Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
A Secure and reliable RFID authentication protocol using digital schnorr cryptosystem for IoT-enabled healthcare in COVID-19 scenario

Mohd Shariq a, Karan Singh a, Mohd Yazid Bajuri b, Athanasios A. Pantelous c, Ali Ahmadian d,e,f,g , Mehdi Salimi d,e,f,g ,

a School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, 110067, India
b Department of Orthopaedics and Traumatology, Faculty of Medicine, Universiti Kebangsaan Malaysia(UKM), Kuala Lumpur, Malaysia
c Department of Econometrics and Business Statistics, Monash Business School, Monash University, Wellington Rd, Clayton, Victoria 3800, Australia
d School of Mathematical Sciences, College of Science and Technology, Wenzhou-Kean University, Wenzhou, China
e Institute of IR 4.0, The National University of Malaysia, 43600 UKM, Bangi, Selangor, Malaysia
f Department of Mathematics & Statistics, St. Francis Xavier University, Antigonish, NS, Canada

1. Introduction

Over the past few decades, the population is rapidly increasing all over the world. As people are paying more attention to higher education. Herein, people are becoming health conscious Haddara and Staaby (2018). In today’s fast-paced, everyone is getting busy in their personal life or maybe own business, or even they may have a shortage of time. The term telecare offers remote monitoring and provides care to patients independently in their homes who are elderly, disabled, or suffering from critical diseases. In the latter 2019 and starting of 2020, the number of cases of novel Corona Virus Disease 2019 (nCOVID-19) increased rapidly throughout the world. Herein, IoT-Health enables the more convenient ways to access remotely and efficiently the medical services for the patients, also provides health monitoring by the doctors, physicians, and nurses over the Internet. However, security and privacy are considered key concerns in RFID-based IoT-health systems due to wireless communication over the channel. There could be huge risks of leakage of the patient’s sensitive information, medical data, privacy of the patients, and so forth. To overcome these shortcomings, we have put forward a secure and reliable RFID authentication protocol using Digital Schnorr Cryptosystem for IoT-Health in COVID-19 patients care named SR2-AP-DSC. Compared with the similar existing protocols, the security analysis followed by the performance evaluation of our proposed protocol demonstrates the minimal computation overheads and also provides resistance to various well-known security attacks. The AVISPA and Scyther simulation results confirm that the proposed protocol is safe under active and passive attacks. The overall analysis shows that the SR2AP-DSC is relatively superior to the other similar existing protocols.
assistance to patients without any risk Chauhan et al. (2020); Moazzami et al. (2020); Vidal-Alaball et al. (2020).

More specifically, Telecare Medical Information System (TMIS) plays a crucial role for those patients or people who are suffering from a disability or any other critical disease and cannot attend the hospital for their routine health check-ups. In order to provide convenience for such people, the facility of TMIS has come into the existence. TMIS is a telecare system based on a group of different technologies that provides effective and online medical and healthcare services to help the patients to avail treatment facilities in minimum time at a minimum cost over the Internet. TMIS has numerous applications such as e-healthcare, distant nursing, and home monitoring facility, etc. The TMIS consists of two typical entities one is a user (or patient) and another is a medical server.

The main feature of IoT-Health is that the patients (or users) can securely exchange their important medical data or records with the concerned doctor(s) through a reliable medical server over a wireless communication channel. The medical server handles the responsibility for ensuring the accessibility of healthcare services for the registered users via mobile network or the Internet. It also stores the information associated with the registered users. This information could be the patient’s name, sex, age, mobile number, Electronic Medical Record (EMR), credit card details, etc Guo et al. (2020); Xie et al. (2020). The disclosure of any of these records can be a serious threat to the user’s privacy. Considering the various risks to user’s safety and privacy, RFID technology can be successfully applied to IoT-Health to resolve these issues. The RFID-based solutions can be deployed effectively in IoT-Health for online accessing the healthcare services with several advantages including effective continuation, reducing administrative cost, accurate record-keeping, preventive care, and improving quality of care, etc. However, RFID is gaining more popularity and also has the rapid development in medically oriented services as non-contact, wireless, and unique identification technologies.

