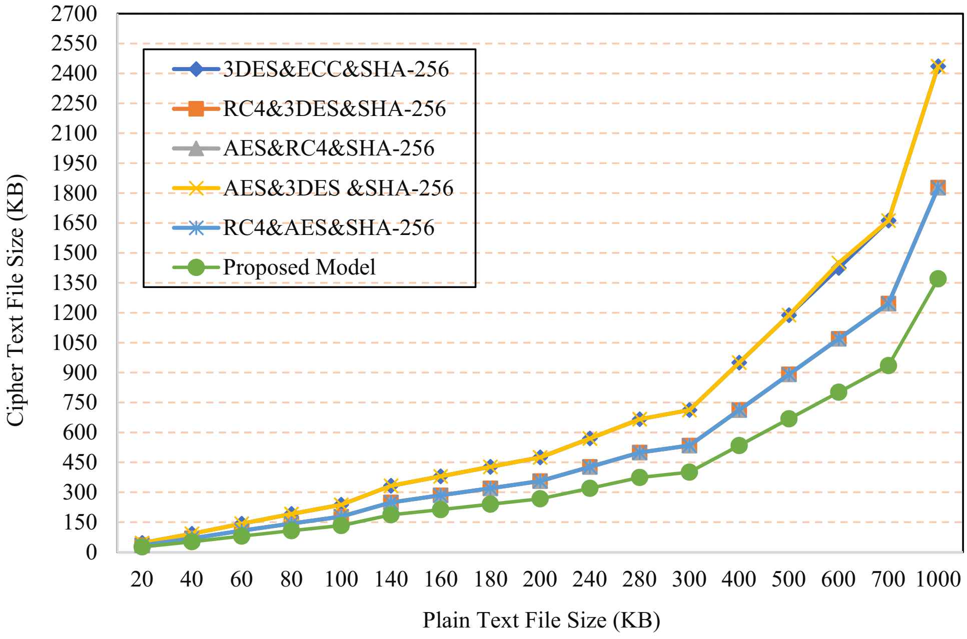


Fig. 4 Comparison of different models based on plaintext and ciphertext

Table 3 A comparison of encryption time of the proposed model and other models

Input file size (KB)	Encryption execution time (ms)					
	3DES & ECC & SHA-256	RC4 & 3DES & SHA-256	AES & RC4 & SHA-256	AES & 3DES & SHA-256	RC4 & AES & SHA-256	Proposed model
20	3	3	7	7	7	2
40	5	7	10	9	9	4
60	7	9	13	13	12	6
80	11	12	16	14	14	8
100	14	15	19	15	17	10
140	19	22	25	21	22	15
160	22	26	28	23	24	17
180	25	30	31	26	27	19
200	27	32	35	28	30	21
240	32	39	41	32	35	26
280	37	46	46	35	40	32
300	42	49	50	38	42	33
400	54	65	65	48	53	43
500	64	79	78	59	65	56
600	80	95	94	70	78	66
700	88	111	108	82	89	76
1000	130	168	165	119	129	109
Total time	660	808	831	639	693	543
Average Time	38.82	47.52	48.88	37.58	40.76	31.94

divided by total decryption time. Decryption throughput of the proposed model is higher than other models. Table 6 shows decryption throughput of the proposed model based on file size. Encryption and decryption throughputs of the proposed model are calculated using Eqs. (4) and (5) as follows:

Fig. 5 Encryption time of the proposed model and other models based on file size

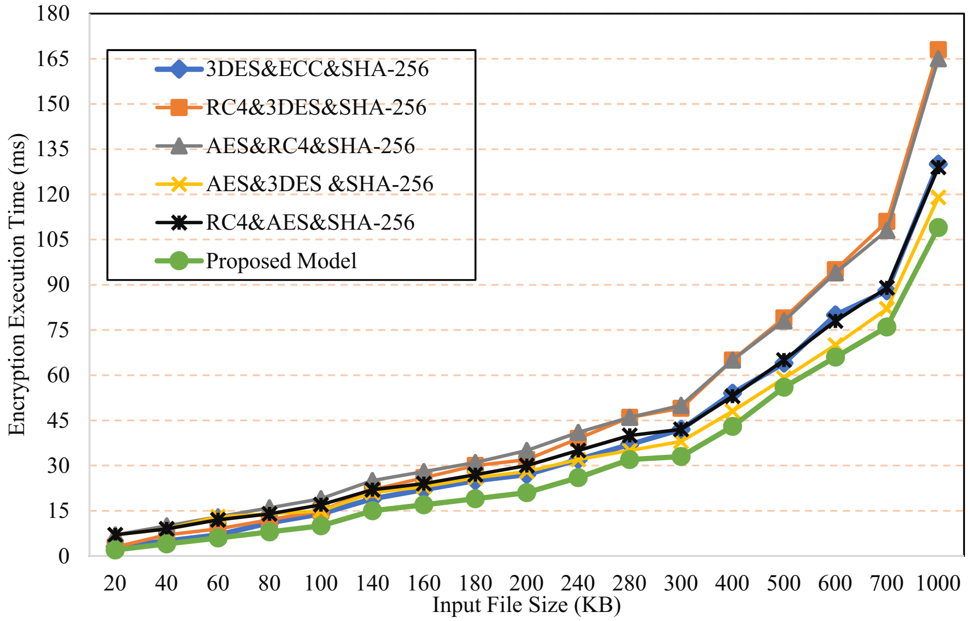


Table 4 A comparison of decryption time of the proposed model and other models

Input file size (KB)	Decryption execution time (ms)					
	3DES & ECC & SHA-256	RC4 & 3DES & SHA-256	AES & RC4 & SHA-256	AES & 3DES & SHA-256	RC4 & AES & SHA-256	Proposed model
20	3	3	7	7	7	2
40	5	6	10	10	9	4
60	7	9	13	13	12	5
80	11	12	16	14	14	7
100	12	15	20	16	16	9
140	19	21	25	19	20	13
160	21	24	28	22	24	15
180	23	27	31	23	26	17
200	26	31	33	27	29	19
240	31	37	38	32	33	23
280	36	43	46	35	36	27
300	40	47	47	36	40	29
400	53	64	60	46	50	39
500	67	75	74	60	60	48
600	79	90	98	67	72	59
700	92	107	107	78	84	66
1000	135	165	164	118	135	97
Total time	660	776	817	623	667	479
Average Time	38.82	45.64	48.05	36.64	39.23	28.17

