IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 31, 2019, accepted November 15, 2019, date of publication November 19, 2019, date of current version December 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2954413

Prototype Device With Lightweight Protocol for Secure RFID Communication Without Reliable Connectivity

AHMED ALAMER[1,2], BEN SOH[1], (Senior Member, IEEE), AHMED H. ALAHMADI[3], AND DAVID E. BRUMBAUGH[4]

[1]Department of Computer Science and Information Technology, School of Engineering and Mathematical Sciences, La Trobe University, Melbourne, VIC 3086, Australia
[2]Department of Mathematics, College of Science, Tabuk University, Tabuk 7149, Saudi Arabia
[3]Department of Computer Science and Information, Taibah University, Medina 41477, Saudi Arabia
[4]Techno Authority, Digital Consultant, Mobile, AL 36609, USA

Corresponding author: Ahmed Alamer (a.alamer@latrobe.edu.au)

The work of A. ALAMER was supported by the Ph.D. Scholarship provided by Tabuk University.

ABSTRACT The increased use of radio frequency identification (RFID) technology has made it difficult to maintain secure operations in RFID environments. In addition, establishing a secure system when the internet is unavailable and finding a secure method to share keys are challenges that must be addressed. Considering the limitations of RFID tagging in terms of space, power, and storage, there is a need for practical low-power hardware microcontrollers with lightweight encryption methods suitable for implementation. Our secure system, which is based on the use of hardware-embedded RFID tags, is a novel approach that employs four initialization vectors (IVs) and key pairs to develop solutions for the secure storage, distribution, and alteration of the IVs and keys for use with the MICKEY 2.0 stream cipher. We propose the use of a low-power RFID-compatible device to provide a secure solution for exchanging and storing IVs and key pairs in the absence of an internet or wireless connection. We call this device the near-field secure data extractor (NFSDE). In addition, we demonstrate its operation in a practical eHealth scenario. Software emulation of the device is used to test the related processes and evaluate their efficiency and security. The use of this simple RFID-compatible prototype device with a lightweight encryption system, which provides public-key-like security but is not internet-dependent, alleviates healthcare security issues and encourages the development of similar tools that can be adapted for use in other fields that require sensitive data to be securely handled.

INDEX TERMS eHealth, Internet connection, lightweight cryptosystem, MICKEY 2.0, RFID.

I. INTRODUCTION

Radio frequency identification (RFID) tags are frequently used in sensor networks for identification and security, and they play an important role in the Internet of Things (IoT). The initial use of the IoT showed that low-power devices exhibited inadequate security [1]. In several cases, poor or nonexistent IoT encryption led to the devices being seriously compromised via the internet connection [2].

RFID tags allow data transfer in a contactless manner, as seen in the verification of goods in supermarkets [3]. This makes RFID tags an easy-to-use and cost-effective verification tool for applications such as access keys, tracking (for example, the tracking of animals, items in markets, and shipped goods), and communicating information such as details of the contents of a container and instructions to receivers. In addition, RFID technology is a low-cost solution [4], and because RFID tags use timely data that can be checked directly, using them prevents human error because manual intervention is reduced. Owing to these advantages, RFID tags can improve the quality of healthcare, especially considering the ageing population and the incidents of medical errors by health practitioners. They can provide solutions to ensure the privacy and safety of patient information and store critical medical records on ID cards. However, owing to the ease of use and low cost of RFID tags, implementers have tended to overlook the need for data privacy and security [5]. Researchers have begun to propose stream ciphers as an attractive solution to this problem [6]. Current solutions employing stream ciphers

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed Farouk.

generally lack confidentiality, integrity, and authentication provided by asymmetric algorithms of a public key infrastructure (PKI) [7]. An ongoing challenge associated with cryptography is secure key exchange [8], [9]. Although asymmetric algorithms can address this problem, they require more computing power than is typically available in low-power devices. Zhang *et al.* [10] preferred a hardware solution to this problem and proposed custom-designed hardware called Recryptor. Conti *et al*. [11] used a custom chip called *Fulmine* for near-sensor IoT applications. In the current study, we achieved the same goal by using a fundamentally different approach. Our solution involves the use of software and off-the-shelf hardware, making the proposed device more flexible and less expensive and allowing the use of standardized hardware.

Stream ciphers use symmetric algorithms that require a novel key-exchange solution to maintain confidentiality, integrity, and authentication.

MICKEY 2.0, Trivium, and Grain (refer to [12] for design details and source codes) are among the most lightweight stream ciphers used in digital and portable devices, and they have found many uses in hardware and various IoT systems [13]. Although MICKEY 2.0 was designed for hardware implementation, it is also suitable for use in software. Its performance has been tested and compared with that of Trivium and Grain to determine its effectiveness (refer to [14] for further details). Banik [15] demonstrated that MICKEY 2.0 was resistant to attacks because it uses an irregular mix (clocking) to update its internal components, making it difficult to locate a random input in either the R or S register (R and S are linear and nonlinear registers, respectively). In addition, the complexity of its interior design ensures that the cipher shows more resistance to differential fault attacks than do Trivium and Grain ciphers [16]. Users may choose another lightweight encryption method, MICKEY 2.0, as an example for the reasons mentioned before. One recent encryption method for tiny devices with limited computation power was investigated [17], using software implementations to evaluate 8 lightweight hash functions with built-in block ciphers. They used a passive CRFID (computational radio-frequency identification) for implementation and, based on their experiments, recommend using the MD5 hash function.

This study aimed to implement a lightweight synchronous encryption algorithm, which is suitable for RFID technology, by considering the need for optimized cost, power, and computation necessary when using RFID tags without internet connectivity while gaining and maintaining the advantages typically found in a PKI. To achieve such a secure system, we designed a prototype device called the near-field secure data extractor (NFSDE), implemented a MICKEY 2.0 cipher, and conducted a secure key/IV exchange. In addition, we implemented a software emulation of the proposed system with a secure encryption protocol. The security of this device relies on a trusted record keeper (R) and a secure flash drive (SD).

As an example, we present a scenario in which important medical information (such as allergies and existing conditions) must be carried by a patient. In this scenario, neither the internet nor wireless communication is available or reliable, for example, in cases in which the patient is located at a remote site or in a disaster zone. Although this medical scenario is only one example of the application of our proposed device, it was chosen for our study because its security requirements are the same as those of a more powerful system. The concept can be applied and adapted for almost every security-critical application that requires access to secure systems in situations in which wireless access is unreliable. Because the absence of internet connectivity is assumed, this novel technology can also be used for out-of-band authentication.

A. MAIN CONTRIBUTIONS

1. This study introduces a secure RFID-based sensitive data-protection system called NFSDE. This device can be considered a prototype that can either be used directly or modified as required by users. It can also be used as a guide to create dedicated hardware with features consistent with those of our prototype device.
2. We propose a secure eHealth system (proof of concept) by implementing the MICKEY 2.0 cipher. The system is readily adaptable for applications other than those in the healthcare field. It provides a framework instead of using a PKI to enable the use of different lightweight stream ciphers.
3. Our system is designed to use off-the-shelf hardware encryption and optimized lightweight stream ciphers to enhance RFID security and provide similar advantages as public key exchange.
4. This study led to the development of reliable and secure key and initialization vector (IV) exchange (for multiple keys) as well as key-management and key-update solutions for a secure RFID reader that is neither dependent on the internet nor on wireless communication.
5. The processing time for using NFSDE to decrypt 4K of RFID data using Mickey 2.0 is a matter of milliseconds.

B. ORGANIZATION OF THE PAPER

In Section II, we present related work to highlight our contribution to the field of lightweight security. Section III describes the application of our system to the specifically chosen field of eHealth. In Section IV, we discuss the practical applications of lightweight cryptosystems. Section V provides our reasoning and functional details on the process and system operation. Section VI presents the specifications of the NFSDE prototype device. Section VII presents the processes of tag creation, MICKEY 2.0 implementation, key generation, and Key/IV creation and exchange. In addition, the process for key alterations is described. We also describe how our system fits into the framework of modern security principles and finally discuss the software emulation and testing of the system. Section VIII portrays the extent to which the proposed system is resistant to anticipated attacks and the mitigation of known weaknesses. Section IX provides the overall analysis and discussion of the results. Finally, Section X draws the conclusions and presents the direction for future work.

II. RELATED WORK

One method for providing reasonable security levels for anti-counterfeiting resistance is the use of a lightweight encryption (LWE) system, such as exclusive OR (XOR) manipulation, which is suitable for implementation in small devices [18]. Lee *et al.* [18] demonstrated the hardware implementation of a lightweight authentication protocol and explained how their protocol could enhance RFID security.

To understand the use of an LWE system with RFID and in the IoT in general, readers may refer to a previous review [19], in which the authors provide an overview of lightweight cryptographic solutions. In addition, they discuss the security provided by each solution and demonstrate the software and hardware implementations. Moreover, they explain the advantages of such a cryptosystem over the advanced encryption standard (AES).

A. LWE SYSTEM

The LWE system has become increasingly popular in recent years, as it provides a significant amount of protection and is appropriate for devices with limited computational capacity and available memory, such as contactless cards [20]. Although other cryptographic systems, such as AES, possess higher arithmetic capability, LWE, which allows increased communication between devices, is faster and allows a larger amount of information to be transferred in a shorter time. Therefore, studies have focused on lightweight stream ciphers; for example, the eSTREAM project [12] evaluated the proposals of ciphers based on their suitability for software or hardware implementations. The systems selected for hardware implementation in the final stage of the eSTEAM project were Trivium, Grain, and MICKEY 2.0.

In addition, readers may want to refer to a more extensive overview of lightweight cryptosystems, their applications, and the classification of such systems as lightweight and ultra-lightweight with respect to their properties and requirements [21].

Lightweight stream ciphers have been used for identification purposes. This is achieved by designing a protocol that develops a cipher tool for simultaneous identity verification with a high degree of protection and sufficiency, with a focus on immunity against denial-of-service (DoS) attacks and with its compatibility with RFID tags [22].

On the other hand, lightweight block ciphers, such as LBlock [23], have been used to ensure a lower number of gate equivalents (GEs: 1320). However, these are vulnerable to attacks, as demonstrated by Karakoç *et al.* [24], who carried out an attack on a 23-round LBlock.

Inspired by the data encryption standard (DES) encryption system, DES Lightweight [25] was designed with fewer transistors—25% fewer than in DES and 45% fewer than in AES—to ensure compatibility with RFID tags and was a competitor among the lightweight stream ciphers in the eSTREAM project [26].

In cases where the key for encryption/decryption is reused and stored, the storage needs to be secured; in this regard, one of the proposed techniques is the use of electrically erasable programmable read-only memory (EEPROM). A comparison of this technique with other storage techniques has been published in [27].

Several practical applications require adequate encryption for a low-power device. Babbage and Dodd [28] were the first to introduce the MICKEY 2.0 encryption algorithm. In the present study, we explored several applications for which this would be useful (see Section 3).

As RFID tags are the preferred security system owing to their cost efficiency, they must be registered, verified, and updated to ensure their security [29]. In our system, an RFID tag can be authenticated without an internet connection by using a microcontroller, such as Raspberry Pi, which is connected to an RFID reader and a fingerprint scanner.

B. PHYSICALLY UNCLONABLE FUNCTIONS

An interesting area in the resource-constrained security study is in the field of physically unclonable functions (PUFs). For our current application, using PUF is not optimal because commercially available PUF devices are still sensitive to environmental factors. In extreme conditions, PUFs may provide inaccurate authentication. Since the eHealth scenario is specifically targeted for adverse conditions, we chose not to use PUFs in our design. Current research is showing promise in mitigating this issue [30], [31].

See Appendix C for more information on the potential for PUF in NFSDE.

C. eHEALTH

An important example application for RFID in the field of eHealth care involves monitoring students' performance in universities by linking the RFID with their health status. This includes their health records, medical history, and important health data, such as blood pressure and prescriptions [32].

In cases of unreliable internet connectivity, attackers' threats can be more aggressive [33], as the existing protocol needs to change to address the communication challenges and provide cryptographic methods that are tailored to the situation. Nevertheless, in their study [33], they provide solutions within anonymous and untrusted networks, while our framework offers security when the internet connection is weak or does not exist.

To investigate the existing eHealth technologies in the IoT [34], the authors stress the importance of policies and regulations in order to have secure communication, and they recommended more extensive research in this area.

For more information regarding improvements and challenges in security with the IoT, see [35]–[37].