The RFID system has three key components: tags, readers, and a backend server. The RFID tags are affixed to identifying objects or things Shariq and Singh (2021). The reader reads and writes data on the tags and the server stores the sensitive information associated with the patients that are sent by the readers. In IoT-Health, the medical data and patient information can be transferred through radio frequency signals over a wireless environment, so there may arise some critical problems of its security and privacy. In the past few decades, several RFID-based authentication protocols have been introduced and have implemented in the public networks by many researchers. These protocols could not resolve the issues related to patient security and privacy over a wireless healthcare environment by employing some different cryptographic primitives-based solutions that have been discussed in the literature.

1.1. Motivation

During the COVID-19 treatment, doctors prefer to diagnose the patients remotely over the Internet. It is well-known that the information of the patients is transmitted over the Internet through an insecure or wireless channel. However, security and privacy are the primary concerns in RFID systems where an adversary can intercept, modify, or even tamper with the sensitive information of the patients. In addition, s/he can also control medical equipment which may cause damage to the medical records, medical equipment, infrastructure, and, facilities, etc. Since the patients access IoT-Health services over the public channel, several critical issues may arise including data confidentiality, integrity, availability, patient’s authenticity, and patient’s privacy protection. To meet these privacy requirements, we need to construct a secure and reliable RFID authentication protocol for IoT-Health sector.

1.2. Key objectives

The followings are the key objectives of our proposed protocol:

1. To accomplish mutual authentication between tag and backend server.
2. To preserve the tag anonymity feature.
3. To achieve the tag location privacy and tag untraceability property.
4. To defeat various known attacks including replay and de-synchronization attacks, among others.
5. To reduce computation overhead of tags.
6. To meet all the necessary security requirements including scalability and availability.

1.3. Our contribution

First, we have reviewed the existing literature includes various security-privacy and performance related issues in the previous protocols. Before introducing the protocol, we highlight key points of the proposed protocol are listed as:

1. We design a secure and reliable RFID authentication protocol using Digital Schnorr Cryptosystem for IoT-Health in COVID-19 patients care.
2. To overcome the computational complexity of tags and server, we utilize only one-way hash $h(\cdot)$, simple bitwise XOR $(\oplus)$, and Digital Schnorr Cryptosystem for improving the security functionalities of our RFID-based TMIS system.
3. The rigorous formal-informal analysis shows that the $SR^2$AP-DSC accomplishes mutual authentication feature as well as resists several known attacks including tag location tracking, server spoofing, tag impersonation, replay, and tag tracking attacks, etc.
4. The performance comparison has been carried out which demonstrates that our protocol significantly reduces computational overhead of tags.

1.4. Paper organization

The remaining part of this paper is structured as follows. A brief description of existing related work is described in Section 2. Next, the proposed RFID authentication protocol for COVID-19 patients care is given in Section 3. Further, Section 4 and Section 5 evaluates the formal and informal security analysis, followed by the performance evaluation of our protocol. In Section 6, AVISPA and Scyther simulation tools have been used for formal verification of the proposed protocol. Finally, Section 7 concludes the proposed work.

2. Related work

RFID-based authentication for TMIS domain has become one of the hot topics of research in IoT healthcare applications Dharminder et al. (2020). In RFID-enabled TMIS systems, the main adoption of RFID technology in healthcare or medical is to ensure secure access for patients, infant protection, patient’s health status tracking, reliability to patient’s medical data, medication safety, the management of patient’s records, managing patient location, equipment, employee, location tracking of medical assets, sensitive information of the patients, and so forth. Several RFID schemes have been introduced in the recent past for safeguarding the RFID system in healthcare sector which provides protection from various malicious attacks. Therefore, we discuss some of the existing RFID-based TMIS schemes in healthcare with their techniques, advantages, and pitfalls to address above mentioned issues.

In Kaul and Awasthi (2013), a secure and efficient lightweight RFID scheme is presented to enhance medication safety for the patients in health care domain. It reduces medication errors which may cause substantial harm to patients’ safety. The scheme utilizes PRNG function, simple XOR, and one-way hash $h(\cdot)$ due to very limited communicational and computational capacity of tags and low memory space.

Chou (2014) introduced an ECC-based RFID authentication system to overcome several security flaws such as tag tracking, impersonation, and
de-synchronization attacks. Although, the author reported that their scheme resists known security requirements including replay, impersonation, and Man-In-The-Middle (MITM) attacks. Later on, Zhang and Qi (2014) report that the protocol Chou (2014) does not provide resistance against forward security, also is insecure against impersonation attacks.