Fig. 6 Decryption time of the proposed model and other models based on file size

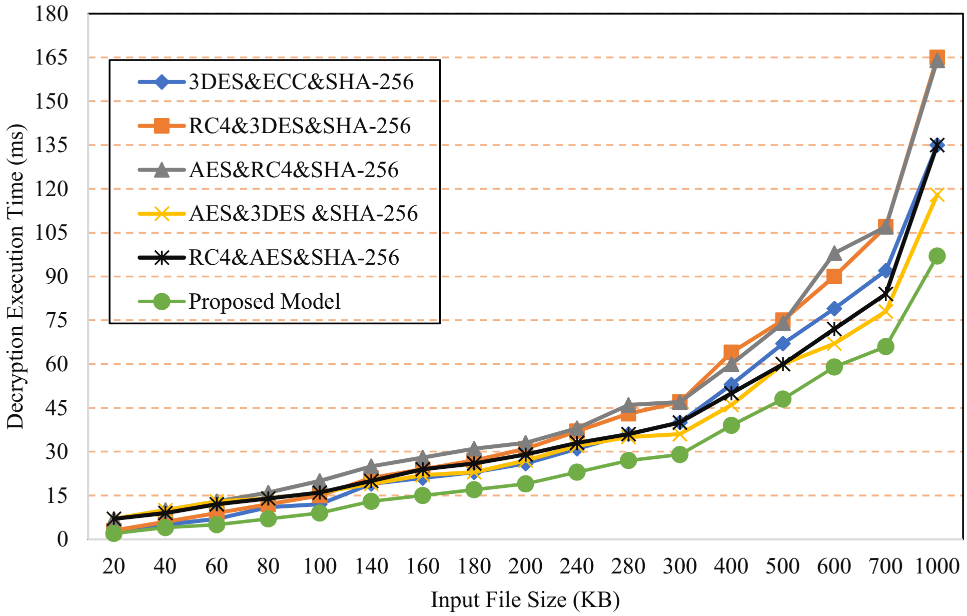


Fig. 7 A comparison of average encryption/decryption time of the proposed model and other models

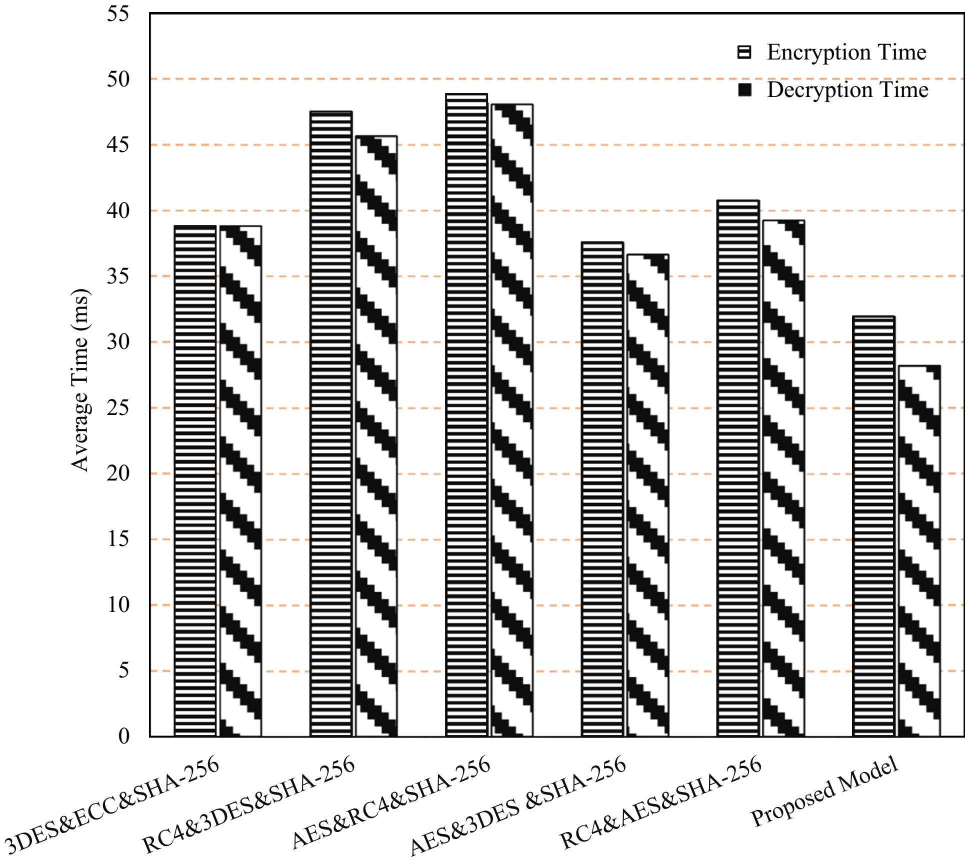


Table 5 Encryption throughput of the proposed model and other models

	Encryption Throughput (KB/ms)					
Input file size (KB)	3DES & ECC&SHA-256	RC4 & 3DES & SHA-256	AES & RC4 & SHA-256	AES & 3DES & SHA-256	RC4 & AES & SHA-256	Proposed model
20	6.66	6.66	2.85	2.85	2.85	10
40	8	5.71	4	4.44	4.44	10
60	8.57	6.66	4.61	4.61	5	10
80	7.27	6.66	5	5.71	5.71	10
100	7.14	6.66	5.26	6.66	5.88	10
140	7.36	6.36	5.60	6.66	6.36	9.33
160	7.27	6.15	5.71	6.95	6.66	9.41
180	7.20	6	5.8	6.92	6.66	9.47
200	7.40	6.25	5.71	7.14	6.66	9.52
240	7.50	6.15	5.85	7.50	6.85	9.23
280	7.56	6.08	6.08	8	7	8.75
300	7.14	6.12	6	7.89	7.14	9.09
400	7.40	6.15	6.15	8.33	7.54	9.30
500	7.81	6.32	6.41	8.47	7.69	8.92
600	7.50	6.31	6.38	8.57	7.69	9.09
700	7.95	6.30	6.48	8.53	7.86	9.21
1000	7.69	5.95	6.06	8.40	7.75	9.17