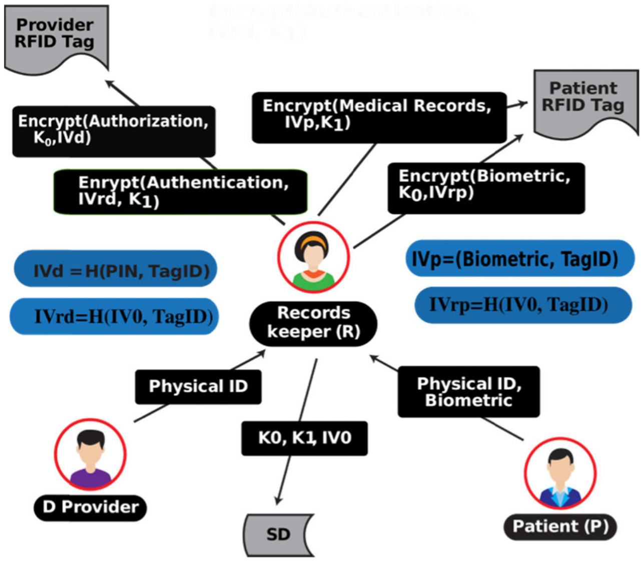

FIGURE 1. Relationship between entities.

III. OVERVIEW OF THE eHEALTH PROCESS

In this scenario (see Fig. 1), the patient (P) is given an RFID tag that can store approximately 4 kB of medical information encrypted with MICKEY 2.0. The information can be saved on a card or wristband worn by the patient or in the form of a key fob. The RFID tag must be cryptographically certified by the entity responsible for storing the medical records.

To create a patient tag, the record keeper (R) scans the patient tag to ascertain its unique ID, scans the patient's fingerprint for biometric validation, and then reads, encrypts, and cryptographically signs the medical data. Then, these data are written onto the patient tag.

The patient carries this tag in one of our previously discussed scenarios to ensure that, in case of a medical emergency, important medical data are readily available to medical care providers.

A provider (D) is any medical professional (doctor, nurse, EMT, paramedic, etc.) who would require rapid access to a patient's medical data (pre-existing conditions, allergies, etc.).

The provider requires an RFID tag that activates the reader and allows decryption of the patient data. The record keeper creates a provider tag by validating the provider's identity and authorization level. The provider supplies a personal identification number (PIN), which is used in the encryption of his or her card. The identity and authorization level are encrypted and then cryptographically signed by the record keeper. The encrypted identity, authorization level, and cryptographic signature are written onto the provider tag.

To access the medical record, an authenticated provider needs only scan the patient's tag and fingerprint, whereupon the device instantly displays the medical record. For authentication, providers need a hardware-encrypted USB drive, their own RFID card, and a PIN. The authentication process is similar to a debit-card-based purchase (extracting the card information by the card issuer for authentication during the payment process); however, it requires less time. This authentication process only needs to take place once per session.

Fig. 1 shows the data provided by each of the parties (record keeper, R; provider, D, patient, P) along with the location of each data item. The record keeper (R) validates the provider (D) and stores the patient's (P) medical records and biometrics on the patient's RFID tag. The provider (D) will unlock the NFSDE with a PIN and his or her RFID tag. When the patient (P) has a matching biometric, the provider (D) will be able to read the patient's medical records with the NFSDE.

A. FRAMEWORK FOR THE MAJOR PROCESSES

In the following section, we describe the major processes in our eHealth system. Table 1 lists the notations used in this paper.

IV. PRACTICAL APPLICATIONS OF LOW-POWER CIPHERS

Due to cost or speed problems, the security features of many commercial RFID systems are disabled. Many researchers have been able to reverse engineer cryptographic algorithms and emulate communication protocols by using low-cost equipment [38], [39].

As stated previously [40], for a long time, the false belief that security is impractical and our privacy is too expensive has created an adverse situation that has essentially impacted the entire world. The theories and practical suggestions presented in this paper show that privacy and security can be quick, inexpensive, and effective.

The eSTREAM ECRYPT project [12] was initiated to address the aforementioned problems. Because security involves more than just algorithms, we chose to define a complete system to demonstrate the security of a low-cost, low-power device.

We propose a standards-based device called NFSDE, which does not require an internet connection and allows sensitive information to be read from and written to an RFID tag in a secure manner.

The proposed system follows the current best practices of confidentiality, integrity, and availability [41] as well as the three A's of data security: authorization, authentication, and accounting. The system is proposed for use in situations in which 1) the internet must be avoided for security reasons or 2) the internet is unavailable (for example, in disaster-prone or remote zones). We present a prototype of a device that 1) consumes low power, 2) is memory constrained, and 3) is slower than a conventional computer. These properties are ideal in emergencies or in situations in which a portable device is desired or power availability is unreliable. Adaptation of these features has caused early adopters to neglect security to the detriment of the online world [42].

Most studies on low-power encryption seem to have focused on the automated tracking of objects, such as shipping containers, vehicles, and robots [3]. In this study, we focus on sensitive personal data; this necessarily adds complexity and criticality because of the involvement of

TABLE 1. Notations.

Notation	Name	Explanation/Notes
R	Medical Record Keeper	This is a role and indicates a person who has privileged access to all data and cryptographic secrets. The person acts as a gatekeeper to sensitive data and authorization. This role is likely to be fulfilled by multiple individuals depending on the administrative organization of the issuing authority.
P	Patient	Is a person in an emergency or a remote situation, where a reliable internet connection is not available. Emergency or rescue workers represent an appropriate use case.
D	Medical Provider, (EMT, paramedic, nurse, etc.)	The provider has a device to read the RFID tag of the patient.
SD	Secure Flash Drive (USB)	It has a "keypad," which is used to independently encrypt/decrypt data stored therein.
K_0	Key to Encrypt/Decrypt Patient Data	K_0 is read from the SD. Together with IV_P, this is used to encrypt or decrypt patient data by using MICKEY.
K_{0e}	Data key stored in the SD before decryption	When the provider enters the passcode for the SD, K_{0e} is decrypted to K_0.
K_1	Key to Encrypt Authentication (From the record keeper)	Authentication from the record keeper is encrypted with MICKEY using K_1, IV_{rp} (for patient), and IV_{rd} (for provider)
K_{1e}	Data key stored in the SD before decryption	When the provider enters the passcode for the SD, K_{1e} is decrypted to K_1.
IV_0	IV "seed" for the record keeper after decryption, which is considered a "Secret"	IV_0 is used as a starting point for generating the initialization vector for encryption of authentication. It is XORd with a unique tag ID to create the final authorization IV. For the provider, it is XORd with the provider's unique tag ID to create IV_{rd}. For the patient, it is XORd with the patient's unique tag ID to create IV_{rp}. These hashes are used because they consume low power/CPU.
IV_{0e}	Encrypted IV_0 (in the SD) for the record keeper	When the provider enters the passcode for the SD, IV0e is decrypted to IV_0.
IV_{rp}	IV for authenticating patient signature	Created by XORing IV_0 with patient's unique tag ID. These hashes are used because they consume low power/CPU.
IV_{rd}	IV for authenticating provider signature	Is created by XORing $IV_0$with provider's unique tag ID. These hashes are used because they consume low power/CPU.
IV_d	IV to encrypt authorization for provider	Is created by hashing the checksum of the provider's PIN with unique ID through concatenation. These hashes are used because they consume low power/CPU.
IV_P	IV to encrypt patient data	Is created by concatenating checksum hash of fingerprint template with unique patient tag ID. These hashes are used because they consume low power/CPU.

additional human factors, including biometric reference capture and "eyes-on" validation of the patient's identity. Our device and processes require that time be spent "up front" to authorize and authenticate the RFID tags. However, the "read time" is very fast, and critical data can be accessed quickly in adverse situations, as demonstrated by our software emulator.

As this is a standards-based device, we hope that it inspires people to consider the addition of security to their own low-power projects or adopt these ideas and modify them beyond the scope of these examples.

We considered several scenarios, many of which are briefly outlined here. We finally settled on the eHealth scenario because it is highly sensitive and requires the most comprehensive security owing to privacy concerns, information criticality, and access speed. The proposed system handles the "worst-case" scenario for confidentiality, integrity, and availability without compromising a low-power environment.

A. SCENARIOS OTHER THAN eHEALTH

In this section, we discuss other possible scenarios for secure RFID applications and provide a relatively brief overview of each.

1) TWO-MAN RULE

In high-security situations, for example, when planning large corporate expenditures, it is often necessary for two employees to authorize an action [43]. This system would allow "offline" multi-factor enforcement of the two-man rule.

2) HUMAN COURIER SCENARIO

As the proposed device does not require access to the internet, it is suitable for securing data that should be isolated from hackers. This is especially appropriate when multiple people need to authorize access to an asset.

Parties requiring *very secure* communication commonly use human couriers rather than, for example, secure cloud-based information exchange. Although the parties involved do not wish to exchange sensitive data via the internet, the cloud can still be used for secure key exchange [44].

Even if reliable internet connectivity is available, this technology can be used for out-of-band authentication. Other potential applications include multi-factor authentication, financial applications, such as "smart wallets," cold storage of cryptocurrency keys, master encryption keys, secure data transport by military organizations, and high-security IT and research departments where it is necessary to control physical access to systems and equipment.

B. OTHER COMMON USES

The most common use of RFID tags is for tracking goods in transit and for compiling inventory. The encrypted information can be stored on the tracking tag. Our system is resistant to eavesdropping and side-channel attacks. In an IoT scenario, no unencrypted data need to be exposed.

RFID tags are also used to track animals [45]. Our proposed system can easily allow researchers or trainers to include sensitive data on a tag attached to an animal.

V. SOLUTION FOR eHEALTH

As shown previously, many practical applications require effective encryption for low-power devices. Frequently, cost is the driving factor; however, another more important consideration involves situations in which internet availability is unreliable or nonexistent. In such a case, critical data would need to be locally available, secure "at rest," and accessible by low-power devices. Encryption is an absolute necessity for personal identifiable information (PII) and sensitive data, such as medical or financial data. Readers interested in the PII application, especially with respect to eHealth, may refer to [46].

Notably, once a sufficiently secure low-power algorithm has been developed, no other major innovation is required for security and privacy.

One of the principles of "privacy by design" is that privacy is not a zero-sum game [40]. In other words, technical limitations cannot be an excuse to compromise on privacy. As stated in Section III, the proposed system implements the three pillars of security: confidentiality, integrity, and availability (also known as the CIA triad). The third pillar, "availability," is especially significant, as it specifically makes critical data available in situations in which it would not be otherwise available. Thus, it would be considered a component of "Security in Depth" [47]. In addition to the CIA triad, our proposed system implements the three A's of data security: authorization, authentication, and accounting" [48]. Our example demonstrates that technical limitations do not necessarily have to be the cause of security limitations.

Security is important even when communication and/or power availability is unreliable, such as in remote areas and during disasters as well as during infrastructure upgrades or technical failures. Low power, encrypted data storage solutions can be critical in such situations.

In this study, as an example of the application of our proposed system, we consider a scenario in which important medical information must be carried by a patient. In this scenario, the internet and wireless communication are unreliable. This could include cases involving remote areas without internet or a disaster situation where communication infrastructure has been damaged. Presumably, in these situations, the battery life of an active device, such as an RFID reader, would be critical, as the ability to recharge would be restricted. In remote hazardous areas, workers or security forces may find themselves in sudden need of medical attention. Immediate access to critical medical information might help save lives.

This system was designed for situations (such as medical emergencies) in which delayed access to sensitive and secure data would be highly undesirable. Therefore, a large amount of time is expended by our processes "up front" to prepare the data and perform authentication and authorization. However, when the data need to be accessed, our processes are designed to be fast and reliable. Authentication and authorization require only a few seconds. This is partly because the keystream generation algorithm (MICKEY 2.0) is relatively fast and partly because we attempted to reduce the number of steps necessary to access the reader without compromising security.

We used MICKEY 2.0 in the demo system. We expect the process to be valid for other lightweight stream ciphers such as Trivium and Grain. Any attack would be unsuccessful because the keys and IVs are protected with physically secure devices (USB).

VI. DEVICE (NFSDE)

In this section, we describe the components of the device required by a medical practitioner to read the medical records securely and quickly.

The selected components can suitably implement the features of our NFSDE system (note that the system can utilize other similar components). Our selections were made on the basis of the performance and cost effectiveness of the prototype and proof of concept. As these are standard-based "off-the-shelf" components, superior (and expensive) components may be readily available for practical use in harsher environments. Fig. 2 shows the components that could be used in a practical prototype.

A. SECURE USB DRIVE

The fundamental problem associated with encryption is the retention and delivery of the encryption key(s).

