Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 14(4): 233-239, Dec. 2016 Regular paper

Secure Message Transmission against Remote Control System

Taehwan Park, Hwajeong Seo, Bongjin Bae, and Howon Kim*, *Member*, *KIICE*

Department of Computer Engineering, Pusan National University, Busan 46241, Korea

Abstract

A remote control system (RCS) can monitor a user's confidential information by using the broadcast receivers in Android OS. However, the current RCS detection methods are based only on a virus vaccine. Therefore, if the user's smartphone is infected by a brand new RCS, these methods cannot detect this new RCS immediately. In this paper, we present a secure message transmission medium. This medium is completely isolated from networks and can communicate securely through a QR code channel by using symmetric key cryptography such as the AES block cipher and public key cryptography such as elliptic curve cryptography for providing security. Therefore, the RCS cannot detect any confidential information. This approach is completely immune to any RCS attacks. Furthermore, we present a secure QR code-based key exchange protocol by using the elliptic curve Diffie–Hellman method and message transmission protocols; the proposed protocol has high usability and is very secure.

Index Terms: Network isolation, QR code, Remote control system, Smartphone

I. INTRODUCTION

These days, smartphones have very powerful network connections and a high computing power. However, recently, malicious attacks based on a remote control system (RCS) have shown that a user's confidential information can be easily exposed to the attacker. Once malware codes are installed in the user's smartphone, they try to get root authority and access the confidential information coming through the event receiver. For example, in 2015, the National Intelligence Agency (NIS) bought some RCS code from an Italian hacking team for gathering someone's confidential information. Thereafter, a number of virus vaccine companies developed a vaccine to detect whether a user's smartphone is infected by a malware code such as the RCS. However, these vaccines can detect malware by using the existing malware's signature or pattern; therefore, if the user's smartphone is infected by a new RCS code, they cannot detect this new code. Hence, we need to develop a high-security technology for secure message transactions. Here, we suggest some secure message transmission methods through a special secure medium. This medium is not connected to the Internet but communicates with a smartphone only through QR code channels and can support security by using a symmetric key block cipher and public key cryptography. If the user communicates with a friend by using the special secure medium, the RCS code in the user's smartphone cannot detect any confidential information. Therefore, the proposed method ensures complete security against any potential and possible attacks. The remainder of this paper is organized as follows: In Section II, we introduce the related works. In Section III, we present some

Received 08 August 2016, **Revised** 10 August 2016, **Accepted** 29 August 2016
***Corresponding Author** Howon Kim (E-mail: howonkim@pusan.ac.kr, Tel: +82-51-510-3927)
Department of Computer Engineering, Pusan National University, 2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan 46241, Korea.

Open Access http://doi.org/10.6109/jicce.2016.14.4.233 print ISSN: 2234-8255 online ISSN: 2234-8883

secure message transaction technologies against RCS attacks. In Section IV, we provide insights on how to present numerical results or applications that illustrate the results. In Section V, we conclude the paper.

II. RELATED WORKS

A. Remote Control System

In this section, we describe RCS working procedures for eavesdropping the information of SMS, SNS, and telephonic conversations (see Fig. 1). Adversaries install RCS malware in the user's smartphones by using a phishing URL or other hacking methods. Once the RCS malware is installed on the user's smartphone, this malware tries to transfer the user's SMS and SNS contents and telephonic conversations to the attacker's server through the Android broadcast receiver and network.

In the RCS malware, a number of authorities to access events and functions are taken over by attackers [1]. In particular, the broadcast receiver recognizes events such as phone calls and SMS send/receive. These features can start activities on Android OS explicitly (class assign method) or implicitly (request action on data). The various intents are exploited, and the detailed intents are described in Table 1. This RCS malware can operate a specific event by using the Android OnReceive function and an intent filter when the specific intent occurs. The architecture of the RCS source code enables actions per specific intent as defined in the OnReceive function and uses it. In this study, we have analyzed the RCS working procedures for an SMS attack.

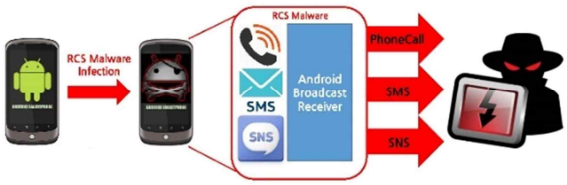

Fig. 1. Remote Control System (RCS) malware operating procedures.