Table 6 Decryption throughput of the proposed model and other models

	Decryption throughput (KB/ms)					
Input file size (KB)	3DES & ECC & SHA-256	RC4 & 3DES & SHA-256	AES & RC4 & SHA-256	AES & 3DES & SHA-256	RC4 & AES & SHA-256	Proposed model
20	6.66	6.66	2.85	2.85	2.85	10
40	8	6.66	4	4	4.44	10
60	8.57	6.66	4.61	4.61	5	12
80	7.27	6.66	5	5.71	5.71	11.42
100	8.33	6.66	5	6.25	6.25	11.11
140	7.36	6.66	5.6	7.36	7	10.76
160	7.61	6.66	5.71	7.27	6.66	10.66
180	7.82	6.66	5.80	7.82	6.92	10.58
200	7.69	6.45	6.06	7.4	6.89	10.52
240	7.74	6.48	6.31	7.5	7.27	10.43
280	7.77	6.51	6.08	8	7.77	10.37
300	7.5	6.38	6.38	8.33	7.5	10.34
400	7.54	6.25	6.66	8.69	8	10.25
500	7.46	6.66	6.75	8.33	8.33	10.41
600	7.59	6.66	6.12	8.95	8.33	10.16
700	7.60	6.54	6.54	8.97	8.33	10.60
1000	7.40	6.06	6.09	8.47	7.40	10.30

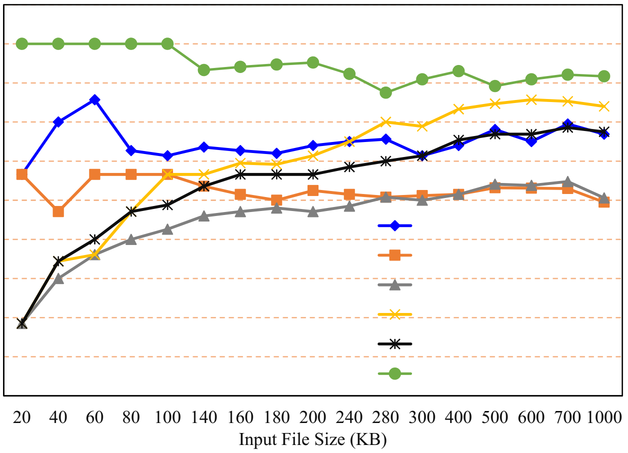


Fig. 8 Encryption throughput of the proposed model and other models based on file size

$$\text{Encryption Throughput}(KB/\text{ms}) = \frac{\sum(\text{Input file})}{\sum(\text{Encryption time})} \tag{4}$$

$$\text{Decryption Throughput}\left(\frac{KB}{\text{ms}}\right) = \frac{\sum(\text{Input file})}{\sum(\text{Decryption})} \tag{5}$$

Figure 8 shows encryption throughput of the proposed model which has a superior performance than other models.

Figure 9 shows decryption throughput of the proposed model which has a better performance than other models.

Figure 10 compares encryption/decryption throughput of the proposed model and other models based on file size.

4.4 Secrecy of cipher

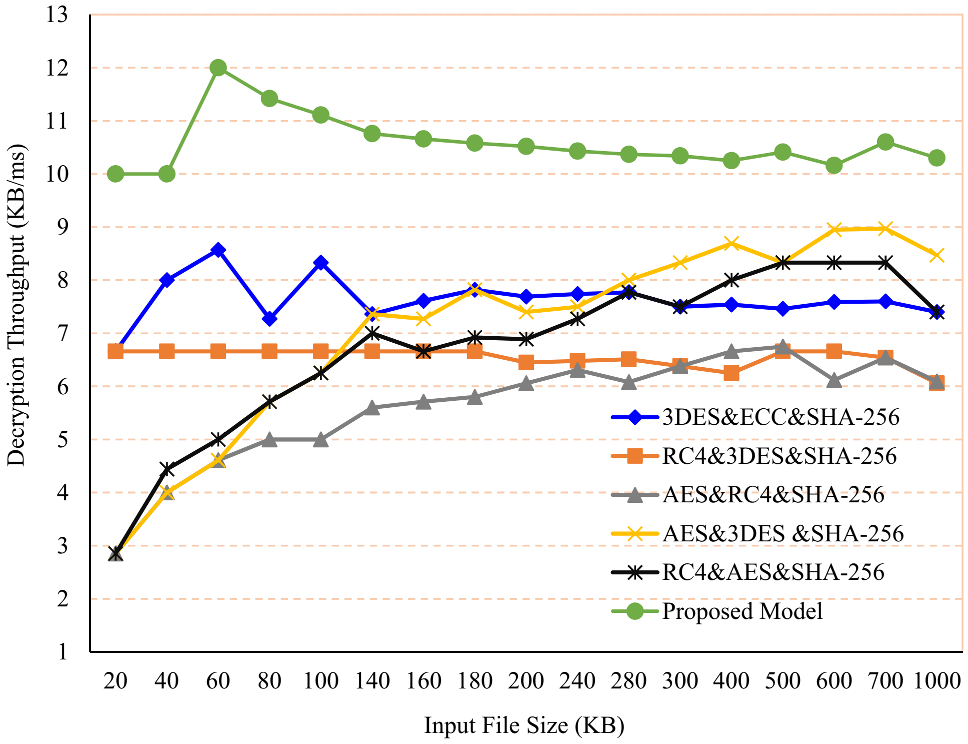
This section deals with the security analysis of the proposed model and other algorithms. The principle of privacy is calculated using Shannon's law (Weerasinghe 2013). The purpose of this metric is to verify the confidentiality of data. Secrecy is one of the most essential metrics in confidentiality. Figure 11 shows the average secrecy value acquired by proposed model, 3DES & ECC & SHA-256, RC4 & 3DES & SHA-256, AES & RC4 & SHA-256, AES & 3DES & SHA-256, and RC4 & AES & SHA-256 are about 1.1715, 0.9828, 0.8443, 1.097, 0.9988, 0.8455 respectively for 1000 KB. It can be seen from Fig. 11 that the proposed scheme has obvious advantages over other schemes in terms of confidentiality, and the secrecy value.

4.5 Security properties

In this section, the security analysis on the basis of six parameters has been done in order to compare the proposed scheme with the other models. Table 7 shows the detailed comparison between the proposed model and other models for securing data communication in IoT.

A replay attack involves retransmitting previously intercepted packets. A replay attack occurs when the attacker has information such as keys as well as previous messages.