In Liao and Hsiao (2014), a secure RFID protocol based on ECC through ID-verifier integrated transfer protocol is presented for the healthcare environment. The protocol guarantees various security features. However, Lio and Hsiao’s protocol does not provide any security requirements because an adversary revealed the tag’s secret key reported by Zhao (2014). Thus, their protocol shows that the vulnerability against impersonation attacks. Subsequently, Peeters and Hermans (2013) show the vulnerabilities against the tag cloning, tag’s location privacy, and tag masquerades attacks on Liao and Hsiao (2014).

Zhao (2014) put forward a secure RFID scheme using ECC. However, Farash et al. (2016) conclude that the schemes Zhang and Qi (2014) and Zhao (2014) cannot preserve property of forward secrecy. Subsequently Farash et al. (2016) put forward a provable secure RFID scheme using ECC for healthcare environment.

In Srivastava et al. (2015), a secure and mutual RFID-based tag authentication scheme is presented for the medical environment. The scheme uses one-way hash operation, simple bitwise exclusive-OR operation, and a synchronized shared secret. They guarantee that their scheme is resistant to known security attacks including replay, de-synchronization, forgery and traceability attacks. Subsequently, Li et al. (2015) report that scheme Srivastava et al. (2015) shows a serious security issue, where an adversary can deploy stolen/lost RFID reader for connecting with the medical server that stores the sensitive data of the tag attached objects. Further, they show that the scheme Srivastava et al. (2015) lacks of mutual authentication and is insecure against the reader stolen/lost attack.

In Jin et al. (2016), an RFID scheme is put forward to enhance the medication safety of patients in the medication system. The scheme utilizes Elliptic Curve Cryptography (ECC) to achieve various required security features and it can withstand various well-known security attacks including Denial-of-Service (DoS), tag impersonation, replay, location tracking, server spoofing, cloning, MITM, and de-synchronization attacks. However, prakash Pokala et al. (2016) report that the scheme Jin et al. (2016) do not preserve tag identity property. Also, the scheme is susceptible to tag impersonation attack. Later on, Li et al. (2015) present an enhancement of the scheme Srivastava et al. (2015) to overcome the security weaknesses and improving efficiency of an RFID system. The scheme utilizes an identifier of the reader, a secret value of the reader, lightweight hash operation, and simple bitwise exclusive-OR. The authors show that their scheme resists reader stolen/lost, de-synchronization, and replay attacks as well as provision of mutual authentication. However, Zhou et al. (2019) conclude that the scheme Li et al. (2015) cannot be applicable on mobile RFID environments because of the secure communication channel. Therefore, the scheme Li et al. (2015) suffers from data integrity problems and replay attack in mobile RFID environment as well as vulnerable to traceability, replay, and de-synchronization attacks.

To fix the shortcomings of the scheme Li et al. (2015), Zhou et al. (2019) propose an improved more secure and efficient version of the scheme Li et al. (2015) particularly RFID authentication scheme suitable for TMIS. The scheme utilizes one-ways hash function, bitwise exclusive-OR operation, and quadratic residue theory. Besides, the tag generates a random number for key updating operation to ensure strong forward untraceability. The major drawback of the scheme is, that it spent more time in execution.

To fix the shortcomings of the scheme Li et al. (2015), Bensalah et al. (2017) put forward an improvement, which provides stronger security with higher efficiency. To fix the shortcomings of Chen and Chou (2015), Shen et al. (2017) presented an efficient RFID authentication using ECC for IoT environment. However, Afroz et al. (2019) note that Shen et al.’s scheme suffers from DoS, cloning, and location tracking attacks.

Alamr et al. (2018) introduce a secure RFID scheme for IoT environment using ECC. In order to generate a shared key, they use the elliptic curve Diffie-Hellman protocol for encrypting transmitted messages. The authors show that their scheme provides resistance against known attacks including impersonation, replay, and MITM attacks. Also, their scheme is susceptible to scalability issues and cannot be suitable for IoT environment. However, Tu et al. (2020) note that the scheme Alamr et al. (2018) is susceptible to full disclosure and de-synchronization attacks.