Table 1. Required intents for RCS

Type	Function	Authority
intent.action.BOOT_COMPLETED	OS booting	-
action.USER_PRESENT	User unlock	-
provider.Telephony.SMS_RECEIVED	SMS receive function	√
intent.action.NEW_OUTGOING_CALL	User phone call	√
intent.action.PHONE_STATE	Phone call status	-
intent.category.DEFAULT	Using activity receive	-

When the intent related to an SMS occurs, the OnReceive function in the RCS source code is executed. If the attacker wants to know the SMS information, he can block the next broadcast intent by using the abortBroadcast() function and execute a thread to handle the SMS data. The executed thread register can be a new receiver when it does not exist for checking the SMS send/receive status. It can save and handle phone numbers and SMS messages. MsgHandler and MsgObserver collect the SMS or MMS data and then send these data to the attacker. The packet format is required for sending data. The moduleMessage.notification function converts the data into the packet format by using an atomic function and sends them to the attacker. In the case of a telephonic conversation, it saves the recording data in a voice file. Then, the processed data can be delivered to the attacker's server.

B. Network Isolation Technologies

Network isolation technologies are efficient counter-measures against advanced persistent threat (APT) attacks. Table 2. describes various types of network isolation technologies. The network isolation technologies can be constructed in two different ways, namely logical and physical network isolations. Physical network isolation involves separating PCs; one PC supports network functions, but the other does not support the network function in order to protect information. This method has an advantage that hackers cannot get any information. However, the cost to set up a physical network isolation environment is high. On the other hand, logical network isolation can be categorized into the virtual desktop infrastructure (VDI), client PC, and OS kernel methods. The logical VDI method supports a server based on a virtual desktop, and its advantage is to block network threats. However, it needs to maintain this server. The logical client PC method involves a software-based virtual desktop, and its advantage is the low maintenance cost; however, it has a reliability issue. There are several advanced network isolations available. Zhang et al. [2] have described an efficient physical network isolation system for protecting information. Skovoroda and Gamayunov [3] describe various mobile malware detection and prevention methods such as static/dynamic analysis, authority analysis,

Table 2. Comparisons of network isolation technologies

Classification	Description	Advantage	Disadvantage
Physical	Separate PCs	Physical separation	High costs
Logical VDI	Server-based VD	Block network threats	Server maintenance
Logical client PC	SW-based VD	Low costs	Reliability issues

and machine learning-based methods. In particular, the paper describes an authority analysis such as the Kirin security service and the WHYPER method. Feizollah et al. [4] proposed a mobile malware detection method using features selection for four types of features (namely static, dynamic, hybrid, and application metadata). Authority, Java code, intent filters, and strings are categorized as static features, and system calls, network traffic, and user interactions as dynamic features. Application metadata features include creator ID and application descriptions. Cyber security companies such as AhnLab and Estsoft distribute RCS detection vaccines. These RCS detection vaccines can detect an RCS in real time in a smartphone environment. However, these vaccines detect RCS malware by using the signature hashing method on the RCS malware; therefore, if the RCS malware is modified, it is difficult to detect the modified RCS malware.

C. Quick Response (QR) Code

The quick response (QR) code is a two-dimensional square lattice information code, which stores a maximum of 7,089 characters as digits and 4,296 characters in the ASCII format. The QR code can be categorized into L, M, Q, and H levels according to the type of error correction function. The QR code has 140 versions. Version 1 is composed of 21 × 21 cells, and the length and the width increase by four cells with every new version until version 40. Version 40 is composed of 177 × 177 cells. A cell is a black/white square dot in the QR code [5]. The QR code can be used in the authentication process. Divya and Muthukumarasamy [6] proposed an authentication method against a key logging attack and can prevent key logging by visualization authentication using the QR code. Murkute et al. [7] proposed a method to eliminate the risk of phishing or user identity checks by using a QR code-based one-time pad (OTP). Kale et al. [8] proposed an anti-phishing single sign on (SSO) authentication model by using the QR code.