Fig. 9 Decryption throughput of the proposed model and other models based on file size



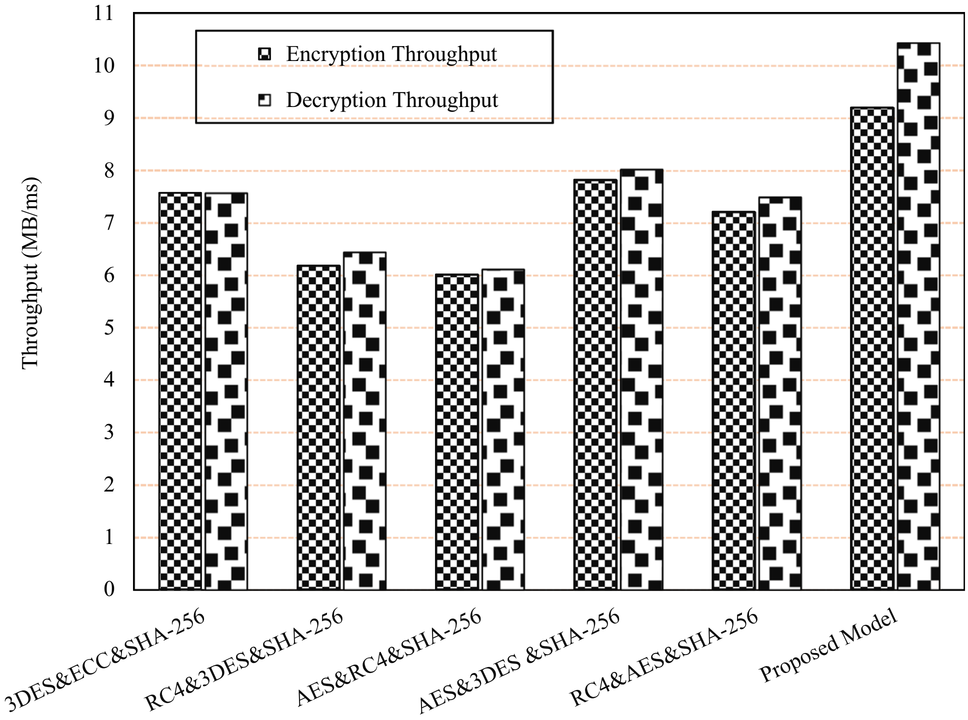


Fig. 10 A comparison of encryption/decryption throughput of the proposed model and other models based on file size

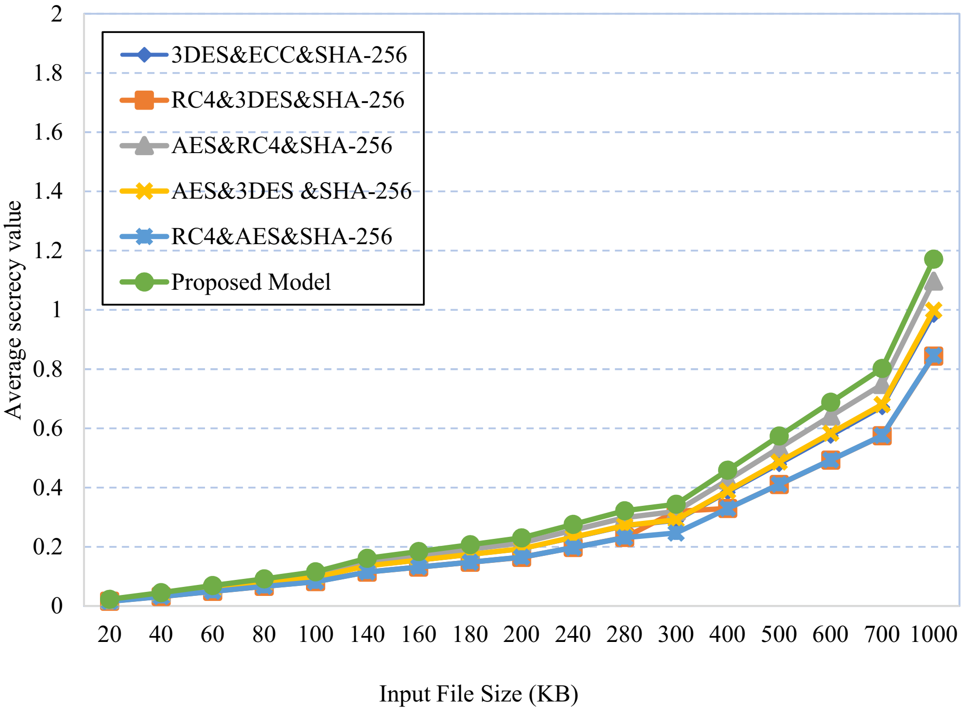


Fig. 11 Secrecy of Ciphers Vs Data Size

When the connection is disconnected, the attacker uses this information to connect the system and to introduce himself as one of the trusted users. In the proposed scheme, due to the encryption of RC4 scheme key by the ECC algorithm, the attacker cannot access the original key and original message.

Man-in-the-Middle (MiM) attack where the attacker interrupts the communication between user and server of IoT and redirects or may modify the exchange messages without knowledge of them. The ECC algorithm prevents the MiM attack. SA is sending kA, the attacker generates $\hat{k}_A$ and sent $\hat{k}_A$ to SB and SB generates kB and sent kB to SA but attacker

Table 7 Comparison of the proposed model based on various security properties

No.	Security property	Replay attack	MiM attack	Session key security	Mutual authentication	Secrecy	Integrity
1	3DES&ECC&SHA-256	–	✓	✓	-	✓	–
2	RC4&3DES&SHA-256	✓	✓	✓	✓	✓	–
3	AES&RC4 &SHA-256	✓	✓	✓	✓	✓	–
4	AES&3DES &SHA-256	✓	✓	✓	✓	✓	–
5	RC4&AES &SHA-256	✓	✓	✓	✓	✓	–
6	Proposed Model	–	–	–	–	–	–