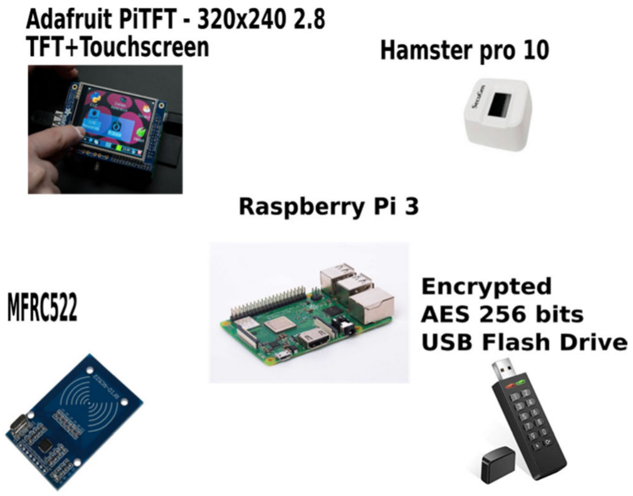

FIGURE 2. Major components of NFSDE.

A typical storage device is fundamentally insecure because gaining access to the device (either directly or by using a side channel) could reveal all the keys. As the scenario assumes the unavailability of an internet connection, a strictly cloud-based key storage solution (such as the AWS Secrets Manager) would also be unacceptable.

The second problem to be addressed is the signing of each tag in a manner that is guaranteed to have been authenticated by a record keeper. This requires a secure and low-power method of signing the tags without exposing the shared authentication value to be created. In more resource-intensive systems, this is accomplished by way of public-key encryption. However, in a resource-constrained environment, an alternative to a public key is desired.

The two aforementioned problems could be solved by the use of a hardware-encrypted USB. A secure drive (SD) can be encrypted and decrypted using a local access key and is inexpensive (a USB-type SD with AES and 256-bit encryption costs approximately US$ 14 [49]).

Note that procedures and policies should be in place to ensure that the storage keys in the SD are separated from the reader when the reader is not in use. In addition, the SD must be regularly re-encrypted at scheduled intervals. The regular rotation of keys and re-encryption of the SD could mitigate side-channel attacks, such as differential scans and algebraic and statistical attacks, as other keys are not affected when one key is compromised.

A practical option for the SD is the device manufactured by STNTUS INNOVATIONS: A USB drive with a storage capacity of 16 GB is more than sufficient for most purposes. The device uses AES 256-bit encryption, and its major advantage is that it performs all encryption and decryption tasks within the device. Therefore, ensuring interactivity with the operating system to guarantee confidentiality is unnecessary. A time-based list for keys is contained in this type of device, making it possible for any new key to be added to this list.

EEPROM: Researchers have argued as to why EEPROM [50] cannot be used in place of a secure USB drive. Presumably, the EEPROM would be ‘‘in the box’’ rather than an external device. If not an EEPROM, perhaps a physically unclonable function (PUF) [30], [31], could be used.

The disadvantage of EEPROM or a PUF is that it eliminates the security provided by a separately encrypted device for key storage. First, EEPROM makes a system vulnerable to side-channel attacks [51]. Second, it eliminates the ease of separating the keys from the device in the field [52]. Presumably, EEPROM or a PUF would need to be installed by a technician, making key updates and rotations slow and inconvenient. Thus, even encrypted EEPROM or a PUF is, at best, an inelegant approximation of the physical USB key.

B. RASPBERRY Pi

Raspberry Pi [53] is an effective, inexpensive, single-board PC (a full computer rather than simply a CPU). We found that Raspberry Pi is considered a lower power, readily available device for resource-constrained environments [54]. Specifically, Raspberry Pi is used in resource-constrained cryptographic and blockchain resource experiments [55].

Although we chose off-the-shelf commodity components, the protocol itself does not require the full power of Raspberry Pi. The protocol requires only simple shifts and XORing and a few ANDs. Raspberry Pi also contains standard drivers for the RFID readers and fingerprint scanner, as well as the display driver. This design evolved on the basis of the total cost of the system we evaluated. We chose to concentrate on the higher cost items (the device, SD, and fingerprint reader) because in a single device fewer of them were required, at most one per provider. RFID tags are typically cheaper per unit than CRFID tags. Since one tag is required per patient, the total cost of implementation was likely to be lower with a large number of patients.

Raspberry Pi has sufficient processing capacity to handle decryption, an RFID reader, and a fingerprint scanner. We recommend ‘‘Adafruit PiTFT (320×240, 2.8’’) as a practical choice for the display because it facilitates touchscreen input [56], [57].

C. HAMSTER PRO 10

This device can create a 500-byte (**ISO/IEC 19794-2 standard**) template, which is sufficient for use. An example of the device is Precise Hamster Pro 10, which is a new portable fingerprint reader with adequate performance for reading a patient’s fingerprint. The major benefit of this product is that it allows the fingerprint design template to be consistently stored within 500 bytes [58].

Hamster Pro 10 can determine whether a fingerprint scan matches the reference template. As biometric data are ‘‘fuzzy,’’ some tuning is typically required. If recognition tuning is ‘‘strict’’, fewer false acceptances but more false rejections would be observed. If recognition tuning is ‘‘relaxed’’, there would be more false acceptances but fewer false rejections. In medical applications, acceptance tuning should be looser than normal.

Additionally, the application can be modified to accept multiple-finger enrolment if the primary finger is injured. Note that each reference template consumes an additional 500 bytes on the RFID card.

Finally, although any biometric could be used, fingerprints were selected because of the matching speed and well-defined standards for fingerprint matching.

D. MFRC522

This is a very inexpensive RFID reader and is an adequate choice for a functional prototype. Its major feature is its ability to perform read and write operations on the MIFARE standard card [59]. A production model may require a more robust (but functionally identical) board.

E. TAGS

We selected the 4K MIFARE [60] contactless smart card for implementation in the RFID tag. In certain scenarios, a wrist tag would be suitable, whereas in others, an ID card would be a more appropriate choice. Other forms of the RFID tag may be used for specific patients: the tag IDs could possibly be incorporated into a necklace or key fob. A distinctive MIFARE tag ID can be specified by a particular provider, and any functionally identical tag can be utilized.

One possibility was to use CRFID tags with PUF features for patient or provider tags [30]. One of our goals was to keep the cost lower at scale. We presume that adding CRFID for each patient would raise the overall cost of implementation for a large number of patients.

VII. TAG CREATION AND IMPLEMENTATION PROCESS

In this section, we provide details of the tag creation process for both provider and patient, together with an explanation of the implementation processes.

A. PROVIDER TAG CREATION

1) USE OF MICKEY 2.0

The MICKEY 2.0 algorithm is used for two distinct purposes within the provider tag. The first use of MICKEY encryption on the provider tag is to encrypt the "provider identity" and the "authorization" fields. The usual presumption is that a medical practitioner requires access to only a portion of the patient's data. When the authorization field is decrypted by the NFSDE reader, the software within the NFSDE device is programmed to display only the data that the provider is authorized to view based on the value of the authorization field.

$IV_{d,}$ which is the IV for the provider, is unique to the provider and the RFID tag and is computed by creating a hash of the provider's selected PIN and unique tag ID. Additionally, the secret key K_0 is also used. K_0 varies over time. MICKEY, IV_d and K_0 are used to encrypt the authorization and identity information before it is stored on the tag.

After the data are encrypted, a 32-bit cyclic redundancy check (CRC) [61] is computed for use in the next phase.

Although 32 bits are not sufficiently strong to create a cryptographically secure hash, it could be used to resist a collision attack [62]. We address this limitation by encrypting the hash with MICKEY.

The process of decryption with MICKEY 2.0, K_0, and IV_d follows that of activating the reader.

The second purpose of the MICKEY 2.0 algorithm is authentication. One of the pillars of data security is "integrity." It is important to know that the data are obtained from the expected source and that they have not been tampered with. As the environment is resource-constrained, lightweight signatures must be utilized without compromising the integrity. To ensure that the data have not been tampered with, we use a 32-bit CRC of the encrypted authorization data. To ensure tag authenticity, we use a second 32-bit CRC of the string "shared salt" + IV_0+ unique ID + KTI (key time index).

We authenticate the authorization using unique aspects of both entities to assure that "a specific tag" has been authorized by "this record keeper." The IV_{rd} IV is created by hashing a secret IV_0 and the unique RFID tag ID. This IV_{rd} is unique to the tag and the record keeper. In addition, a secret key K_1 is used. IV_0 and K_1 vary over time, as indexed by the KTI. The 32-bit CRC of the encrypted authorization is concatenated to a signature consisting of a 4-byte (32-bit) CRC of a "shared salt" + IV_0+ the unique ID + KTI. This signature string is sealed from tampering by encrypting the concatenated string with MICKEY, IV_{rd} and K_1, which has two effects. The first effect is data integrity, in which the first CRC assures the integrity of the identification and authorization fields, whereas the second CRC assures the integrity of "THIS tag" and "this record keeper."

By using two lightweight functions, the record keeper can authenticate the integrity of the authorization. This signature string is stored in a particular data field on the tag, and the two CRC values assure the integrity of all the important fields on the tag, thus providing low power [1], [63] but effective defence against tampering.

When the tag is read by the NFSDE device, the signature is decrypted with IV_{rd} and K_1 and then compared against the expected value of the CRC ("shared salt" + IV_0+ unique ID + KTI). Additionally, the CRC of the encrypted authorization is calculated. The tag is considered genuine if the decrypted field matches both CRC calculations.

2) CREATION STEPS AND SCENARIO

Typically, the creation of an RFID card to activate the NFSDE device would be part of the "on-boarding process" when a medical provider (paramedic, nurse, EMT, doctor, etc.) joins an organization. The "record keeper" might be an employee in the HR department.

Fig. 3 shows the process for creating a tag for a medical provider.

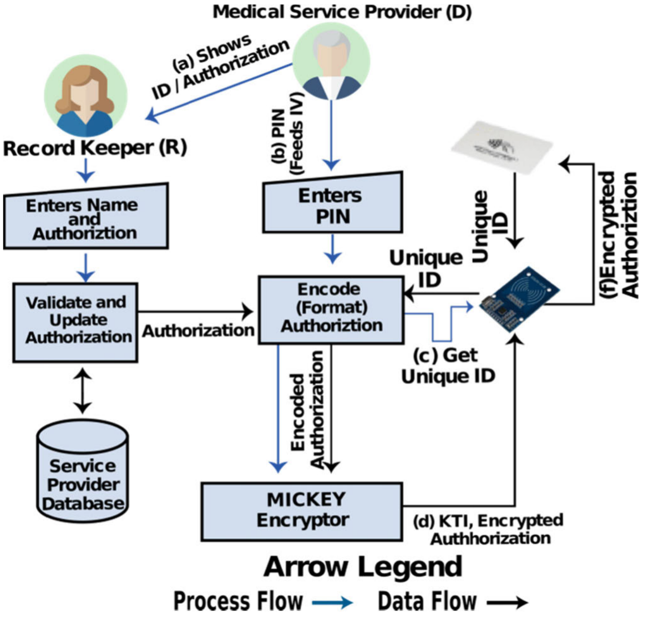
Medical Service Provider (D)
(a) Shows ID / Authorization
Record Keeper (R)
(b) PIN (Feeds IV)
Enters Name and Authoriztion
Enters PIN
Unique ID
Unique ID
(f)Encrypted Authoriztion
Validate and Update Authorization
Authorization
Encode (Format) Authorization
(c) Get Unique ID
Encoded Authorization
Service Provider Database
MICKEY Encryptor
(d) KTI, Encrypted Authhorization
Arrow Legend
Process Flow
Data Flow

FIGURE 3. Creation of provider tag.

The authentication process is as follows.

1. Interaction between the provider (D) and record keeper (R), including validation of the identity and authorization of the provider to view specific medical information, as seen in Fig. 3(a).
2. A PIN is entered by the provider as shown in Fig. 3(b). (see Step 4 for use of the PIN).
3. The card with the unique tag ID is read and recognized by the RFID reader, as shown in Fig. 3(c).
4. The PIN and the unique ID on the card are hashed for the creation of a unique IV_d ($IV_d = h(PIN,ID)$).
5. Identification and authorization are encrypted using the MICKEY protocol with K_0 and IV_d.
6. The current KTI is written onto the RFID tag by the RFID writer, as shown in Fig. 3(d).
7. The encrypted identity and authorization fields are written onto the tag by the RFID writer, as shown in Fig. 3(e)
8. A CRC hash of the encrypted field is calculated to detect whether the data have been tampered with.
9. The authentication field is encrypted via MICKEY using the two CRC values described earlier to assure both authenticity and tamper resistance.
10. The authentication field is written onto the RFID tag by the RFID writer.