III. PROPOSED METHOD

In this section, we present an efficient countermeasure against the RCS through physical network isolation and a QR code-based network channel. The proposed method requires an additional secret message reader for the user. The proposed architecture is illustrated in Fig. 2. The proposed architecture consists of three parts (interface & input/output, crypto module, and algorithm). The interface & input/output part has a touch screen to receive touch signals and a camera to capture the QR code. The crypto module has a symmetric key encryption algorithm (AES-256), a cryptographic hash function (SHA-256), and an asymmetric key encryption algorithm (elliptic curve cryptography, ECC) based on the NIST secp256r1 curve. ECC in the crypto module is used for the elliptic curve Diffie–Hellman (ECDH) key exchange algorithm. The algorithm part has some algorithms for handling the QR code and the keypad.

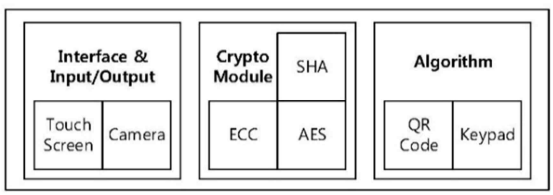

Fig. 2. Proposed architecture for secure communication.

Table 3. Comparison between devices

Requirement	Smartphone	Secure reader
Camera	√	√
Screen	√	√
OR code algorithm	-	√
Cipher (AES and ECC)	-	√
Wireless network (Wi-Fi)	√	-
Data storage	√	-
One-chip design	-	√

In Table 3, we compare the features of a smartphone and a secret message reader. The smartphone needs to support wireless networks, screens, and cameras for checking the QR code. However, the smartphone does not support secure QR code encoding/decoding algorithms and ciphers because the secure QR code information is handled only by the secret message reader. On the other hand, the secret message reader does not support wireless networks such as Wi-Fi or Bluetooth, because the secret message reader is for preventing information leakage through any remote network communication. It also does not have data storage for additional source code or command execution for preventing the execution of a third party's program or command in the secret message reader. The secret message reader only supports the functions that encode/decode a QR code image (set in a specific format). Its internal memory can also be designed to be compact and on one chip for

Fig. 3. Encryption message transmission process based on network isolation.

preventing any malware attempts. The working system consists of an ordinary smartphone and a secret message reader, as shown in Fig. 3.

A. Secure Message Transmission

In this section, we describe the proposed secure message transmission method. This method supports the key agreement and message encryption functions.

Key agreement: The ECDH key exchange protocol is executed in the proposed method. The key exchange scenario is as follows: Alice and Bob exchange the public key by using the opponent's QR codes (public key pairs: *aP* and *bP*) and generate *abP* by using the ECDH algorithm. Therefore, it assumes that users (A and B) have a secret message reader that supports ECDH on the basis of the NIST secp256r1 curve and exchange the public key. Algorithm 1 shows the secure key agreement with the QR code. It requires public key *aP* and private key *b*. If user B has his private key *b*, then it generates his public key *bP* and the QR code of his public key. Next, if user B receives user A's public key *aP* and the key value is not null, then the user can generate the shared key *abP* according to the ECDH algorithm on the basis of user A's public key *aP* and user B's public key *bP*. The output is the shared session key *abP*.

Algorithm 1 Secure key agreement with QR code

Require: Public Key aP, Private Key b
Ensure: Session Key abP
1: **if** b != NULL **then**
2: bP← GenECCKey(b)
3: GenSendQRcode(bP)
4: **if** aP != NULL **then**
5: abP← ECDH(aP, b)
6: **return** Session Key abP

Message encryption: The secret message reader can encrypt the user's input message and decrypt the result of decoding the received opponent QR code. Algorithm 2 describes the generation of the QR code of the encrypted message. It requires the input message *m*. First, if the input message *m* is not null, then it encrypts the message by using a symmetric key cipher and generates the QR code (*q*) of the cipher (*c*). If the message is null, then it returns FAIL. The next step is to send the QR code. If the QR code (*q*) is not null, then it scans this code and sends it (*c*) to the other party and returns SUCCESS. Algorithm 3 describes that the receiver decrypts and parses the QR code of the encrypted message. This requires the received QR code (*q*). If received the QR code (*q*) is not null, then it parses, decrypts this code by using the symmetric key cipher, and sets the message *m* as the decryption result. If the received QR code (*q*) is null, then it sets the message *m* as NULL.