Table 8 A comparison of the proposed model with other models

Time	Models	Input file size (KB)			
		1000	20,000	50,000	100,000
Encryption Execution Time (Sec)	TEA & ECC (Ragab et al. 2019b)	0.14	2.12	6.31	10.42
	XTEA & ECC (Ragab et al. 2019b)	0.15	2.23	6.46	10.81
	XXTEA & ECC (Ragab et al. 2019b)	0.26	3.23	7.74	15.57
	TEA & RSA (Ragab et al. 2019a, b)	0.34	3.69	6.9	12.12
	XTEA & RSA (Ragab et al. 2019a, b)	0.41	4.88	8.04	12.36
	XXTEA & RSA (Ragab et al. 2019a, b)	3.31	6.1	10.62	17.16
	3DES&ECC&SHA-256	0.13	2.7	6.85	13.45
	RC4 & 3DES & SHA-256	0.16	3.31	9.47	17.17
	AES & RC4 & SHA-256	0.16	3.11	7.87	16.1
	AES & 3DES & SHA-256	0.12	2.33	5.78	11.96
	RC4 & AES & SHA-256	0.13	2.54	6.57	13.02
	Proposed Model	0.11	2.26	5.63	10.98
Decryption Execution Time (Sec)	TEA & ECC (Ragab et al. 2019b)	0.12	1.53	3.62	6.01
	XTEA & ECC (Ragab et al. 2019b)	0.13	1.61	3.71	6.96
	XXTEA & ECC (Ragab et al. 2019b)	0.15	2.07	5.01	10.3
	TEA & RSA (Ragab et al. 2019a, b)	0.46	1.72	3.83	6.29
	XTEA & RSA (Ragab et al. 2019a, b)	0.51	3.21	5.55	7.39
	XXTEA & RSA (Ragab et al. 2019a, b)	1.53	4.58	6.6	10.54
	3DES & ECC & SHA-256	0.12	2.6	6.82	13.32
	RC4 & 3DES & SHA-256	0.15	3.32	8.1	17.54
	AES & RC4 & SHA-256	0.14	3	7.92	15.83
	AES & 3DES & SHA-256	0.11	2.29	5.93	12.15
	RC4 & AES & SHA-256	0.12	2.49	6.66	13.42
	Proposed model	0.09	1.07	5.21	10.82

intercepts that message and sends $\hat{k}_A$ to SA. Now, SA compute KAB, SB compute KBA. The MiM attack is detected based on following rules:

$$SA - kA.P \rightarrow \text{Attacker} \rightarrow SA = \hat{k}_A.P$$

$$SB = kB.P$$

$$KAB = \hat{k}A.SB$$

$$KBA = kB.SA$$

$$KAB \neq KBA$$

Because the attacker has no awareness of the random number, the session key cannot be directly calculated, as it is protected by a high entropy ECC point. Thus, the proposed model commitment the session key security.

Data integrity guarantees that the data to be sent has not been changed or modified during transmission. Integrity

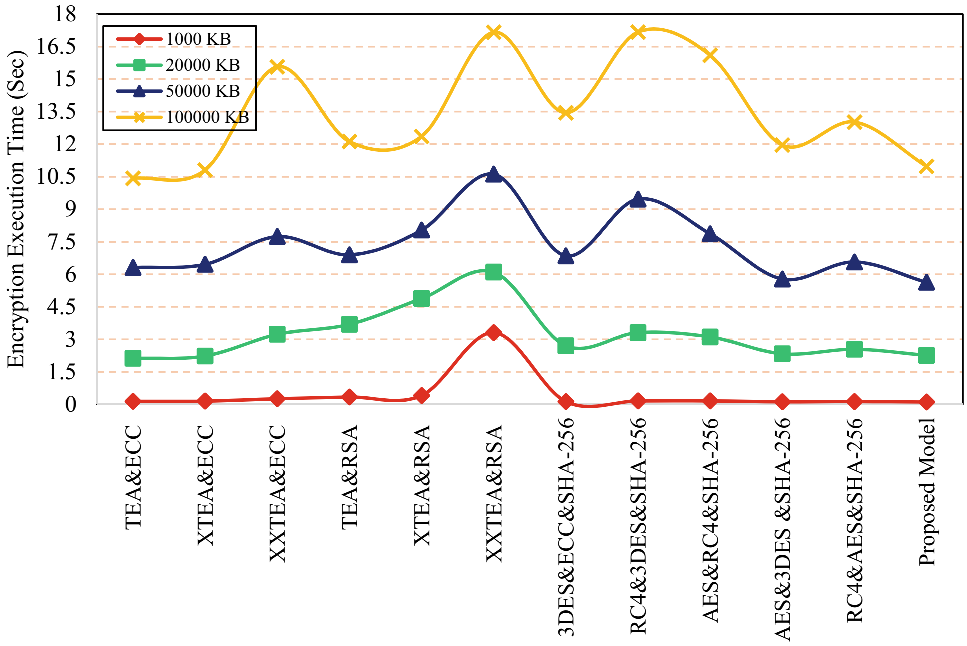


Fig. 12 A comparison of encryption time of the proposed model and other models based on file size

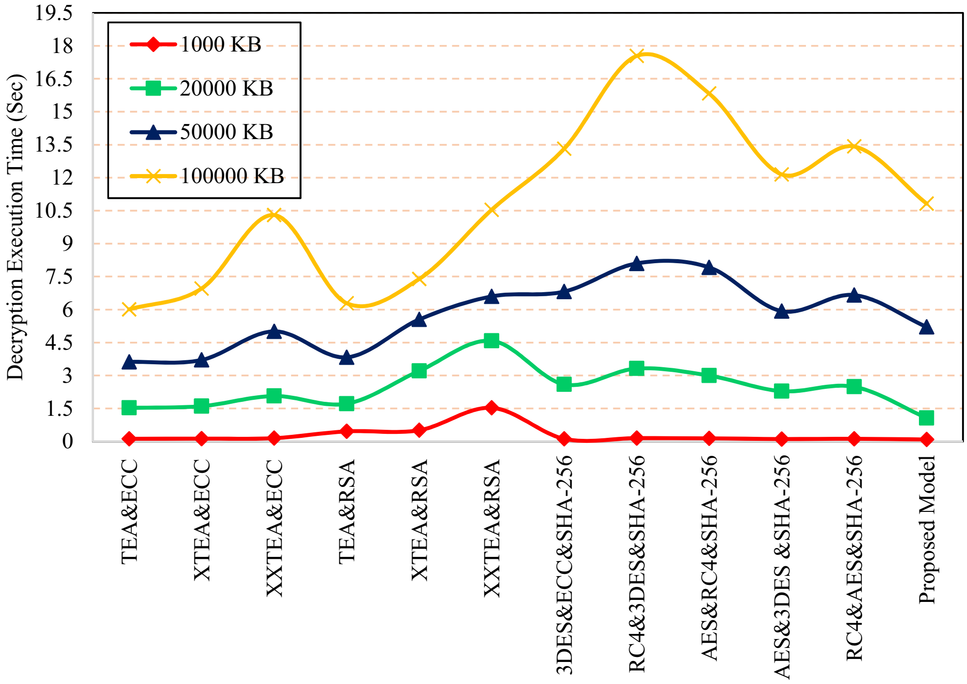


Fig. 13 A comparison of decryption time of the proposed model and other models based on file size

includes maintaining the consistency, accuracy, and validity of the data.