B. PATIENT TAG CREATION

1) USE OF MICKEY 2.0

The patient tag contains important medical data of the patient and could be useful when the patient is in a location where wireless connections are not available, such as in a remote area or a disaster situation.

The utilization of MICKEY 2.0 has four distinct purposes within the patient tag. Its first purpose is to assure that ''THIS Tag'' has been authorized by ''THIS record keeper'' by using unique aspects of both entities. The IV_{rp} is created by hashing a secret IV_0 and the unique RFID tag ID. This IV_{rp} is unique to the tag and the record keeper. Additionally, a secret key K_1 is used. K_1 varies over time. K_1 is stored on the SD and indexed by the KTI, which is stored in a field on the patient tag.

The second purpose of MICKEY 2.0 is the encryption of the fingerprint reference data. General data protection regulation (GDPR) requires that biometric data be considered PII for privacy purposes. When the tag is created, the ''reference fingerprint'' is scanned and summarized in a 500-byte ISO/IEC 19794-2 fingerprint template [64]. This template is encrypted with MICKEY, IV_{rp} and K_1 and stored on the RFID tag.

The third purpose of MICKEY 2.0 is the encryption/decryption of medical data. The IV_P must be unique to the patient and the tag. IV_P is computed from the 500-byte unencrypted reference fingerprint template and unique tag ID. Additionally, a secret key K_0 is used, which varies over time and is indexed by the KTI. MICKEY, IV_P, and K_0 are used to encrypt the medical information before the data are stored on a tag. After the medical data are encrypted, a 32-bit CRC hash is computed for the next step.

The fourth purpose of MICKEY 2.0 is authentication. Two CRC values are calculated: one for the encrypted medical data and another for the authentication string (''shared salt'' + IV_0+ unique ID + KTI). These two fields are concatenated and encrypted with MICKEY, IV_{rp}, and K_1.

After the tag is validated as authentic, IV_{rp} and K_1 are used to decrypt the reference fingerprint template. The IV_P hash is computed from the reference fingerprint template and the unique tag ID. The fingerprint scanner compares the reference fingerprint to the recently scanned fingerprint. If the fingerprints match with respect to the ISO standard, the medical data are decrypted and displayed to the provider.

2) CREATION STEPS AND SCENARIO

The following steps describe the process for creating a patient tag. This would typically happen when the user ''checks out'' of the hospital or clinic.

The steps required for the creation of a patient tag along with its activation are depicted in Fig. 4 (details of these steps are provided in appendix A).

1. The patient presents himself or herself to the record keeper, who verifies the patient ID (i.e., patient checkout by hospital), as shown in Fig. 4(a).
2. Data are retrieved following the verification of the personal ID, as shown in Fig. 4(b).
3. The fingerprint of the patient is scanned, as illustrated in Fig. 4(c).
4. The record keeper verifies the biometric ID.
5. If the biometrics match those of the patient, the fingerprint is summarized with the ISO/IEC 19794-2 fingerprint template. This is the ''reference''

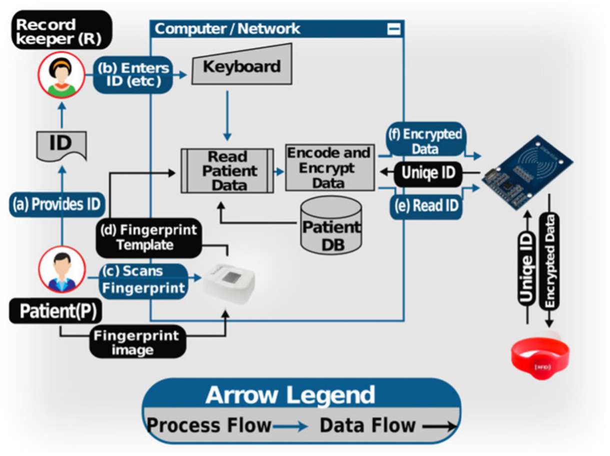

FIGURE 4. Creation of patient tag.

TABLE 2. Example of key generation, storage, and update.

Key Time Index (KTI)	K_0	K_1	IV_0
Jan 1	1AB7Yjj22	2Aabccj	9ymmdq
Jan 2	1Z227aae	Aei79jji	76yghlkjl

fingerprint, as shown in Fig. 4(d). If they do not match, the organization takes appropriate steps, which may include registering the patient. (See detailed diagram, Appendix A)

6. RFID reader scans the unique tag ID; IV_P is created ($IV_P = h(TagID,b)$, where b = Fingerprint (IV_P)). See Fig. 4(e).
7. K_0 and IV_P are used along with MICKEY to encrypt the medical data.
8. A CRC hash is computed for the encrypted medical data.
9. The fingerprint template is encrypted using IV_{rp} and K_1 for GDPR: Practitioner (GDPRP) [65], checked for compliance, and written onto the RFID tag (IV_{rp} = Hash(Tag ID, IV_0)).
10. A CRC is calculated for the string "shared salt" +IV_0+unique ID+KTI. This is concatenated to the previous CRC, encrypted, and written onto the tag, as shown in Fig. 4(f).
11. The current KTI is written onto the tag.

C. KEY GENERATION, STORAGE, AND DISTRIBUTION

The NFSDE device requires two IVs and two keys: IV_0 is used to generate one of these IVs, and the values must be changed periodically. The scheduling of this change depends on the administrative policy. Each set of keys and IVs (IV_0, K_0, K_1) must be randomly generated on schedule and assigned a KTI. The protection of these three values is paramount to device security.

When a new set of keys and IVs is generated, the updated values must be written onto the SD, as shown in the example in Table 2.

The "current" KTI is written onto the tag, and the KTI is used to look up appropriate keys and IV_0 at the time of authentication and decryption.

D. COMMUNICATION

Communication between the provider and the record keeper consists of two parts: 1) the initial creation of the provider's tag ID and 2) the regularly scheduled update of the SD containing the new KTI, the set of keys, and IV_0. The scheduling of the update is a matter of policy, but for pragmatic purposes, all SD updates should occur before the new key sets are used to encrypt the patient data.

Communication between the patient and record keeper occurs initially and whenever the tag needs to be updated. For practical reasons, it would be less expensive and more secure to issue a new tag to the patient whenever the records are updated. The old tag should be completely destroyed in a cryptographically secure manner.

The record keeper has the following functions.

1. Creation of K_0, K_1 and IV_0
2. Viewing patient information in plaintext format.
3. Confirmation of the authorization level of the provider.
4. Creation and update of the secure USB drive.
5. Confirmation of the patient identity.
6. Confirmation of the provider identity.
7. Offering authentication for tags (both provider and patient).

A medical record keeper would require physical access to the following:

1. Secure USB
2. Tags
3. Tag reader/writer
4. Fingerprint scanner (if providing patient tags)
5. PIN pad (if providing provider tags).

The medical record keeper would be granted authorization to access the patient record database and provider record system.

E. KEY ROTATION

The basic and essential features of the system security depend on the use of a secure physical device for the key. The key should be rotated regularly according to a specific schedule. The key file storage can be based on an encrypted cloud location (i.e., AWS secrets management system), and the key access can be out of band. The authorized medical provider can obtain the key file from the cloud and store it in the secure USB drive.

F. DEVICE ACTIVATION (UNLOCK)

The NFSDE device must be activated (unlocked) by an authorized medical practitioner to read the patient data. Fig. 5 illustrates the process of unlocking the NFSDE device.

The provider has his/her own unique RFID tag, which is cryptographically signed by the record keeper. The provider

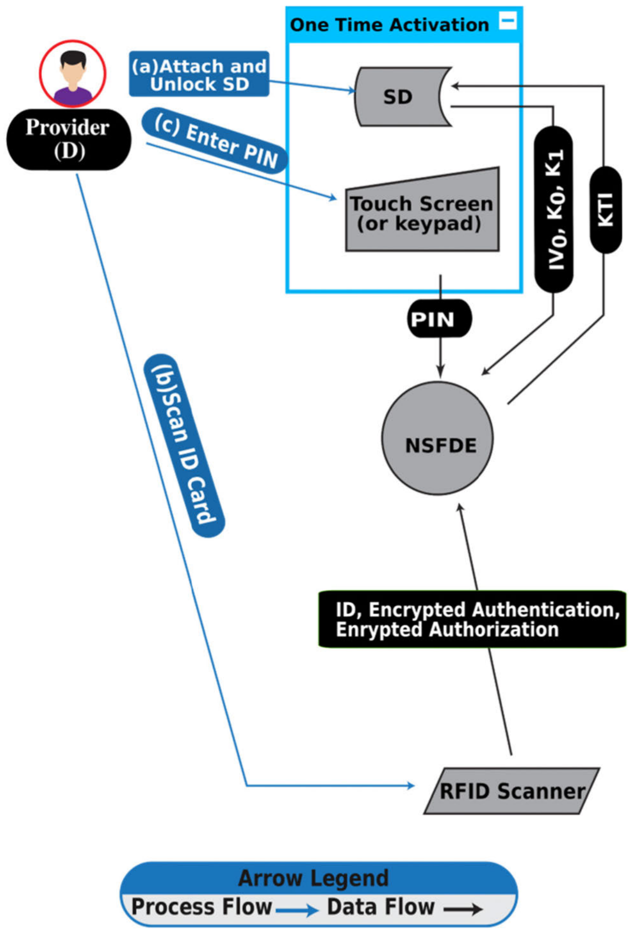

FIGURE 5. Unlocking the NFSDE.

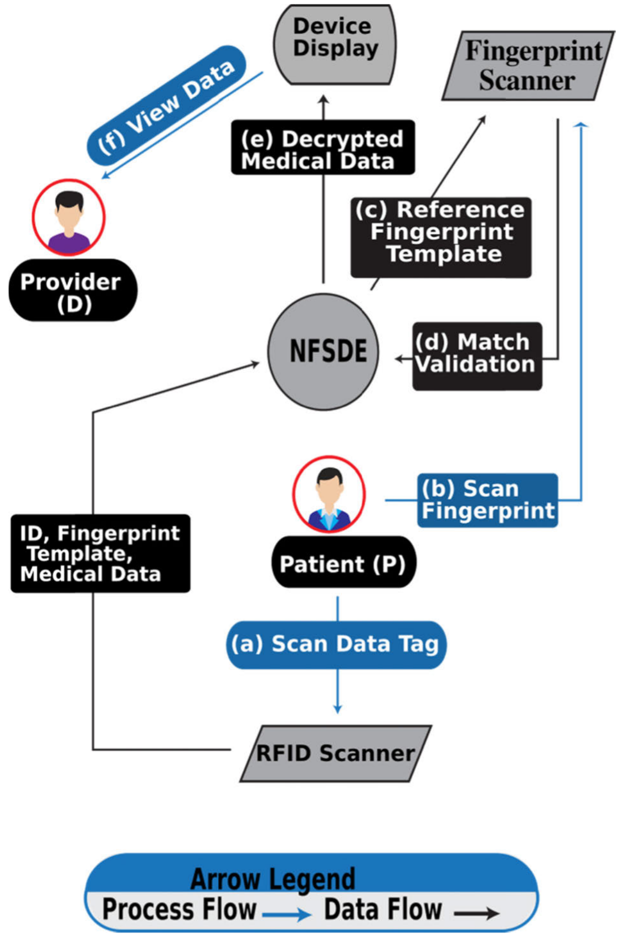

FIGURE 6. Patient display process.

also carries a secure USB drive containing the time-indexed keys and IV_0.

The provider should also provide his or her own PIN to unlock the device.

Fig. 5 illustrates the unlocking process.

The NFSDE activation (unlock) steps are as follows.

1. The provider (D) enters the password in the secure USB drive, which is connected to Raspberry Pi in the NFSDE device. This entry is required only once per session. See Fig. 5(a).
2. Fig. 5(b) shows that after the provider scans the RFID tag, the device validates whether the tag belongs to the provider and if it has a correct KTI.
3. The NFSDE device validates the authentication by reading K_1 and IV_0 from the SD. It creates IV_{rd} ($IV_{rd} = h(IV_0$, provider UID)), and by using MICKEY, K_1, and $IV_{rd,}$ it decrypts the signature and validates the authentication code and CRC.
4. If the tag is authentic, the provider enters his or her PIN, as shown in Fig. 5(c). The NFSDE determines the authorization level by decrypting the identity and authorization fields by using MICKEY, IV_d (IV_d = h(unique ID, PIN)), and K_0.
5. The device is now ready to display the data that the provider is authorized to view. (The actual process of unlocking is very similar to an ATM withdrawal.)