Noori et al. (2020) put forward an efficient, and scalable RFID authentication using ECC for IoT healthcare systems. They show that their scheme provides resistance against known security attacks including insider, forgery, masquerade, replay, and MITM attacks. However, their scheme does not take advantage of provable security.

Salem and Amin (2020) put forward an RFID authentication scheme for enhancing patients medication safety in TMIS systems. The scheme uses El-Gamal cryptosystem for privacy-preserving property. The scheme can withstand several known cryptographic security attacks including DoS, replay, location tracking, de-synchronization, and tag impersonation attacks.

Safkhani and Vasilakos (2019) presented cryptanalysis of a new authentication scheme in mobile RFID that was proposed by Zheng et al. (2018). They pointed out that their scheme can withstand replay, impersonation, and de-synchronization attacks. Thereafter, they proposed a new scheme which is secure against possible attacks.

Chen et al. (2020) presented cryptanalysis of two RFID authentication protocols that were proposed by Fan et al. (2018) and Bensalah et al. (2017). They showed that their protocols were vulnerable under reader impersonation, tracking, and tag impersonation attacks. Later on, they proposed an enhanced protocol called TMIS-+ for RFID-based TMIS systems.

Sharif et al. (2021) proposed an lightweight RFID-enabled authentication protocol using permutation operation called URASP. The protocol utilizes simple bitwise XOR, circular left rotation Rot(…), and permutation Per(…) operations on the passive RFID tags. The protocol can withstand various security attacks and also preserves information privacy and tags untraceability properties under Juels and Weis privacy model. The verification of the protocol was done by using Scyther tool and BAN logic.

Xiao et al. (2021) proposed a lightweight access control authentication protocol for TMIS. The protocol uses ECC and Physically Unclonable Function (PUF) to establish a secure authentication between tag and server. The PUF generates the key information which reduces the cost of algorithm and also prevents information leakage. The ProVerif tool shows that the protocol is safe against major security attacks.

### Table 1
Notations and definitions.

| Notation | Definition |
|----------|------------|
| $p, q$   | Two large prime numbers |
| $ID_i$   | Unique identification number of $i$th tags |
| $k$      | Random integer generated at reader |
| $r_1, r_2$ | Random integers generated at tag |
| $a$      | Secret key stored at the reader-server |
| $a_i$    | Random number shared between tag and reader |
| $h_i(\cdot)$ | One-way cryptographic hash function |
| $\|$    | Concatenation operation |
| $\oplus$ | XOR operation |
3. Proposed RFID authentication protocol

The proposed protocol comprises of the initialization and authentication phases respectively. Table 1 puts the useful notations with their respective definitions.

3.1. System model

The proposed SR²AP-DSC protocol uses a system model for IoT-enabled healthcare in COVID-19 scenario as shown in Fig. 1. The figure shows a smart healthcare system where COVID-19 infected patients are being diagnosed. The system consists of RFID tags, readers, and a trusted medical server. Moreover, the tags are equipped with the patients as well as medical equipment, who is admitted in ICU, isolation wards, or even at their homes. The reader reads the information of the tags attached to the patients and sends it to the server for storing and further processing.

3.2. Adversary model

In smart healthcare systems, RFID tags store real-time information such as RT-PCR, chest X-ray, CT-scan, and other diagnosis reports.
Table 2
Comparison of informal security analysis among various RFID-based authentication protocols.

| Protocol        | Chou (2014) | Zhang and Qi (2014) | Liao and Hsiao (2014) | Zhao (2014) | Farash et al. (2016) | Jin et al. (2016) | Shen et al. (2017) | Alamr et al. (2018) | Noori et al. (2020) | Salem and Amin (2020) | Proposed |
|-----------------|-------------|---------------------|-----------------------|-------------|---------------------|------------------|-------------------|------------------|------------------|-------------------|----------|
| SF1             | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF2             | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF3             | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF4             | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF5             | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF6             | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF7             | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF8             | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF9             | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF10            | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF11            | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF12            | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF13            | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF14            | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF15            | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF16            | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF17            | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |
| SF18            | ✓           | ✓                   | ✓                     | ✓           | ✓                   | ✓                | ✓                 | ✓                | ✓                | ✓                 | ✓        |

Acronyms: SF1: Confidentiality; SF2: Mutual authentication; SF3: Tag’s privacy; SF4: Availability; SF5: Backward and forward security; SF6: Resists tag impersonation attack; SF7: Resists server spoofing attack; SF8: Resists replay attack; SF9: Resists DoS attack; SF10: Resists location tracking attack; SF11: Resists cloning attack; SF12: Resists De-synchronization attack; SF13: Resists man-in-the-middle attack; SF14: Provable security; SF15: Easiness of secure implementation; SF16: Scyther tool simulation; SF17: AVISPA tool simulation; SF18: Secure under any security model; N/A: Not applicable.