4.6 Comparison and analysis

Table 8 compares the proposed model with other models. Encryption and decryption times are shorter than ECC (Ragab et al. 2019b) and RSA (Ragab et al. 2019a, b) algorithms. File sizes of 1 MB, 20 MB, 50 MB, and 100 MB, along with a 128-bits key length, were used. The key length in TEA&ECC (Ragab et al. 2019b) was 32-bits. Results show that the encryption time of a 100 MB file in TEA&ECC is 10.42 s and the proposed model is 10.98 s with 128-bits key length. Encryption time of the proposed model was shorter than XXTEA&ECC, XTEA&ECC which are developed from TEA&ECC. Decryption times of the same file for TEA&ECC and the proposed model were 6.01 s and 10.82 s. The decryption time of the proposed model is shorter than XXTEA&ECC, XTEA&ECC.

Table 9 A comparison of encryption/decryption throughput of the proposed model and other models

Models	Input File Size (KB)				Avg	File Size (MB)
	1000	20,000	50,000	100,000		
Encryption						
XXTEA & ECC (Ragab et al. 2019b)	3846.15	6191.95	6459.95	6422.61	5730.17	5.7
XXTEA & RSA (Ragab et al. 2019a, b)	302.11	3278.68	4708.09	5827.51	3529.1	3.5
3DES & ECC & SHA-256	7692.3	7407.4	7299.27	7434.95	7458.48	7.4
RC4 & 3DES & SHA-256	6250	6042.29	5279.83	5824.11	5849.05	5.8
AES & RC4 & SHA-256	6250	6430.86	6353.24	6211.19	6311.32	6.3
AES & 3DES & SHA-256	8333.33	8583.69	8517.88	8361.2	8449.02	8.4
RC4 & AES & SHA-256	7692.3	7874.01	7610.35	7680.5	7714.3	7.7
Proposed model	9090.9	8849.55	8880.99	9107.47	8982.22	8.9
Decryption						
XXTEA & ECC (Ragab et al. 2019b)	6666.67	9661.84	9980.04	9708.74	9004.32	9
XXTEA & RSA (Ragab et al. 2019a, b)	653.6	4366.81	7575.76	9487.67	5520.96	5.5
3DES & ECC & SHA-256	8333.34	7692.3	7331.38	7507.51	7716.14	7.7
RC4 & 3DES & SHA-256	6666.67	6024.09	6172.83	5701.25	6141.22	6.1
AES & RC4& SHA-256	7142.86	6666.67	6313.14	6317.11	6609.95	6.6
AES & 3DES & SHA-256	9090.9	8733.63	8431.71	8230.45	8621.68	8.6
RC4 & AES & SHA-256	8333.34	8032.12	7507.51	7451.57	7831.13	7.8
Proposed model	11,111.12	18,691.59	9596.93	9242.15	12,160.44	12.1
Average throughput	The average throughput of encryption and decryption					
	XXTEA & ECC (Ragab et al. 2019b)					5.7 + 9 = 15
	XXTEA & RSA (Ragab et al. 2019a, b)					3.5 + 5.5 = 9
	3DES&ECC&SHA-256					7.4 + 7.7 = 15
	RC4&3DES&SHA-256					5.8 + 6.1 = 12
	AES&RC4&SHA-256					6.3 + 6.6 = 13
	AES&3DES&SHA-256					8.4 + 8.6 = 17
	RC4&AES&SHA-256					7.7 + 7.8 = 16
	Proposed Model					8.9 + 12.1 = **21**
Throughput Efficiency	$\text{Efficiency} = \left[(\text{Proposed model} - \text{Other models})/\ \text{Proposed model}\right] \times 100$					
	XXTEA & ECC (Ragab et al. 2019b)					28%
	XXTEA & RSA (Ragab et al. 2019a, b)					57%
	3DES & ECC & SHA-256					28%
	RC4 & 3DES & SHA-256					42%
	AES & RC4 & SHA-256					38%
	AES & 3DES & SHA-256					19%
	RC4 & AES & SHA-256					23%

Table 8 shows that a 100 Mb file is encrypted in TEA&RSA model in 12.12 s, which is done in 10.98 s in the proposed model. The same file is decrypted in 6.29 s and 10.82 s, respectively. Encryption and decryption times of the proposed model are shorter than XXTEA&ECC (Ragab et al. 2019a, b), XTEA&ECC (Ragab et al. 2019a, b) models. Figures 12 and 13 compare encryption/decryption time of the proposed model and other models (Ragab et al. 2019a, b) based on file size.

Table 9 compares encryption/decryption throughput of the proposed model with other models. It is revealed that efficiency of the proposed model compared to XXTEA&ECC (Ragab et al. 2019b), XXTEA&RSA (Ragab et al. 2019a, b), XXTEA&RSA (Ragab et al. 2019a, b), 3DES & ECC & SHA-256, RC4 & 3DES & SHA-256, AES & RC4 & SHA-256, AES & 3DES & SHA-256, and RC4 & AES & SHA-256 is 28%, 57% and 50%, 28%, 42%, 38%, 19%, 23% respectively. Hence, the proposed scheme has obvious advantages over other algorithms in terms of encryption/decryption throughput, and has excellent encryption efficiency.

5 Conclusion and future works

In this paper we focus, on the security of IoT based irrigation system using RC4, ECC, and SHA-256 algorithms. Firstly, we use The ECC algorithm for improving the security of RC4 scheme by encrypting the key of this scheme. Then, SHA-256 algorithm is used to hashing the encrypted data. We then proved the security of the proposed scheme, as well as demonstrating the utility of the scheme in comparison to other related works in the literature. Extensive simulations validate the effectiveness of the proposed scheme on performance, encryption/decryption time, throughput, and security. Future works will focus on evaluating and refining the proposed scheme to make it applicable for real irrigation system.