G. PROCEDURE FOR READING THE PATIENT MEDICAL RECORD

Fig. 6 illustrates the process of using the NFSDE device.

1. The patient RFID tag is scanned, as in Fig. 6(a).
2. The reference fingerprint is decrypted (IV_{rp}, K_1, MICKEY) and provided to the fingerprint scanner, as shown in Fig. 6(b).
3. After the patient's fingerprint is scanned, if the fingerprint matches, the reference fingerprint, the decryption of medical data commences, as seen in Fig. 6(c) and (d).
4. IV_P is computed as IV_P = h(reference fingerprint template, unique tag ID). Then, IV_P, K_0, and MICKEY are used to decrypt the medical data.
5. The device displays the medical data appropriate to the authorization level determined during device activation, as shown in Fig. 6(e) and (f).

H. NFSDE EMULATION

We created a C-language software emulation of the NFSDE device to demonstrate the major processes and components,

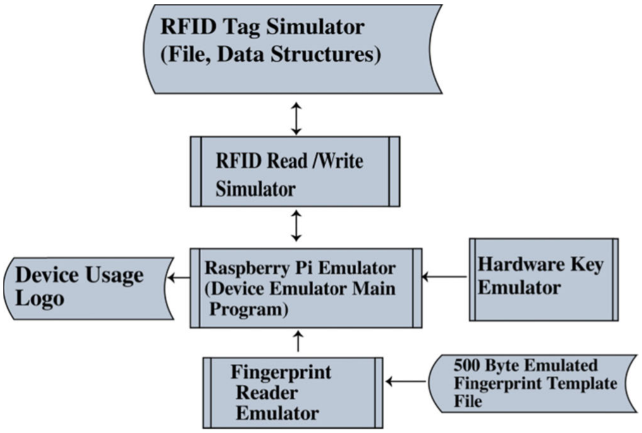
RFID Tag Simulator (File, Data Structures)
RFID Read /Write Simulator
Device Usage Logo
Raspberry Pi Emulator (Device Emulator Main Program)
Hardware Key Emulator
Fingerprint Reader Emulator
500 Byte Emulated Fingerprint Template File

FIGURE 7. Software emulation of NFSDE.

as shown in Fig. 7. The major components include a single card computer, an SD, a fingerprint scanner, an RFID reader, and an RFID writer. The major processes include the creation (and encryption) of both provider and patient tags, authentication of these tags, activation of the device by the provider, decryption of medical data, and display of medical data to the provider. The device-level access is logged (for accountability purposes) in a file called *device-log.txt*. If the emulator creates both patient and provider tag facsimiles, the process will not be logged because, in reality, this would be performed on external systems, which presumably have their own logging processes.

The ISO/IEC fingerprint template occupies 500 bytes. We used three files of 500 hex bytes each to emulate the fingerprint scanner. Each file represents a person's ISO/IEC fingerprint template. These values were random, and no attempt was made to actually emulate the referenced ISO/IEC standard. An important aspect of the emulator is the number of bytes; therefore, the size of the RFID data structure should be correct, and the encryption speed measurements should be accurate.

We used normal file operations (fread, fwrite) to emulate the RFID tag reader and writer. In addition, we created data structures that would be compatible with the 4K MIFARE standard. These data structures were written to be read from a normal file.

Furthermore, we created a command line program that includes the following options: (1) create patient tag, (2) create provider tag, (3) activate reader, (4) read patient tag, (5) unlock secure drive, and (6) change key number (time stamp emulator).

Each option prompts input during component emulation. The logic within each process follows the steps previously enumerated. All the aforementioned enumerated functions are performed or simulated.

I. TESTING AND RUNNING THE EMULATOR

The emulator functions as a command line program that displays a numbered menu. Each menu item represents a step in the creation or authentication process. See Appendix C for the menu details.

Appendix B, gives step-by-step instructions on how to run the emulator.

TABLE 3. Time to unlock.

Event	Time
Scan the ID Card (D)	1 Second, see reference See [66]
Insert and Unlock the SD	2 Seconds*
Enter the PIN	2 Seconds*

Total time to Unlock: 5 seconds
* Average Measured Manual Key Time for 4 Digit Entry

TABLE 4. Read patient data.

Event	Time
Scan the ID Card (P)	1 Second, see reference See [66]
Scan the Fingerprint	1 Second see [67]
Decode 4K Data	66291 Microseconds**

** Average Measured
Total Time to Display Encrypted Data: 2.07 Seconds

J. EMULATOR TEST RESULTS

We tested our process using the emulator. To reiterate, the unlock time occurs once per session (perhaps the beginning of a shift). Other than the manual keying time, which, of course, varies from person to person, our emulator shows that the actual authentication processing time is a matter of milliseconds.

The following Tables (3 and 4) show our results:

As is intuitively obvious, the time bottleneck is the speed at which the human can type, scan, etc. rather than a technical limitation.

VIII. ATTACK ANALYSIS

In our scenario, we used MICKEY 2.0 to encrypt five values on two types of RFID tags. (1) The system encrypted the identity and medical data of the patient (P). (2) A reference fingerprint template, which was separately encrypted, was stored on the patient's RFID tag. (3) The authentication signature provided by the record keeper (R-P) on the patient's tag was encrypted.

The fourth and fifth encrypted fields were on the provider's (D) tag. (4) The system encrypted the identity and authorization level of the provider (D). (5) Finally, the encrypted field showed the authentication signature provided by the record keeper (R-D) on the provider's tag.

From a pragmatic perspective, each attack analysis must address any unique feature in the nature of each of the five encrypted values.

To encrypt these five fields, we require three secrets: IV_0, K_0 and K_1. IV_0 is specific to the record keeper (R). As long as all three secrets are protected, the system is secure from common attack scenarios. Additionally, two non-secret parameters are required as IVs for the patient and provider (IV_P and $IV_{d,}$ respectively). By using different keys, different IVs, and three secrets, we showed that the discovery of a single secret does not result in the discovery of the other secrets.

The following methods were used to compute the IVs.

- $IV_{rd} = h(IV_0$, provider tag unique ID): This is used to encrypt/decrypt authentication (R) for the provider.
- $IV_{rp} = h(IV_0$, patient tag unique ID): This is used to encrypt/decrypt authentication (R) for the patient and the reference fingerprint.
- $IV_d = h$(PIN, provider tag unique ID): This is used to encrypt/decrypt the identity and authorization level for the provider.
- $IV_P = h$(reference fingerprint template, patient tag unique ID): This is used to encrypt/decrypt the medical data.

  The following keys are used for encryption/decryption:
- K_0 - For the provider: used to encrypt/decrypt the identity and authorization level (with IV_d).
- K_0 - For the patient: used to encrypt/decrypt medical data (with IV_p).
- K_1 - For the provider: used to encrypt/decrypt authentication data (With IV_{rd}).
- K_1 - For the patient: used to encrypt/decrypt the reference fingerprint template and authentication data. (With IV_{rp}).

A. KNOWN PLAINTEXT ATTACK

A known plaintext attack occurs when an attacker has access to both the plaintext message and the ciphertext of the same message. The attacker then attempts to derive the key from the relationship between the plaintext and ciphertext [41], [68].

In general, obtaining the "known plaintext" for any field on any tag would be nearly impossible if the attacker did not already possess the plaintext information. The MICKEY cipher is designed to prevent the attacker from deriving the key/IV pair from the keystream [16], [28].

For example, a patient who is allowed to know his or her own plaintext data could at best derive the keystream of his or her own data. As this keystream is specific to a patient's tag and biometrics, it provides no information about another patient's (or provider's) keystream. Finally, even though the patient can legitimately compute his or her own IV_P, the MICKEY 2.0 algorithm does not allow the derivation of the secret key from a known IV/plaintext [69]. This is true for all other combinations. No plaintext related to the authentication fields can be known; all encrypted values are hashes and not actual values.

In the case of the fingerprint template, which follows the ISO template standard, no two scans are identical, and therefore retrieving a known plaintext of the reference fingerprint is not feasible; even if it could be retrieved, the MICKEY 2.0 algorithm would not allow the derivation of keys and IVs from a known plaintext.

B. BRUTE FORCE ATTACK

In our case, a brute force attack would be a trial-and-error-attack strategy in which the keys and IVs are "guessed" by the attacker.

In our system, IV_0, K_0 and K_1 are unknown or "secret." These secret values are stored on a hardware-encrypted SD.

To perform a brute force attack, an attacker would need to emulate the SD with a computer [41]. Assuming that the attackers possess a legitimate tag and legitimate fingerprint scan, they would need to generate three 80-bit values correctly (IV_0, K_0, K_1), i.e., 240 bits or 2^{240}. Even if this were successful (this is not feasible), it would only compromise the values for one KTI. Therefore, a brute force attack is not truly feasible.

C. CHOSEN IV ATTACK

In a stream cipher, an IV is typically used to seed or initialize a pseudo-random function to generate a key stream. The use of the same IV for generating multiple key streams is considered unsafe. In a chosen IV attack, this "unsafe" behaviour is exploited by repeatedly using an IV of a known value to compute the value of the secret key. The MICKEY family is vulnerable to a chosen IV attack [70].

In our proposed system, we mitigated this risk by disallowing the choice of an IV. As noted earlier, there are four separate IVs. All the IVs are computed by hashing known immutable values. IV_d and IV_P use the unique ID of the RFID manufacturer. In the case of IV_{rp} and $IV_{rd,}$ a secret value (IV_0) is also part of the hash. Thus, it is simply not possible to "Choose" an IV, as this would degrade into a brute force attack because of the presence of data integrity features (hash signatures).

D. TWO-TIME PAD/REUSE KEY

The two-time pad attack (also known as a reused key attack) [71] is possible if an attacker intercepts two ciphertext messages encrypted with the same key. The attacker performs an XOR operation on both ciphertext messages followed by a frequency analysis.

For the MICKEY 2.0 stream cipher, the key is "shuffled" with the IV such that each key–IV pair provides a unique key stream.

In our system, we use four unique key–IV pairs for five encrypted fields. Furthermore, two keys are rotated (using KTI) over time. Four separate IVs are used (two for each type of tag). Each IV is generated using at least two authentication factors. In the case of the provider, this is classified as ''something you have'' + ''something you know'' (RFID tag + PIN). In the case of the patient, this is classified as ''something you have'' + ''something you are'' (RFID + biometric).

Additionally, three of the five encrypted fields are not subject to frequency analysis. The authentication fields are hashes. One of the encrypted files on the tag is a biometric template. As frequency analysis is not possible for biometric and hash fields, the ''reused'' key pair encrypting the authentication and biometric fields on the patient tag ID implies that it is not vulnerable to a reused-key attack.

E. DoS ATTACK

In a DoS attack [22], the attacker creates a situation in which a system cannot be used for its intended purpose by authorized users in a timely manner.

Our device assumes that no internet connection is available; thus, any DoS attack would require physical proximity to the device. Two options exist for a DoS attack on this device: 1) overwhelming the device with EMF interference from a nearby source and 2) physically damaging or disabling the device. In a production device, adequate shielding should be used to mitigate EMF-style attacks or interference. In actual use, the physical security of the device and its components (such as the SD) should be a matter of policy and enforcement.

F. INSIDER ATTACK

In an ''insider attack,'' an authorized privileged user deliberately uses a system in an unauthorized, malicious, or unintended manner [72]. Our system has two types of ''insiders'': the provider and the record keeper.

Because an authorized user necessarily has access to view and manipulate data, technical controls alone cannot prevent this type of attack. We provide nonrepudiation in the form of logging. The device logs activity and access by the provider, which conforms to the ''accounting'' principle in the three A's of data security. Thus, while a provider might abuse the ability to access a device, he or she would not be able to repudiate (deny) his/her activity.

The creation of the tags and distribution of the keys by the record keeper requires similar logging for nonrepudiation. Additionally, if a cloud system were used to update the SD, a secure system, such as the AWS key management system, should be used.

G. IMPERSONATION ATTACK

An impersonation attack [73] occurs when an attacker attempts to assume the identity of an authorized user for accessing the system.

The system includes three authorized roles: the record keeper, provider, and patient.

As the record keeper creates the keys and manages the tags as part of his or her daily routine, the tag-creation software and key-update software should be implemented within the security guidelines currently established by the organization.

The resilience of the device against an impersonation attack is assured by means of multi-factor authentication. For the provider, this is termed ''something you have'' (the RFID card) and ''something you know'' (the PIN), whereas for the patient, this is termed ''something you have'' (the RFID card) and ''something you are'' (the biometric fingerprint scan).