![Fig. 3. Comparison of total computational complexity.](image)

Table 3
Comparison of computational complexity among various RFID-based authentication protocols.

| Protocol          | Entity | Computational Complexity | Total |
|-------------------|--------|--------------------------|-------|
| Chou (2014)       | Tag    | 2T_{enc} + 2T_{mac} + 2T_{es} | ≈ 22.0876 |
|                   |        | ✓                        |       |
| Zhang and Qi (2014)| Tag    | 2T_{enc} + 2T_{mac} + T_{es} + 2M_{i} | ≈ 22.0969 |
|                   |        | ✓                        |       |
| Liao and Hsiao (2014)| Tag   | 5T_{mac}                 | ≈ 73.5 |
|                   |        | ✓                        |       |
| Zhao (2014)       | Tag    | 3T_{enc} + 5T_{mac}      | ≈ 73.554 |
|                   |        | ✓                        |       |
| Farash et al. (2016)| Tag  | 2T_{enc} + 1T_{mac} + 2T_{es} | ≈ 21.9796 |
|                   |        | ✓                        |       |
| Jin et al. (2016) | Tag    | 2T_{enc} + 1T_{mac} + 1T_{es} | ≈ 29.4316 |
|                   |        | ✓                        |       |
| Shen et al. (2017)| Tag    | 4T_{mac} + 3T_{mac}      | ≈ 44.1208 |
|                   |        | ✓                        |       |
| Alamr et al. (2018)| Tag  | 1T_{mac} + 4T_{es}       | ≈ 66.168 |
|                   |        | ✓                        |       |
| Noori et al. (2020)| Tag  | 2T_{enc} + 2T_{es} + 2T_{mac} | ≈ 14.7196 |
|                   |        | ✓                        |       |
| Salem and Amin (2020)| Tag  | 2T_{enc} + 1T_{mac} + 1T_{es} + 1M_{i} | ≈ 5.5216 |
|                   |        | ✓                        |       |
| Proposed          | Tag    | 3T_{mac} + 1T_{mac} + 1M_{i} | ≈ 1.8624 |
|                   |        | ✓                        |       |

3.3. Valid assumption considered

The proposed protocol operates under the following necessary assumptions as listed:

1. The communication channel is considered to be secure or wired between reader $\mathcal{R}$ and reader-server unit $\mathcal{S}$.
2. The communication channel is considered to be insecure or wireless between tag $\mathcal{F}$ and reader $\mathcal{R}$ i.e. susceptible to all possible attacks.

associated with the infected patients. The widely-accepted Dolev-Yao (DY) adversary model Dolev and Yao (1983) is taken into account by considering the following powers of an adversary $\mathcal{A}$:

1. $\mathcal{A}$ has full control over the communication links.
2. $\mathcal{A}$ can listen to the communicated messages between participating entities namely, tags and readers.
3. $\mathcal{A}$ can eavesdrop, alter, tamper, add, delete, modify, and even forge the messages during communication.
4. $\mathcal{A}$ can extract sensitive information by capturing the messages over the insecure communication channel.
5. $\mathcal{A}$ can reveal all secret information and also can impersonate a legitimate tag or a reader.
3.4. Setup phase

The server and tags produce some systems parameters which are as follows:

1. \( p, q \) are two large prime numbers with a length of 1024-bits and 160-bits respectively.
2. \( g \) is a generator with \( 1 < g < p - 1 \).
3. \( a \) is a shared secret key with \( 0 < a < q \).
4. \( v \) is a public key, where \( v = g^a \mod p \).
5. Choose a random number \( a_i \) which is shared between tag and reader.
6. The random numbers are unique for each tag. So, the reader stores securely all the random numbers for each tag at the reader-server unit.
7. \( ID_i \) is an \( i \)th tag identifier with a length of 160-bits stored in the \( i \)th tag’s memory.
8. The identifier of each tag and its associated information is also stored in the database of backend server.