References

Abinaya E, Aishwarva K, Lordwin CPM, Kamatchi G, Malarvizhi I (2018) A performance aware security framework to avoid software attacks on Internet of Things (IoT) based patient monitoring system. Int Conf Curr Trends Towards Converg Technol (ICCTCT). https://doi.org/10.1109/ICCTCT.2018.8550955

Agale RR, Gaikwad DP (2017) Automated irrigation and crop security system in agriculture using Internet of Things. Int Conf Comput Commun Control Autom (ICCUBEA). https://doi.org/10.1109/ICCUBEA.2017.8463726

Alaba FA, Othman M, Hashem IAT, Alotaibi F (2017) Internet of Things security: A survey. J Netw Comput Appl 88:10–28

Azari L, Ghaffari A (2015) Proposing a novel method based on network-coding for optimizing error recovery in wireless sensor networks. Indian J Sci Technol 8:859–867

Babayiğit B, Büyükpatpat B (2019) Design and implementation of IoT-based irrigation system. Int Conf Comput Sci Eng (UBMK). https://doi.org/10.1109/UBMK.2019.8907066

Belguith S, Kaaniche N, Laurent M, Jemai A, Attia R (2018) PHO-ABE: Securely outsourcing multiauthority attribute based encryption with policy hidden for cloud assisted IoT. Comput Netw 133:141–156

Burton L, Dave N, Fernandez R, Jayachandran K, Bhansali S (2018) Smart gardening IoT soil sheets for real-time nutrient analysis. J Electrochem Soc 165:B3157

Chandu Y, Kumar KSR, Prabhukhanolkar NV, Anish AN, Rawal S (2017) Design and implementation of hybrid encryption for security of IOT data. Int Conf On Smart Technol Smart Nation (SmartTechCon). https://doi.org/10.1109/SmartTechCon.2017.8358562

Chen J, Tian Z, Cui X, Yin L, Wang X (2019) Trust architecture and reputation evaluation for internet of things. Ambient Intell Humaniz Comput 10:3099–3107. https://doi.org/10.1007/s12652-018-0887-z

Chhabra A, Arora S (2017) An elliptic curve cryptography based encryption scheme for securing the cloud against eavesdropping attacks. Int Conf Collab Internet Comput (CIC). https://doi.org/10.1109/CIC.2017.00040

Ghaffari A (2014) Designing a wireless sensor network for ocean status notification system. Indian J Sci Technol 7:809

Ghaffari A, Rahmani A (2008) Fault tolerant model for data dissemination in wireless sensor networks. In: 2008 International Symposium on Information Technology, IEEE, pp 1–8.

Ghaffari A, Takanloo VA (2011) QoS-based routing protocol with load balancing for wireless multimedia sensor networks using genetic algorithm. World Appl Sci J 15:1659–1666

Gilbert H, Handschuh H (2004) Security analysis of SHA-256 and Sisters. In: Matsui M, Zuccherato RJ (eds) Selected areas in cryptography, Berlin, Heidelberg. Springer, Berlin, pp 175–193

Gulati A, Thakur S (2018) Smart irrigation using Internet of Things. Int Conf Cloud Comput Data Sci Eng (Conflu). https://doi.org/10.1109/CONFLUENCE.2018.8442928

Hamidi H (2019) An approach to develop the smart health using Internet of Things and authentication based on biometric technology. Future Gener Comput Syst 91:434–449. https://doi.org/10.1016/j.future.2018.09.024

Han Q, Zhang Y, Li H (2018) Efficient and robust attribute-based encryption supporting access policy hiding in Internet of Things. Future Gener Comput Syst 83:269–277. https://doi.org/10.1016/j.future.2018.01.019

Hendrawan INR, Yulyantari LP, Pradiptha GA, Starriawan PB (2019) Fuzzy based internet of things irrigation system. Int Conf Cybern Intell System (ICORIS). https://doi.org/10.1109/ICORIS.2019.8874900

Hu Z (2011) A method for the signature of things. In: International Conference on Intelligence Science and Information Engineering, pp. 366–369.

Hussain I, Negi MC, Pandey N (2017) A secure IoT-based power plant control using RSA and DES encryption techniques in data link layer. Int Conf Infocom Technol Unmanned Syst. https://doi.org/10.1109/ICTUS.2017.8286054

Jan MA, Khan F, Alam M, Usman M (2019) A payload-based mutual authentication scheme for Internet of Things. Future Gener Comput Syst 92:1028–1039. https://doi.org/10.1016/j.future.2017.08.035

Jazebi SJ, Ghaffari A (2020) RISA: routing scheme for Internet of Things using shuffled frog leaping optimization algorithm. Ambient Intell Human Comput. https://doi.org/10.1007/s12652-020-01708-6

Jisha S, Philip M (2016) Rfid based security platform for internet of things in health care environment. Online Int Conf Green Eng Technol. https://doi.org/10.1109/GET.2016.7916693

KeyKhosravi D, Ghaffari A, Hosseinalipour A, Khasragi BA (2010) New clustering protocol to decrease probability failure nodes and increasing the lifetime in WSNs. Int J Adv Comp Techn 2:117–121

Khabiri M, Ghaffari A (2018) Energy-aware clustering-based routing in wireless sensor networks using cuckoo optimization algorithm. Wirel Pers Commun 98:2473–2495

Khader M, Alian M, Hraiz R, Almajali S (2017) Simplified AES algorithm for healthcare applications on Internet of Thing. Int Conf Inform Technol (ICIT). https://doi.org/10.1109/ICITECH.2017.8080056

Kothmayr T, Schmitt C, Hu W, Brünig M, Carle G (2012) A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication. Ann IEEE Conf on Local Comput Netw-Workshops. https://doi.org/10.1109/LCNW.2012.6424088

Lee J, Oh S, Jang JW (2015) A work in progress: context based encryption scheme for internet of things. Proced Comput Sci 56:271–275. https://doi.org/10.1016/j.procs.2015.07.208

Li M, Sun Y, Lu H, Maharjan S, Tian Z (2019) Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems. IEEE Internet Things J. https://doi.org/10.1109/JIOT.2019.2962914

Liu Z, Huang X, Hu Z, Khan MK, Seo H, Zhou L (2016) On emerging family of elliptic curves to secure internet of things: ECC comes of age. IEEE Trans Dep Sec Comput 14:237–248

Mao J, Zhu H, Liu Y, Liu Y, Qian W, Zhang J, Huang X (2018) RSA-based handshake protocol in internet of things. Int Conf