H. MAN-IN-THE-MIDDLE ATTACK

A man-in-the-middle (MIM) attack [74] occurs when the attacker secretly interposes between the sender and receiver and intercepts secret information (typically the keys). Our system requires a physical device to store and use the encryption keys.

An MIM attack can occur when the record keeper delivers the SD to the provider. Two types of controls need to be in place to mitigate this threat. 1) Administrative processes should be in place at the record keeper's facility to assure that only authorized personnel have access to the SD, and 2) physical forms of protection such as locks, robust boxes, or similar arrangements should be in place at those facilities.

If the SD were to be updated using cloud technology, a secure-key delivery system, such as the AWS Secrets Manager, would have to be used.

In addition, an MIM attack on the actual device is not feasible, as the SD is physically connected to the remainder of the system.

I. SIDE CHANNEL ATTACKS

A side channel attack [75] uses system features or weaknesses (hardware and/or operating system) rather than the features or weaknesses of the encryption algorithm to derive or extract secret information.

1) DIFFERENTIAL POWER ANALYSIS

A common side channel attack for embedded systems, such as ours, is the differential power analysis attacks (DPA) [76], which statistically analyses the power used in a device to attempt to derive the secret keys. This is especially effective when a secret key is stored within the device. When a device accesses the portion of memory (or EEPROM) that contains the secret key, it is possible to ''read'' that key using DPA.

Because DPA is a statistical method, it requires a sufficient number of data samples. This implies that long-term physical proximity to our device would be required. The threat of DPA is mitigated in our system by using an externally encrypted device on which the secret keys are stored and by using key rotation to avoid having the same key sets together.

2) CHALLENGE/RESPONSE ATTACK

An RFID-specific side channel attack has been demonstrated in [77]. This attack entails carrying out a correlation power analysis in the presence of the RFID field to break into the standard ''challenge/response'' protocol used by many RFID systems.

Our system does not use a "challenge/response" protocol. The keys are not transmitted via RFID but are physically connected to the system. Additionally, the authentication data stored on the device are not a response to a challenge but an encrypted hash. Thus, our device is not vulnerable to this type of side channel attack.

IX. OBSERVATION AND OVERALL ANALYSIS

Early adopters of the IoT and RFID discovered that low-power devices have inadequate security [1] We propose a low-cost, standards-based system that remains secure in worst-case scenarios in which access to low power, secure systems capable of operating without the internet or other connections are required. Our solution includes key-management and key-update solutions for a secure RFID reader that is neither dependent on the internet nor on wireless communication, as discussed in subsection A below. Specifically, we proposed and emulated an example of a real-life application for patients located in areas without reliable communication (including Wi-Fi, internet, 4G, or 5G networks), in emergency situations, and in remote areas. This system can provide access to patient health records under adverse conditions. The proposed system provides a low-cost, reliable, and secure solution to the CIA triad normally provided by a PKI. By adopting a fundamentally different approach and using software and off-the-shelf hardware, we avoided a more expensive "custom-hardware" asymmetric solution [10], [11].

A. MULTI-FACTOR AUTHENTICATION WITHOUT CONNECTIVITY

Our approach is a combination of "something you know" (i.e., a PIN and SD passcode), "something you have" (i.e., an RFID card and SD) and "something you are" (i.e., biometrics). In traditional key management, the entire key is stored on "something you have" (i.e., a data drive with the shared key) or is "something you know" (a password or PIN). When the keys are changed, the key must be transmitted and stored, or the password or PIN must be entered. The key and PIN must be defended from interception. Instead of using public-key encryption in NFSDE, we mitigate this vulnerability with the SD. The SD does not contain the entire set of required keys but instead contains the parameters used to compute the keys. These parameters are secured by a passcode that can be unique to each SD. It is not unreasonable to expect a health service provider to show due care in protecting the SD or to have it updated on a schedule. Updating the parameters on the SD is a matter of administrative procedure, rather than a technical challenge. The SD could be updated by taking it to a secure facility or by using a secure system such as the AWS Secrets Manager. A debit-card-like distribution system could be used where the SD is physically delivered in one package by a courier and the passcode could be delivered electronically using a service such as "One Time Secret" [78]. In any case, the administration is no more complicated than any other key distribution system.

Before the private data are accessed, a total of six factors must be mutually authenticated, and all three types of factors are used at least once: something you have - the patient RFID card, the provider RFID card, the SD; something you know - the provider PIN, the SD card passcode; and something you are - the patient biometric. The decryption keys are not available until all factors have been presented and mutually authenticated. In other words, even if the SD is compromised, access to private data is not possible without four other authenticating factors. Therefore, using the SD to store parameters is more secure than traditional key distribution and no more complex than obtaining a debit card with a new PIN.

B. PRIVACY BY DESIGN

The fourth principle of privacy by design [40] is that privacy should be "integral to the system, without diminishing functionality." A functional requirement in emergency situations is rapid access to critical data. Thus, we designed and emulated a system in which authentication, access, and accounting are no more complicated than withdrawing cash from an ATM. In addition, we were able to mitigate or eliminate known cryptographic attacks by using the principles of security-in-depth [47].

By using lightweight keystream-based cryptography and a low-cost, reliable, secure key distribution system, our system was able to achieve the CIA triad in a worst-case life-and-death scenario.

C. KEY DISTRIBUTION WITHOUT CONNECTIVITY

A customary solution for key distribution is public key encryption. Typically, public key encryption is used for authentication, confidentiality, and integrity; this requires a PKI (typically, connectivity is required for access to a certifying authority) and significant CPU power, both of which are absent in our scenario. Therefore, we achieved authentication, confidentiality, and integrity without PKI by using the MICKEY 2.0 stream cipher, hardware-based security, and unique processes. Our novel approach to achieve this is to compute each of four IV–key pairs by using a combination of hardware-secured values, shared values, and embedded RFID tags.

In conventional symmetric cryptography, it is necessary for the sender to create a key and transmit it to the receiver. The number of keys required is known to be $n(n-1)/2$, where n is the number of parties that need to communicate. The risk of the key being intercepted by a third party and the requirement of creating a unique key for each sender/receiver pair is known as "the key distribution problem" [44].

Suppose Alice, Bob, Charlie, and Dave want to exchange secure messages, in which case six keys would have to be created. Each key must be transmitted from the sender to the receiver such that an enemy, say Eve, does not have the opportunity to copy or intercept the key. The key-distribution problem to be solved is two-fold: 1) how does Alice create unique keys for Bob, Charlie, and Dave? 2) How does she transmit those keys to Bob, Charlie, and Dave without Eve intercepting them?

We solved the key-distribution problem and implemented the principle of security comprehensively [47] by utilizing a separately encrypted off-the-shelf hardware device in which to store the key parameters that are used to compute the shared symmetric key for each sender/receiver pair. The attack analysis showed that this same solution mitigated common RFID attacks.

As the device is designed specifically not to connect to the internet, it can also be used for out-of-band authentication (and possibly authorization).

Our sample application is intended to inspire advances in eHealth technology. Furthermore, the low power, high-quality encryption we propose would be especially applicable to IoT security as well as security in general.

X. CONCLUSION AND FUTURE WORK

Owing to the rapid growth of RFID technology and its applications, meeting security-related needs in this field remains challenging, especially when the internet connection malfunctions or is unavailable. In response, we developed an NFSDE device with a secure protocol to enhance security in general and RFID technology in particular.

Previously, others solved the problem by using custom-designed hardware chips rather than using a software solution. Our NFSDE device takes a fundamentally different approach by using custom software and off-the-shelf hardware, thereby enhancing its flexibility and affordability by allowing the use of standardized hardware.

The protocol and its system provide a framework for RFID security, as demonstrated by developing a prototype eHealth application, and it is likely to be feasible for other applications. Moreover, we described the software emulation of the NFSDE device and the MICKEY 2.0 cipher implementations for encryption in addition to testing procedures such as tag identification, fingerprint scanner emulation, and unlocking a device by a medical provider. This software emulation provides a framework for device functionality, security, and confidentiality. Our protocol can be implemented using different lightweight ciphers, and reduced versions of other ciphers, such as AES, may be suitable for RFID technology. Consequently, future research on this topic could focus on the following:

1. Implementing the NFSDE device with the protocol in applications other than eHealth;
2. Designing housing for the NFSDE device to make it suitable for field use;
3. Designing a single-board device to integrate the major components while maintaining the functional design;
4. Adapting the security protocol for ciphers other than MICKEY 2.0.

In summary, by offering LWE with simple design, low power consumption, and non-dependency on the internet, our prototype NFSDE device serves to advance eHealth security as well as the overall security of RFID technology. Our results indicate that the LWE of the device, when coupled with a solution for secure key storage and exchange, can provide a representative process for public key exchange without relying on an internet connection. In addition, our system reduces the margin for human error significantly. As a result, this system enhances the overall RFID tag security, which can guide further research on secure communication when a wireless internet connection is either malfunctioning or unavailable. Moreover, the device is an effective tool for out-of-band authentication and identification.

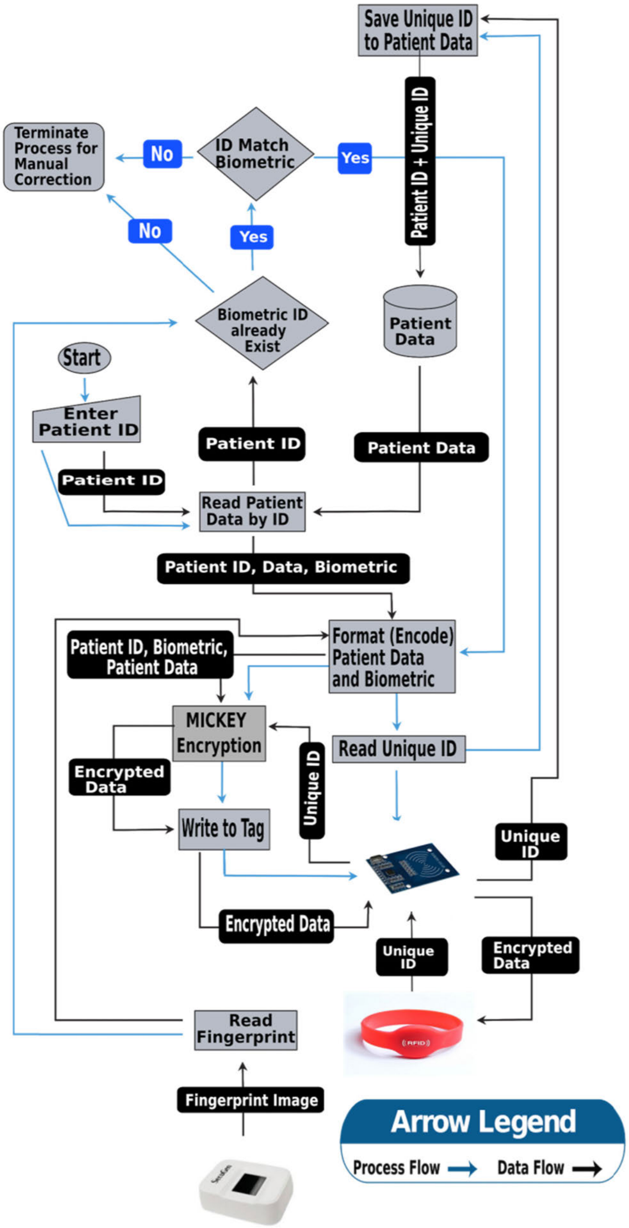

FIGURE 8. Detailed flow of the patient tag creation process.

APPENDIXES

APPENDIX A

See Fig. 8.

APPENDIX B

The emulator functions as a command line program that displays a menu with the following options.

These are the main options suggested for display on the user interface and can be modelled according to user preferences.