Algorithm 2 Generate and send QR code of encrypted message

Require: Input Message m
Ensure: Result of generate and send QR code
1: **if** m != NULL **then**
2: c← Enc_k(m)
3: q← GenQRcode(c)
4: **else**
5: q←NULL
6: **return** FAIL
7: **if** q != NULL **then**
8: p← ScanQRcode(q)
9: Send(p)
10: **return** SUCCESS

Algorithm 3 Decrypting QR code of encrypted message

Require: QR code q
Ensure: Decrypted Message m
1: **if** q != NULL **then**
2: c← QRcodeScanParse(q)
3: m← Dec_k(c)
4: **else**
5: m← NULL
6: **return** Message m

IV. EVALUATION

In this section, we describe the experimental environment, procedures, security analysis, and performance. There are two scenarios tested in this experiment. The first is the secure message transmission between users (A and B).

The procedure for the first scenario is as follows: user A inputs a message and encrypts the message by using the block cipher. Then, the QR code of the encrypted message is created and presented on the screen of the secret message reader. User A scans the QR code of the encrypted message and sends it to user B through a communication network (3G, LTE, Wi-Fi, etc.). Then, user B receives the QR code of the encrypted message from user A and scans this code by using his own secret message reader. After scanning, user B can read the message by decoding the scanned QR code and decrypting it. The second experiment scenario is the key agreement between users (A and B). This scenario is based on the Diffie−Hellman key exchange protocol. First, user A generates his private key (*a*) and public key (*aP*). Then, he generates the QR code of his public key (*aP*) and delivers the QR code message to user B. Second, user B scans user A's QR code and generates the session key (*abP*). User A

can also generate the session key (*abP*) by getting the QR code of user B's public key.

A. Experimental Setup

The operating system of the smartphones and the secret message readers used is Android 5.0.1 Lollipop. The development environment is Android Studio 2.1.1. The proposed method is developed by using the Java language. For conducting this experiment, we need to set up two smartphones and two secret message readers. In the case of the secret message readers, we turn off all of communication functions. We use the javax.crypto and java.security packages for the AES-256 cipher and used the Spongy Castle ECC library [9] for ECC secp256r1 and ECDH [10]. For the QR code generation, we use the zxing library [11]. For the sake of convenience, we omit smartphone communications.

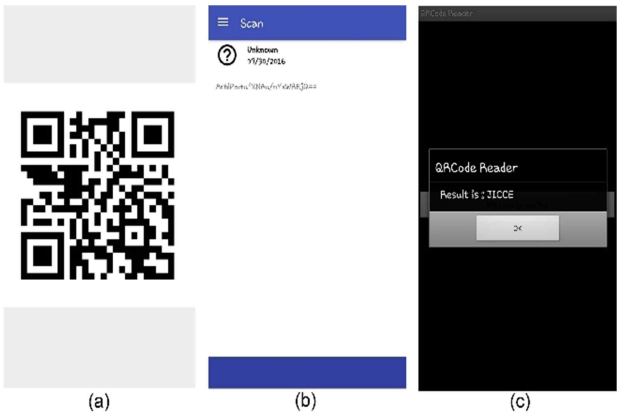

Fig. 4. Secure encrypted message transmission: (a) QR code of the encrypted message, (b) decoding result of the QR code, and (c) decoding and decrypting result of the QR code.

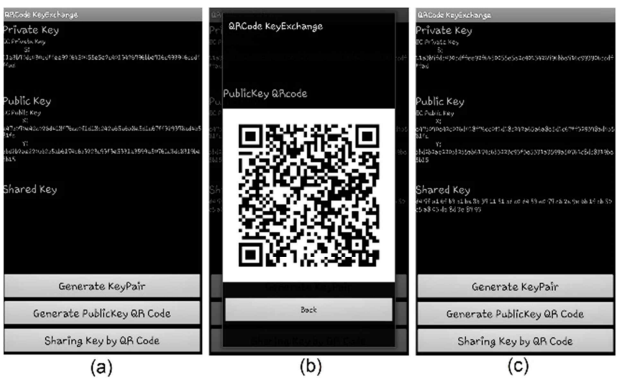

Fig. 5. Secure key exchange: (a) user's private key and public key, (b) QR code of the user's public key, and (c) after secure key exchange and generation of the shared key.