3.5. Authentication phase

Fig. 3 illustrates the steps of the authentication process are given as:

Step 1: \( \mathcal{S} \rightarrow \mathcal{S} : \{C_1\} \).

The reader-server unit randomly chooses an integer \( k \in \mathbb{Z}^+ \), sets \( C_1 = k \), and transmits it to the tag.

Step 2: \( \mathcal{S} \rightarrow \mathcal{T} : \{x_1, x_2, Auth_i\} \).

Upon receiving \( C_1 \), the tag generates randomly two integers \( (r_1, r_2) \in \mathbb{Z}^+ \), performs the following computations and subsequently sends \( \{x_1, x_2, Auth_i\} \) to the server.

- Computes \( x_1 = g^{r_1} \mod p \).
- Computes \( x_2 = (r_2 v^{r_1}) \mod p \).
- Computes \( e = h(r_2 || x_2 || x_1) \).
- Computes \( y = (r_2 + ae) \mod q \).
- Computes \( Auth_i = ID_i \oplus h(r_2, C_1, e, y) \).

Step 3: \( \mathcal{T} \rightarrow \mathcal{S} : \{C_2\} \).

After receiving \( x_1, x_2 \) and \( Auth_i \), the server searches tag’s identifier \( ID_i \) in the database. If it is found, the server authenticates the tag; otherwise, it detects malicious behavior. After that, the server performs the following computations and subsequently sends \( C_2 \) to the tag.

- Computes \( S_1 = x_2^a \mod p \).
- Computes \( S_2 = S_1 \mod p \).
- Computes \( r_2 = (x_2 S_2^{-1}) \mod p \).
- Computes \( e' = h(r_2 || x_2 || x_1) \).
- Computes \( y' = (r_2 + ae') \mod q \).
- Extracts \( ID_i = Auth_i \oplus h(r_2, C_1, e, y) \).
- Computes \( C_2 = h(ID_i, r_2, C_1, e, y, Auth_i) \).

Step 4 Verification at Tag.

Upon receiving \( C_2 \), the tag computes the value of \( C_2 \) and checks
whether $C'_2 = C_2$, if so, the tag authenticates the server.

- Computes $C'_2 = h(ID_i, r_2, C_1, e_y, Auth_i)$.
- Verify $C'_2 = C_2$.

4. Security analysis

This section mainly focuses on the formal and informal security and privacy requirements. The formal security is carried out under Ouafi and Phan’s privacy model Ouafi and Phan (2008). Compared to similar existing protocols, the informal security comparison of our protocol achieves strong protection from various known security attacks including location tracking, tag traceability, tag impersonation, replay, de-synchronization attacks, etc., and also meet various privacy features are shown in Table 2.

4.1. Formal security model

The formal Ouafi and Phan’s security model Ouafi and Phan (2008) consists of a protocol party $T \in Tags$ or $R \in Readers$ which interacts in a protocol authentication session. It will interact until the end of the protocol execution session till then each party outputs Accept during the execution with the correct parties. This model defines the capabilities of an adversary $\mathcal{A}$ who can perform eavesdropping on the radio communication link among all protocol parties (i.e. tag $T$ and reader $R$). Moreover, $\mathcal{A}$ can launch different types of active or passive attacks. The adversary has capability to issue oracle queries in polynomial time as follows:

**Execute($T, R, i$):** The passive attacks can be modelled by this query. In this regard, the adversary $\mathcal{A}$ can perform eavesdropping to an honest protocol execution session $i$ between $T$ and $R$. In addition, $\mathcal{A}$ can obtain all shared secret messages between $T$ and $R$.

**Send($U, V, m, i$):** The active attacks can be modelled by this query. In this regard, $\mathcal{A}$ can impersonate some reader $U \in Readers$ (respective tag $U \in Tags$) in a protocol execution session $i$ and subsequently transmits a message $m$ of its choice to some tag $V \in Tags$ (respective reader $V \in Readers$).