A. MAIN MENU

1. Create Patient Tag
2. Create Provider Tag
3. Activate Reader
4. Read Patient Tag
5. Unlock the USB
6. Change Key Number (Time stamp emulator)
7. Exit

B. TEST RUN USING THE EMULATOR

1. To unlock the USB drive by using option 5, which makes the key available, the password is hard coded and displayed for convenience within the simulator. If the USB device is not unlocked, K_1, K_0, and IV_0 are not available to the simulator, and error messages are displayed. If the SD card is not "unlocked," no creation, read, or activation can be performed.
2. Option 6 allows the simulation of a KTI rotation. For demonstration purposes, only two key sets were provided. This proves it is possible to encrypt the provider with one set of keys and the patient with another.
3. Creating a provider tag by using option 2 prompts the following process.
   a. The provider's identification, PIN, and authorization are entered.
   b. A 7-byte unique ID is generated randomly.
   c. Encryption is performed, and authentication code (as described above) is generated.
   d. A file of the form "[uniqueid].enc" is used to simulate the tag (in this case, the provider tag.) This includes an unencrypted value of "2" in the tag type field to ensure that subsequent scans "know" this is a provider tag rather than a patient tag.
   e. A plaintext file of the form "[uniqueid].txt" is also created for checking the accuracy of the decryption process.
4. Option 1 for creating a patient tag has two major features:
   a. Emulation of the reference fingerprint scan, which is performed by simply specifying one of the three hex files provided to serve as the reference fingerprint (we used three to enable us to emulate incorrect or failed scans).
   b. Emulation of reading and encrypting the identity and "medical data." We used a random name generator and a random string generator to emulate the patient identity and medical data.
   c. A file of the form "[uniqueid].enc" is used to simulate the tag (in this case, the patient tag). This includes the unencrypted value of "1" in the tag type field to enable subsequent scans to "know" this is a patient tag rather than a provider tag.
   d. A plaintext file of the form "[uniqueid].txt" is also created to verify the accuracy of the decryption process.
5. Option 3 (reader activation) begins by prompting the provider tag to be scanned. This scan is emulated by entering the filename of a provider tag that has already been created ("[uniqueid].enc"). If the file contains 2 (provider) in the tag type field, the authentication signature is decrypted and checked (including CRC). If the signature matches all acceptance criteria and it is assured that the data have not been tampered with, the device is "activated" and the authorization level of the provider is stored in the device memory.
6. Option 4 reads the patient tag. If the device has not been activated (Option 3), Option 4 fails immediately, prompting for activation. The RFID card is "scanned" by entering the filename of a previously created patient tag ("[uniqueid].enc"). If the file contains the patient value of 1 in the tag type field, the authentication signature is decrypted and checked (including CRC). Further, if the signature matches all acceptance criteria and it is assured that the data have not been tampered with, fingerprint scanning is performed in the next step. Fingerprint scanning is emulated by entering one of the fingerprint file numbers. A "good scan" is emulated by entering the same number as in the reference fingerprint template, whereas a "bad scan" is emulated by entering one of the other numbers. If the fingerprint matches, the medical data will be decrypted and displayed (based on the authorization level at activation.)

APPENDIX C

Physically unclonable functions (PUFs) have two major functional benefits: key generation and lightweight authentication [30], [31]. We do not need PUFs for key generation but could use them for lightweight authentication.

When an environmentally stable PUF becomes readily available for emergency use, the protocol we have developed should dove-tail into this technology. The emulator functions can be modified to use the protocol.

In a scenario that requires less frequent key rotation and in which potential problems caused by the environment are not life-threatening, off-the-shelf components could be replaced by commercially available special order CRFID tags. These tags, for providers only, could serve the same function as the SD. IV_0, K_1 and K_0 could be encrypted and stored on the providers' CRFID cards and decrypted when his or her PIN is authenticated.

To pursue this line of inquiry, the software-based components of the emulator could be adopted to experiment on the best way to implement the protocol to guide the designer for specific hardware implementations without limiting designer creativity [79]. Adding a fuzzy extractor [31] to the emulator would definitely give us some better insights on implementation details necessary for PUFs.

For example, the RFID tag simulator component might be replaced by a CRFID with a PUF simulator component. The USB component may, for example, be used for storing expected responses to registered challenges or may be replaced or removed entirely.

REFERENCES

[1] J. Kim, J. Cho, and D. Park, "Low-power command protection using SHA-CRC inversion-based scrambling technique for CAN-integrated automotive controllers," in *Proc. IEEE Conf. Dependable Secure Comput. (DSC)*, Kaohsiung, Taiwan, Dec. 2018, pp. 1–2.

[2] BBC News. (2019). *Fridge Sends Spam Emails*. Accessed: Jun. 6, 2019. [Online]. Available: https://www.bbc.com/news/technology-25780908

[3] M. Kaur, M. Sandhu, N. Mohan, and P. S. Sandhu, "RFID technology principles, advantages, limitations & its applications," *Int. J. Comput. Elect. Eng.*, vol. 3, no. 1, pp. 151–157, Jan. 2011, doi: 10.7763/IJCEE.2011.V3.306.

[4] Y. Xiao, X. Shen, B. O. Sun, and L. Cai, "Security and privacy in RFID and applications in telemedicine," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 64–72, Apr. 2006, doi: 10.1109/MCOM.2006.1632651.

[5] S. Li, T. Tryfonas, and H. Li, "The Internet of Things: A security point of view," *Internet Res.*, vol. 26, no. 2, pp. 337–359, Apr. 2016, doi: 10.1108/IntR-07-2014-0173.

[6] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of Things security: A survey," *J. Netw. Comput. Appl.*, vol. 88, pp. 10–28, Jun. 2017, doi: 10.1016/j.jnca.2017.04.002.

[7] M. Chamekh, M. Hamdi, S. El Asmi, and T.-H. Kim, "Security of RFID based Internet of Things applications: Requirements and open issues," in *Proc. 15th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Hammamet, Tunisia, Mar. 2018, pp. 699–703.

[8] J. J. Stapleton, *Security Without Obscurity: A Guide to Confidentiality, Authentication, and Integrity*. New York, NY, USA: Auerbach, 2014.

[9] S. D. Galbraith, "Authenticated key exchange for SIDH," *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 266, Mar. 2018.

[10] Y. Zhang, L. Xu, Q. Dong, J. Wang, D. Blaauw, and D. Sylvester, "Recryptor: A reconfigurable cryptographic cortex-M0 processor with in-memory and near-memory computing for IoT security," *IEEE J. Solid-State Circuits*, vol. 53, no. 4, pp. 995–1005, Apr. 2018, doi: 10.1109/JSSC.2017.2776302.

[11] F. Conti, R. Schilling, P. D. Schiavone, A. Pullini, D. Rossi, F. K. Gürkaynak, M. Muehlberghuber, M. Gautschi, I. Loi, G. Haugou, S. Mangard, and L. Benini, "An IoT endpoint system-on-chip for secure and energy-efficient near-sensor analytics," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 9, pp. 2481–2494, Sep. 2017, doi: 10.1109/TCSI.2017.2698019.

[12] (2019). *The Estream Portfolio Page*. Accessed: Jun. 6, 2019. [Online]. Available: http://www.ecrypt.eu.org/stream/

[13] A. Canteaut, S. Carpov, C. Fontaine, J. Fournier, B. Lac, M. Naya-Plasencia, R. Sirdey, and A. Tria, "End-to-end data security for IoT: From a cloud of encryptions to encryption in the cloud," in *Proc. IEEE Conf. (Cesar)*, Nov. 2017, pp. 1–21.

[14] L. Ertaul and A. Woodall, "IoT security: Performance evaluation of grain, MICKEY, and trivium-lightweight stream ciphers," in *Proc. Int. Conf. Secur. Manage. (SAM)*, Hayward, CA, USA, 2017, pp. 32–38.

[15] S. Banik, "Some studies on selected stream cipher, analysis, fault attack & related results," Ph.D. dissertation, Indian Stat. Inst., Kolkata, India, 2015.

[16] S. Banik, S. Maitra, and S. Sarkar, "Improved differential fault attack on MICKEY 2.0," Cryptol. ePrint Arch., IACR, Lyon, France, Tech. Rep. 2013/029, 2013.

[17] Y. Su, Y. Gao, O. Kavehei, and D. C. Ranasinghe, "Hash functions and benchmarks for resource constrained passive devices: A preliminary study," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Kyoto, Japan, Mar. 2019, pp. 1020–1025.

[18] J.-Y. Lee, W.-C. Lin, and Y.-H. Huang, "A lightweight authentication protocol for Internet of Things," in *Proc. Int. Symp. Next-Gener. Electron. (ISNE)*, Kwei-Shan, Taiwan, May 2014, pp. 1–2.

[19] W. J. Buchanan, S. Li, and R. Asif, "Lightweight cryptography methods," *J. Cyber Secur. Technol.*, vol. 1, nos. 3–4, pp. 187–201, Sep. 2017, doi: 10.1080/23742917.2017.1384917.

[20] M. Katagi and S. Moriai, *Lightweight Cryptography for the Internet of Things*. Tokyo, Japan: Sony, 2008.

[21] A. Biryukov and L. P. Perrin. (2017). *State of the Art in Lightweight Symmetric Cryptography*. Accessed: Jun. 19, 2017. [Online]. Available: http://hdl.handle.net/10993/31319

[22] O. Billet, J. Etrog, and H. Gilbert, "Lightweight privacy preserving authentication for RFID using a stream cipher," in *Proc. Int. Workshop Fast Softw. Encryption*, Berlin, Germany, Feb. 2010, pp. 55–74.

[23] W. Wu and L. Zhang, "LBlock: A lightweight block cipher," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, Berlin, Germany, Jun. 2011, pp. 327–344.

[24] F. Karakoç, H. Demirci, and A. E. Harmancı, "Impossible differential cryptanalysis of reduced-round Lblock," in *Information Security Theory and Practice. Security, Privacy and Trust in Computing Systems and Ambient Intelligent Ecosystems*. Berlin, Germany: Springer, Jun. 2012, pp. 179–188.

[25] A. Poschmann, G. Leander, K. Schramm, and C. Paar, "A family of lightweight block ciphers based on DES suited for RFID applications," in *Proc. Workshop RFID Secur.*, vol. 6, Jul. 2006, pp. 1–16.

[26] F. K. Gurkaynak. *Hardware Evaluation of eSTREAM Candidate Algorithms*. Accessed: Oct. 21, 2019. [Online]. Available: http://asic.ethz.ch/estream/E.png

[27] V. Mikhalev, F. Armknecht, and C. Müller, "On ciphers that continuously access the non-volatile key," in *Proc. IACR Trans. Symmetric Cryptol.*, Feb. 2016, pp. 52–79.

[28] S. Babbage and M. Dodd, "The stream cipher MICKEY 2.0," ECRYPT Stream Cipher, EU ECRYPT Netw., Denmark, U.K., Tech. Rep., 2006. [Online]. Available: https://www.ecrypt.eu.org/stream/index.html and https://www.ecrypt.eu.org/stream/e2-mickey.html

[29] Y. Bendavid, N. Bagheri, M. Safkhani, and S. Rostampour, "IoT device security: Challenging 'a lightweight RFID mutual authentication protocol based on physical unclonable function,'" *Sensors*, vol. 18, no. 12, pp. 4444–4463, Oct. 2018, doi: 10.3390/s18124444.

[30] Y. Gao, Y. Su, W. Yang, S. Chen, S. Nepal, and D. C. Ranasinghe, "Building secure SRAM PUF key generators on resource constrained devices," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Kyoto, Japan, Mar. 2019, pp. 912–917.

[31] Y. Gao, Y. Su, L. Xu, and D. C. Ranasinghe, "Lightweight (reverse) fuzzy extractor with multiple reference PUF responses," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1887–1901, Jul. 2019, doi: 10.1109/TIFS.2018.2886624.

[32] T. Takpor and A. A. Atayero, "Integrating Internet of Things and EHealth solutions for students' healthcare," in *Proc. World Congr. Eng.*, London, U.K., vol. 1, 2015, pp. 1–4.

[33] J. Hiller, J. Pennekamp, M. Dahlmanns, M. Henze, A. Panchenko, and K. Wehrle, "Tailoring onion routing to the Internet of Things: Security and privacy in untrusted environments," in *Proc. 27th IEEE Int. Conf. Netw. Protocols*, Chicago, IL, USA, Oct. 2019, pp. 1–12.

[34] J. J. P. C. Rodrigues, D. B. De Rezende Segundo, H. A. Junqueira, M. H. Sabino, R. M. Prince, J. Al-Muhtadi, and V. H. C. De Albuquerque, "Enabling technologies for the Internet of health things," *IEEE Access*, vol. 6, pp. 13129–13141, 2018, doi: 10.1109/ACCESS.2017.2789329.

[35] Y. Ma, Y. Wu, J. Ge, and L. I. Jun, "An architecture for accountable anonymous access in the Internet-of-Things network," *IEEE Access*, vol. 6, pp. 14451–14461, 2018, doi: 10.1109/ACCESS.2018.2806483.

[36] D. Mocrii, Y. Chen, and P. Musilek, "IoT-based smart homes: A review of system architecture, software, communications, privacy and security," *Internet Things*, vols. 1–2, pp. 81–98, Sep. 2018, doi: 10.1016/j.iot.2018.08.009.