Inform Technol Med Educ (ITME). https://doi.org/10.1109/ITME.2018.00220

Matsemela G, Rimer S, Ouahada K, Ndjiongue R, Mngomezulu Z (2017) Internet of things data integrity. IST-Afr Week Conf (IST-Afr). https://doi.org/10.23919/ISTAFRICA.2017.8102332

Mektoubi A, Hassani HL, Belhadaoui H, Rifi M, Zakari A (2016) New approach for securing communication over MQTT protocol A comparaison between RSA and Elliptic Curve. Third Int Conf Syst Collab (SysCo). https://doi.org/10.1109/SYSCO.2016.7831326

Miller VS (1986) Use of elliptic curves in cryptography. In: Williams HC (ed) Advances in Cryptology CRYPTO '85 proceedings, Berlin, Heidelberg. Springer, Berlin, pp 417–426

Mohammadi R, Ghaffari A (2015) Optimizing reliability through network coding in wireless multimedia sensor networks Indian. J Sci Technol 8:834

Odelu V, Das AK, Khan MK, Choo KR, Jo M (2017) Expressive CP-ABE scheme for mobile devices in IoT satisfying constant-size keys and ciphertexts. IEEE Access 5:3273–3283. https://doi.org/10.1109/ACCESS.2017.2669940

Pant VK, Prakash J, Asthana A (2015) Three step data security model for cloud computing based on RSA and steganography. Int Conf Green Comput Internet Things (ICGCIoT). https://doi.org/10.1109/ICGCIoT.2015.7380514

Qiu J, Du L, Zhang D, Su S, Tian Z (2020a) Nei-TTE: Intelligent Traffic Time Estimation Based on Fine-Grained Time Derivation of Road Segments for Smart City. IEEE Trans Ind Inform 16:2659–2666. https://doi.org/10.1109/TII.2019.2943906

Qiu J, Tian Z, Du C, Zuo Q, Su S, Fang B (2020b) A survey on access control in the age of internet of things. IEEE Inter of Things J. https://doi.org/10.1109/JIOT.2020.2969326

Ragab A, Selim G, Wahdan A, Madani A (2019a) Robust hybrid lightweight cryptosystem for protecting IoT smart devices International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, SpaCCS 2019: Security, Privacy, and Anonymity in Computation, Communication, and Storage, LNCS 11637:5–19.

Ragab AAM, Madani A, Wahdan AM, Selim GMI (2019b) Hybrid cryptosystems for protecting IoT smart devices with comparative analysis and evaluation. In: Arai K, Bhatia R, Kapoor S (eds) Proceedings of the Future Technologies Conference (FTC). Springer International Publishing, Cham, pp 862–876

Saha R, Geetha G, Kumar G, Kim T-H, Buchanan WJ (2019) MRC4: a modified rc4 algorithm using symmetric random function generator for improved cryptographic features. IEEE Access 7:172045–172054

Sharma G, Kalra S (2018) A lightweight multi-factor secure smart card based remote user authentication scheme for cloud-IoT applications. J Netw Comput Appl 42:95–106. https://doi.org/10.1016/j.jisa.2018.08.003

Singh S, Sharma PK, Moon SY, Park JH (2017) Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. Ambient Intell Human Comput. https://doi.org/10.1007/s12652-017-0494-4

Stinson DR (1995) Cryptography: theory and (2005th ed). CRC Press, Boca Raton

Suárez-Albela M, Fernández-Caramés TM, Fraga-Lamas P, Castedo L (2018) A practical performance comparison of ecc and rsa for resource-constrained IoT devices. Glob Internet Things Summit (GIoTS). https://doi.org/10.1109/GIOTS.2018.8534575

Tian Z et al (2019) Real-time lateral movement detection based on evidence reasoning network for edge computing environment. IEEE Trans on Indust Inform 15:4285–4294. https://doi.org/10.1109/TII.2019.2907754

Tian Z, Gao X, Su S, Qiu J (2020) Vcash: a novel reputation framework for identifying denial of traffic service in internet of connected vehicles. IEEE Internet Things J 7:3901–3909. https://doi.org/10.1109/JIOT.2019.2951620

Tian Z, Luo C, Qiu J, Du X, Guizani M (2020) A distributed deep learning system for web attack detection on edge devices. IEEE Trans Ind Inform 16:1963–1971. https://doi.org/10.1109/TII.2019.2938778

Touati L, Challal Y (2016) Collaborative KP-ABE for cloud-based Internet of Things applications. IEEE Int Conf Commun (ICC). https://doi.org/10.1109/ICC.2016.7510836

Weerasinghe TDB (2013) An effective RC4 stream cipher. IEEE Eight Int Conf Ind Inform Syst. https://doi.org/10.1109/ICIInfS.2013.6731957

Wei P, Zhou Z (2018) Research on security of information sharing in Internet of Things based on key algorithm. Future Gener Comput Syst 88:599–605. https://doi.org/10.1016/j.future.2018.04.035

Wu L, Chen B, Choo K-KR, He D (2018) Efficient and secure searchable encryption protocol for cloud-based Internet of Things. J Parall Distrib Comput 111:152–161. https://doi.org/10.1016/j.jpdc.2017.08.007

Xu L, Li J, Chen X, Li W, Tang S, Wu H-T (2019) Tc-PEDCKS: Towards time controlled public key encryption with delegatable conjunctive keyword search for Internet of Things. J Netw Comput Appl 128:11–20. https://doi.org/10.1016/j.jnca.2018.12.003

Xu S, Yang G, Mu Y, Liu X (2019) A secure IoT cloud storage system with fine-grained access control and decryption key exposure resistance. Future Gener Comput Syst 97:284–294. https://doi.org/10.1016/j.future.2019.02.051

Yang Y, Zheng X, Tang C (2017) Lightweight distributed secure data management system for health internet of things. J Netw Comput Appl 89:26–37. https://doi.org/10.1016/j.jnca.2016.11.017

Yao X, Chen Z, Tian Y (2015) A lightweight attribute-based encryption scheme for the Internet of Things. Future Gener Comput Syst 49:104–112. https://doi.org/10.1016/j.future.2014.10.010

Yoshida H, Biryukov A (2006) Analysis of a SHA-256 variant. In: Preneel B, Tavares S (eds) Selected areas in cryptography, Berlin, Heidelberg, 2006. Springer, Berlin, pp 245–260

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.