**Query($T, m_1, m_2$):** This query allows the abilities of an adversary $\mathcal{A}$ for tag investigation. In this context, $\mathcal{A}$ sends a message $m_1$ to the tag $T$ and subsequently receives a message $m_2$ from the tag $T$.

**Block($\mathcal{A}$):** This query allows the abilities of an adversary to launch a Denial of Service attack, where $\mathcal{A}$ can block a part of the protocol.

![Fig. 5. HLPSL specification of server role.](image-url)
role session(Ti,S : agent,
H: hash_func,
ADD: hash_func)
def=
local SI,SJ,RI,RJ: channel(dy)
composition
tag(Ti,S,H,ADD,SI,RI)
\server(Ti,S,H,ADD,SI,RI)
end role
role environment()
def=
const ti,s : agent,
h: hash_func,
add: hash_func,
g,a,p,q,y,idi,k,x1,x2,s1,s2,
x1,x2,e,y,authi,s1,s2,c1,c2: text,
tag_server_r1,tag_server_r2,server_tag_idi,tag_server_idi,
subs1,subs2,subs3: protocol_id
intruder_knowledge={ti,s,h,add,x1,x2,e,y,authi,c1,c2}
composition
\session(ti,s,h,add)
\session(ti,s,h,add)
end role
goal
secrecy_of_subs1
secrecy_of_subs2
secrecy_of_subs3
authentication_on_tag_server_r1
authentication_on_tag_server_r2
authentication_on_server_tag_idi
authentication_on_tag_server_idi
end goal
environment()

---

Fig. 6. HLPSL specification of session, goal and environment role.

---

Fig. 7. OFMC and CL-AtSe back-ends simulation results.
Moreover, s/he can break the synchronization between $T$ and $S$.

**Corrupt** $(T, K)$: This query gives permission to an adversary $A$ to obtain the tag’s shared secret key ($K$) stored in tag’s internal memory space.

**Test** $(T_0, T_1, i)$: This query does not correspond to any real-world event or any of $A$’s abilities. It defines the indistinguishability-based notion of untraceable privacy ($UPriv$). In this way, $A$ runs $G$ to issue the following oracle queries:

**Learning Phase:** $A$ may send Execute, Send, and Corrupt oracles at his/her will and can interact with the randomly chosen tags $T_0, T_1$ and reader-server unit $S$.

**Challenging Phase:** $A$ randomly selects two tags $T_0, T_1$ and sends a Test $(T_0, T_1, i)$ query to the challenger $C$. Thereafter, the adversary $A$ computes a tag $T_b \in \{T_0, T_1\}$ by calling Execute and Send queries, where $b$ is a randomly chosen bit $b \in \{0, 1\}$ by the challenger $C$.

**Guessing Phase:** Eventually, $A$ terminates the simulation of game $G$ and produces outputs $b'$, which denotes the adversary’s guess of value $b$. Therefore, $A$'s success in winning the game $G$ which leads to breaking the notion of untraceability privacy and can be defined as $A$’s advantage to distinguish whether $A$ received $T_0$ or $T_1$, i.e. s/he guesses the correct value of $b$. The $A$’s advantage can be represented as $Adv_{UPriv}(A) = \| \mathbb{P}_b[b' = b] - \frac{1}{2} \|$, where $k$ represents security parameter.

---

**Definition 1 (Partnership & Session Completion).** A tag instance $T_i$ and a reader instance $R_j$ are said to be partners if, and only if, both $T_i$ and $R_j$ have outputs $\text{Accept}(T_i)$ and $\text{Accept}(R_j)$ respectively, which denotes the protocol completion session.

**Definition 2 (Freshness).** A protocol party instance is said to be fresh at the end of the protocol execution session if, and only if,

1. it has given output $\text{Accept}$ with or without a partner instance,
2. both the instances $T_i$ and $R_j$ and its partner instance (if such a partner exists) has not been sent a Corrupt query.

**Definition 3 (Untraceable Privacy ($UPriv$)).** It defines the game $G$ played between a collection of tag and reader instances and a malicious adversary $A$. In this way, $A$ runs $G$ to issue the following oracle queries:

**Learning Phase:** $A$ may send Execute, Send, and Corrupt oracles at his/her will and can interact with the randomly chosen tags $T_0, T_1$ and reader-server unit $S$.