[37] M. Ammar, G. Russello, and B. Crispo, "Internet of Things: A survey on the security of IoT frameworks," *J. Inf. Secur. Appl.*, vol. 38, pp. 8–27, Feb. 2018, doi: 10.1016/j.jisa.2017.11.002.

[38] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, "SoK: Security evaluation of home-based IoT deployments," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2019, pp. 208–226.

[39] M. T. Fernández-Caramés, P. Fraga-Lamas, M. Suárez-Albela, and L. Castedo, "Reverse engineering and security evaluation of commercial tags for RFID-based IoT applications," *Sensors*, vol. 17, no. 1, pp. 28–59, Oct. 2017, doi: 10.3390/s17010028.

[40] A. Cavoukian, *Privacy by Design: The 7 Foundational Principles*. Toronto, ON, Canada: Information and Privacy Commissioner of Ontario, 2009.

[41] H. C. van Tilborg and S. Jajodia, *Encyclopedia of Cryptography and Security*. Berlin, Germany: Springer, 2014.

[42] C. Metz, ''AAA protocols: Authentication, authorization, and accounting for the Internet,'' *IEEE Internet Comput.*, vol. 3, no. 6, pp. 75–79, Nov./Dec. 1999, doi: 10.1109/4236.807015.

[43] T. K. Kuppusamy, L. A. DeLong, and J. Cappos, ''Uptane: Security and customizability of software updates for vehicles,'' *IEEE Veh. Technol. Mag.*, vol. 13, no. 1, pp. 66–73, Mar. 2018, doi: 10.1109/MVT.2017.2778751.

[44] A. Singh, ''Centralized key distribution using quantum cryptography,'' *Int. J. Comput. Sci. Mobile Comput.*, vol. 6, no. 7, pp. 208–213, Jul. 2017.

[45] C. Krull, L. F. McMillan, R. M. Fewster, R. van der Ree, R. Pech, T. Dennis, and M. C. Stanley, ''Testing the feasibility of wireless sensor networks and the use of radio signal strength indicator to track the movements of wild animals,'' *Wildlife Res.*, vol. 45, no. 8, pp. 659–667, Jan. 2019, doi: 10.1071/WR18013.

[46] L. Sweeney, ''Replacing personally-identifying information in medical records, the scrub system,'' in *Proc. Amer. Med. Inform. Assoc. Annu. Fall Symp.*, Washington, DC, USA, 1996, pp. 333–337.

[47] M. Sourour, B. Adel, and A. Tarek, ''Ensuring security in depth based on heterogeneous network security technologies,'' *Int. J. Inf. Secur.*, vol. 8, no. 4, pp. 233–246, Aug. 2009, doi: 10.1007/s10207-009-0077-2.

[48] S. K. Sah, S. Shakya, and H. Dhungana, ''A security management for Cloud based applications and services with diameter-AAA,'' in *Proc. Int. Conf. Issues Challenges Intell. Comput. Techn. (ICICT)*, Ghaziabad, India, Feb. 2014, pp. 6–11.

[49] Amazon.com. *USB 16GB (Encrypted USB3.0 Flash Drive (256-Bit AES Encryption)*. Aug. Jun. 27, 2019. [Online]. Available: https://www.amazon.com/Integral-Secure-Encrypted-256-bit-Encryption/dp/B00TUBOTEI/ref=sr_1_6?qid=1561614555&refinements=p_n_feature_keywords_browse-bin%3A6813186011&s=pc&sr=1-6

[50] E. Harari, R. D. Norman, and S. Mehrotra, ''Flash EEPROM system,'' U.S. Patent 529 714 8A, Mar. 22, 1994.

[51] L. Chen, K. Cong, and S. Sultana, ''Side-channel attack detection using hardware performance counters,'' U.S. Patent 16 234 085, May 2, 2019.

[52] J. A. Nix, ''Cryptographic unit for public key infrastructure (PKI) operations,'' U.S Patent 15 575 908, May 24, 2018.

[53] E. Upton and G. Halfacree, *Raspberry Pi User Guide*. Hoboken, NJ, USA: Wiley, 2014.

[54] A. Sforzin, F. G. Mármol, M. Conti, and J.-M. Bohli, ''RPiDS: Raspberry Pi IDS—A fruitful intrusion detection system for IoT,'' in *Proc. Int. IEEE Conf. Ubiquitous Intell. Comput. Adv. Trusted Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart World Congr. (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)*, Toulouse, France, Jul. 2016, pp. 440–448.

[55] D. Puthal, S. P. Mohanty, P. Nanda, E. Kougianos, and G. Das, ''Proof-of-authentication for scalable blockchain in resource-constrained distributed systems,'' in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, Jan. 2019, pp. 1–5.

[56] M. T. Wayland and M. Landgraf, ''A cartesian coordinate robot for dispensing fruit fly food,'' *J. Open Hardw.*, vol. 2, no. 1, pp. 1–8, Jul. 2018, doi: 10.5334/joh.9.

[57] Adafruit Industries. *'Adafruit Pitft—320×240 2.8' TFT+Touchscreen for Raspberry Pi*. Accessed: Jun. 27, 2019. [Online]. Available: https://www.adafruit.com/product/1601

[58] SecuGen. *Fingerprint Reader User Guide*. Accessed: Jun. 8, 2019. [Online]. Available: https://secugen.com/guides/

[59] *MFRC522 Standard Performance MIFARE and NTAG Frontend. Rev. 3.9–27 April 2016*, Standard 112139, Semiconductors NXP. Accessed: Jun. 27, 2019. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf

[60] Stronglink-Rfid.Com. (2019). *MIFARE Classic 4K Contactless Smart Card*. Accessed: Jun. 15, 2019. [Online]. Available: http://www.stronglink-rfid.com/en/rfid-cards/mifare-4k.html/

[61] Z. Xiong, Y. Wu, C. Ye, X. Zhang, and F. Xu, ''Color image chaos encryption algorithm combining CRC and nine palace map,'' *Multimedia Tools Appl.*, vol. 78, no. 22, pp. 31035–31055, Nov. 2019, doi: 10.1007/s11042-018-7081-3.

[62] T. Fuhr, G. Leurent, and V. Suder, ''Collision attacks against CAESAR candidates,'' in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT)*, Berlin, Germany, Dec. 2015, pp. 510–532.

[63] S. Farrell, *Low-Power Wide Area Network (LPWAN) Overview*, document RFC: 8376, 2018.

[64] T.-P. Chen, W.-Y. Yau, and X. Jiang, ''ISO/IEC standards for on-card biometric comparison,'' *Int. J. Biometrics*, vol. 5, no. 1, pp. 30–52, Jan. 2013, doi: 10.1504/IJBM.2013.050732.

[65] P. de Hert and V. Papakonstantinou, ''The new general data protection regulation: Still a sound system for the protection of individuals?'' *Comput. Law Secur. Rev.*, vol. 32, no. 2, pp. 179–194, Apr. 2016, doi: 10.1016/j.clsr.2016.02.006.

[66] Secugen.com. (2019). *The Highest Quality in Fingerprint Biometrics at Affordable Prices*. Accessed: Oct. 5, 2019. [Online]. Available: http://secugen.com/wp-content/uploads/SecuGenHamsterPro10_HU10_1811.pdf

[67] Nxp.com. (2019). *Standard Performance MIFARE and NTAG Frontend*. Accessed: Oct. 5, 2019. [Online]. Available: https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf

[68] J. Katz, A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1996.

[69] C. Manifavas, G. Hatzivasilis, K. Fysarakis, and Y. Papaefstathiou, ''A survey of lightweight stream ciphers for embedded systems,'' *Secur. Commun. Netw.*, vol. 9, no. 10, pp. 1226–1246, Jul. 2016, doi: 10.1002/sec.1399.

[70] S. Maitra, ''Chosen IV cryptanalysis on reduced round ChaCha and Salsa,'' *Discrete Appl. Math.*, vol. 208, pp. 88–97, Jul. 2016, doi: 10.1016/j.dam.2016.02.020.

[71] S. Fluhrer, ''Cryptanalysis of ring-LWE based key exchange with key share reuse,'' *IACR Cryptol. ePrint Arch.*, vol. 2016, pp. 85–91, Jan. 2016.

[72] X. Yang, C. Xu, and C. Li, ''A privacy model for RFID tag ownership transfer,'' *Secur. Commun. Netw.*, vol. 2017, no. 1, pp. 1–10, Mar. 2017, doi: 10.1155/2017/5084636.

[73] J. Munilla, M. Burmester, and A. Peinado, ''Attacks on ownership transfer scheme for multi-tag multi-owner passive RFID environments,'' *Comput. Commun.*, vol. 88, pp. 84–88, Aug. 2016, doi: 10.1016/j.comcom.2016.05.007.

[74] M. Brooks and B. Yang, ''A man-in-the-middle attack against opendaylight SDN controller,'' in *Proc. 4th Annu. ACM Conf. Res. Inf. Technol.*, New York, NY, USA, Sep. 2015, pp. 45–49.

[75] T. Zhang, Y. Zhang, and R. B. Lee, ''CloudRadar: A real-time side-channel attack detection system in clouds,'' in *Proc. Int. Symp. Res. Attacks, Intrusions, Defenses*, Cham, Switzerland, Sep. 2016, pp. 118–140.

[76] C. Chen, T. Eisenbarth, I. von Maurich, and R. Steinwandt, ''Differential power analysis of a McEliece cryptosystem,'' in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, Cham, Switzerland, Jan. 2015, pp. 538–556.

[77] T. Kasper, D. Oswald, and C. Paar, ''New methods for cost-effective side-channel attacks on cryptographic RFIDs,'' in *Proc. Workshop RFID Secur.*, Princeton, NJ, USA, Jan. 2009, pp. 1–15.

[78] Onetimesecret.com. (2019). *Share a Secret—One Time*. Accessed: Oct. 5, 2019. [Online]. Available: https://onetimesecret.com/

[79] J. K. Oberg, J. Valamehr, R. Kastner, and T. Sherwood, ''Generating hardware security logic,'' U.S. Patent 10 289 873 B2, May 14, 2019.

AHMED ALAMER received the Graduate Diploma degree from Adelaide University, the bachelor's degree in mathematical sciences from King Khalid University, and the master's degree from QUT University, majoring in mathematical sciences with a research focus in cryptography. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Computer Engineering, La Trobe University, where he has published and under-review articles in cryptology, neural networks applications in security, statistical analysis and randomness testing, and lightweight cryptosystem design and cryptanalysis. He currently focuses on the applications of mathematics in the field of cryptanalysis and security. He has been a Lecturer with the University of Tabuk since 2012, where he has organized several workshops in mathematics applications.

BEN SOH (S'89–M'92–SM'03) received the Ph.D. degree in computer science and engineering from La Trobe University, Melbourne, Australia, in 1995. He is currently an Associate Professor with the Department of Computer Science and Computer Engineering, La Trobe University. He had numerous successful Ph.D. graduates. He has authored more than 180 peer-reviewed research articles. He has made significant contributions in various research areas, including fault-tolerant and secure computing, and web services.

AHMED H. ALAHMADI received the Ph.D. degree in computer science and engineering from La Trobe University. His Ph.D. research was in e-health business requirements engineering. Since then, he has published various peer-reviewed research articles. He worked as the Dean of the College of Computer Science and IT, AlBaha University. He is currently an Assistant Professor with the Department of Computer Science and Information, Taibah University, Saudi Arabia. He is also the Dean of the Khaybar Community College, Taibah University. In addition to research, he is also skilled in accreditation and college recruiting. He has made significant contributions in various research areas, including e-health, software engineering, business process modelling, requirements engineering, and process mining. He also has a demonstrated history of working in the higher education industry.

DAVID E. BRUMBAUGH is a Software Engineering consultant, speaker, and author. He has been practicing his profession, since 1986. His clients have included the Security Research and Operations Group at CISCO, Caterpillar, American Greetings, Colonial Penn, and the Museum of Science and Industry, Chicago. He has served as an expert witness in Federal court in a software copyright infringement case. His published works include *Object-Oriented Development, Building CASE Tools in C++* (John Wiley and Sons, 1993), *Object-Oriented Programming in C* (C User's Journal, July 1990) and *How to Encrypt Large Messages with Asymmetric Keys* (Site Point, January 19, 2015). He was a subject matter expert author for Microsoft's Exam 98-374 MTA: Gaming Development Fundamentals. He has been a featured speaker at WordCamp Dallas, Fort Worth, and WordCamp Atlanta, focusing on security fundamentals for WordPress. As an Artificial Intelligence and Physics Simulation Programmer, for a major game developer, his credits include, NFL Full Contact Football, Indy Racing, Jeff Gordon XS Racing, and Centipede 3D.