Secure message transmission

Fig. 4. describes secure message transmission procedures by capturing picture at each step.

The secure message transmission procedures are such as follow.

1. Input a message and generate the QR code: User A inputs a message on his secret message reader, and then, this reader encrypts the input message by using the AES-256 encryption.
2. Generate the QR code: User A's secret message reader generates the QR code of the encrypted message.
3. Scan the QR code: User A scans the QR code of the encrypted message by using the smartphone camera.
4. Transmit the QR code: User A transmits the scanned QR code to user B through a wireless network.
5. User B scans the QR code: User B receives the QR code from user A through the wireless network and scans it by using his secret message reader.
6. User B decrypts the QR code and checks the message: User B decrypts the scanned QR code and checks the message by using his secret message reader.

Secure key agreement: The main screen of users A and B's secret message reader has three buttons (generate private/public key pair, generate public key QR code, and sharing key by QR code) and three information windows (private key, public key, and shared key information).

Fig. 5. describes secure key exchange procedures by capturing picture at each step.

The secure key exchange procedures are such as follow.

1. User A generates the key pair: User A can generate his private key and public key pair by clicking on the Generate KeyPair button.
2. User A generates the public key QR code: User A can generate the QR code of his public key by clicking on the Generate Public Key QR Code button.
3. User B generates a key pair: User B can generate his private key and public key pair by clicking on the Generate KeyPair button.
4. User B scans user A's public key QR code: User B can scan user A's public key QR code by using his own secret message reader.
5. User B generates the shared key QR code: After scanning user A's public key (*aP*) QR code, user B can generate the shared key (*abP*) by multiplying his private key (*b*) and user A's public key (*aP*).
6. User A generates the shared key QR code: User A can generate the shared key (*abP*) by scanning user B's public key (*bP*) and multiplying his private key (*a*) and user B's public key (*bP*).

B. Performance and Security Analysis

In this section, we analyze the performance and the security of the proposed method. Here, we compare the existing methods with the proposed method.

Security analysis: The main goal of a security analysis is to verify the security against a malware attack such as smartphone root authority acquisition. The existing secure communication application programs such as Telegram support TLS or SSL network security on the network. However, if the malware seizes root authority, the decrypted data on user's screen can be revealed to the attacker. This can offset the advantage of network security communication; therefore, the proposed method has an advantage in that it separates the smartphone and the secret message reader physically for protecting information leakage from the root authority of the smartphone. If the user's smartphone is infected by a malware code such as RCS, the proposed method can protect the confidential information on the basis of network isolation and a secure QR code-based network (QR code includes the encrypted information).

Skovoroda and Gamayunov [3] suggested mobile malware detection based on a static/dynamic analysis, and Feizollah et al. [4] proposed mobile malware detection by using features selection. However, these methods cannot detect modified malware, if the malware is modified to conceal its features or signature. On the other hand, the proposed method can provide secure message transactions and prevent a secret outflow while the existing methods cannot detect or prevent a malware infection. The proposed method is an efficient countermeasure for RCS attacks that the existing method cannot counter, as described in Table 4.

Performance analysis: The existing security communication on a smartphone has a security problem in that an attacker can acquire the root authority of the smartphone. The method proposed in this paper can solve this problem by applying network isolation and adding a secret message reader for generating and reading the QR code in an efficient manner. The costs for the secret message reader are as follows: first, in the case of message transmission, the user needs to input a message and generate the corresponding QR code. Next, in the case of message transmission, it operates with two more operations (QR code generation and reading). In the case of receiving the message, it operates with two more operations (QR code generation and decryption). QR code generation and decryption are executed by the processor. Reading the QR code may take less than 3 seconds; thus, the proposed method has an advantage in that it does not need an additional operation as compared to the existing method. Further, message encryption and decryption takes less than 1 seconds; therefore, users do not feel uncomfortable using the proposed method.

Table 4. Comparison between existing methods and the proposed method

	Secure transaction	RCS prevention
Skovoroda and Gamayunov [3]	-	√
Feizollah et al. [4]	-	√
Proposed method	√	√

V. CONCLUSION

In this paper, we proposed a secure and efficient countermeasure method against malware such as RCS. The existing methods against malware cannot protect information because they focus on detecting malware rather than prevention of information loss. Furthermore, the methods need to be updated when the malware is modified or new malware appears. On the other hand, the proposed method is secure under all circumstances and uses a network isolation technique and a QR code-based secure network channel. The proposed method uses a symmetric key cipher for message encryption and decryption. The ECDH algorithm [10] is used for key agreement. Reading the QR code takes less than 3 s and message encryption and decryption are carried out within a second. Therefore, by using the proposed method, users can protect their confidential information. Lastly, the proposed method is also comfortable to use.

ACKNOWLEDGMENTS

This research was supported by a grant from the Advanced Technology Center R&D Program funded by Ministry of Trade, Industry & Energy of Korea (No. 10048537).

REFERENCES

[1] Citizen Lab, RCS agent for Android [Internet]. Available: https://github.com/hackedteam/core-android.

[2] D. G. Zhang, Y. Wu, W. B. Zhang, D. H. Zhang, and S. Q. Zhang, "The design of a physical network isolation system," *Applied Mechanics and Materials*, vol. 687, pp. 2192-2195, 2014.

[3] A. Skovoroda and D. Gamayunov, "Securing mobile devices: malware mitigation methods," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications* (*JoWUA)*, vol. 6, no. 2, pp. 78-97, 2015.

[4] A. Feizollah, N. B. Anuar, R. Salleh, and A. W. A. Wahab, "A review on feature selection in mobile malware detection," *Digital Investigation*, vol. 13, pp. 22-37, 2015.
[5] QR Code.com, Types of QR code and information/version of QR code [Internet]. Available: http://www.qrcode.com/ko/codes/.
[6] R. Divya and S. Muthukumarasamy, "Visual authentication using QR code to prevent keylogging," *International Journal of Engineering Trends and Technology*, vol. 20, no. 3, pp. 149-154, 2015.
[7] J. Murkute, H. Nagpure, H. Kute, N. Mohadikar, and C. Devade, "Online banking authentication system using QR-code and mobile OTP," *International Journal of Engineering Research and Applications*, vol. 3, no. 2, pp. 1810-1815, 2013.
[8] V. Kale, Y. Nakat, S. Bhosale, A. Bandal, and R. G. Patole, "A mobile based authentication scheme using QR code for bank security," *International Journal of Advance Research in Computer Science and Management Studies*, vol. 3, no. 2, pp. 192-196, 2015.
[9] Spongy Castle [Internet]. Available: https://rtyley.github.io/spongycastle/.
[10] ECC Reference, "SEC 1: elliptic curve cryptography," 2009 [Internet]. Available: http://www.secg.org/sec1-v2.pdf.
[11] ZXing code [Internet]. Available: https://github.com/zxing/zxing.

Taehwan Park
received his B.S.E.E. from Pusan National University, Pusan, Republic of Korea, in 2013. He is currently pursuing a combined M.S. and Ph.D. course in Computer Engineering at Pusan National University. His research interests include IoT device security, information security, elliptic curve cryptography, and post quantum cryptography.

Hwajeong Seo
received his B.S.E.E. from Pusan National University, Pusan, Republic of Korea, in 2010. He also received his M.S. and Ph.D. in Computer Engineering from the same university. His research interests include sensor networks, information security, elliptic curve cryptography, and RFID security.

Bongjin Bae
received his B.S.E.E. from Pusan National University, Pusan, Republic of Korea, in 2015. Currently, he is pursuing a master's degree in Pusan National University. His research interests include IoT, information security, and post quantum cryptography.

Howon Kim
received his B.S.E.E. from Kyungpook National University, Daegu, Republic of Korea, in 1993, and his M.S. and Ph.D. in Electronic and Electrical Engineering from Pohang University of Science and Technology (POSTECH), Pohang, Republic of Korea, in 1995 and 1999, respectively. From July 2003 to June 2004, he studied with the COSY group at the Ruhr-University of Bochum, Germany. He was a senior member of the technical staff at the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Republic of Korea. He is currently working as an associate professor with the Department of Computer Engineering, School of Computer Science and Engineering, Pusan National University, Busan, Republic of Korea. His research interests include sensor networks, information security, and computer architecture. Currently, his main research focus is on the Internet of Things (IoT) technology, public key cryptosystems, post quantum cryptography, and the related security issues.