**Challenging Phase:** $A$ randomly selects two tags $T_0, T_1$ and sends a Test $(T_0, T_1, i)$ query to the challenger $C$. Thereafter, the adversary $A$ computes a tag $T_b \in \{T_0, T_1\}$ by calling Execute and Send queries, where $b$ is a randomly chosen bit $b \in \{0, 1\}$ by the challenger $C$.

**Guessing Phase:** Eventually, $A$ terminates the simulation of game $G$ and produces outputs $b'$, which denotes the adversary’s guess of value $b$. Therefore, $A$’s success in winning the game $G$ which leads to breaking the notion of untraceability privacy and can be defined as $A$’s advantage to distinguish whether $A$ received $T_0$ or $T_1$, i.e. s/he guesses the correct value of $b$. The $A$’s advantage can be represented as $Adv_{UPriv}(A) = \| \mathbb{P}_b[b' = b] - \frac{1}{2} \|$, where $k$ represents security parameter.
role ReaderServer
{
    const C1,C2,x1,x2,S1,S2,e,y,Authi,g,v,a,ai,IDi;
    fresh k: Nonce;
    macro C1=k;
    send_11(ReaderServer,Tag,C1);
    recv_12(Tag,ReaderServer,{x1,x2,Authi}pk(ReaderServer));
    macro S1=Exp(x1,a);
    macro S2=S1;
    macro r2=MUL(x2,Inverse(S2));
    macro e'=Hash(r2,x2,x1);
    macro y'=ADD(r2,MUL((ai,e')));
    macro IDi=XOR(Authi,Hash(Concatenate(r2,C1,e,y)));
    macro C2= Hash(Concatenate(IDi,r2,C1,e',y',Authi));
    send_13(ReaderServer,Tag,C2);
    claim(ReaderServer, Secret, IDi);
    claim(ReaderServer, Secret, a);
    claim(ReaderServer, Secret, k);
    claim(ReaderServer, Niagree);
    claim(ReaderServer, Nisynch);
    claim(ReaderServer, Alive);
    claim(ReaderServer, Weakagree);
}

Fig. 9. SPDL specification of reader and server.

Fig. 10. Scyther simulation results.
4.2. Formal security analysis

Theorem 1. SR^TP-DSC cannot perform any traceability attacks.

Proof. In SR^TP-DSC, the tag updates its random number a in each authentication session. Besides, the tag identifier IDi changes in each authentication session. Thus, \( \varnothing \) cannot perform any traceability attack under these phases which are given below:

Learning phase: \( \varnothing \) sends an Execute \((J, T, i)\) query and obtains the parameters \{Auth^S_{i}, ID^T_{i}\} in the ith round.

Challenging phase: \( \varnothing \) selects two fresh tags \( J_0, J_1 \) and sends a Test \((J_0, J_1, i+1)\) query. Next, \( \forall \) is given a tag \( T_{b} \in \{J_0, J_1\} \), where \( b \) is randomly chosen bit i.e. \( b \in \{0, 1\} \). After that, \( \forall \) sends an Execute \((J, S_{b}, i+1)\) query and obtains the parameters \{Auth^S_{i+1}, ID^T_{i+1}\}.

Guessing phase: The random number \( a \) of a tag updated in the Learning phase. Therefore, the parameters Auth^S_{i+1} and Auth^S_{i} for two subsequent sessions, \( i, i+1 \) can be computed as: Auth^S_{i} = ID^T_{i} \oplus h(r_2, c_1, e, y) and Auth^S_{i+1} = ID^T_{i+1} \oplus h(r_2, c_1, e, y). Since Auth^S_{i} \neq Auth^S_{i+1} and ID^T_{i} \neq ID^T_{i+1}, \) therefore the adversary is required to perform random guessing. In this regard, \( \forall \)'s advantage to distinguish whether \( \forall \) received \( J_0 \) or \( J_1 \), can be represented as \( Adv_{\forall}^{Guess}(k) = \| \mathbb{P}[b' = b] - \frac{1}{2} \| \mathbb{E}(k) \).

Theorem 2. SR^TP-DSC achieves mutual authentication.

Proof. Consider, the reader and server as a single unit \( J \). Moreover, there is a game modelled between \( \forall \) and \( \forall \), where \( \forall \) may try to authenticate his/herself as a legitimate tag. The following steps are executed by game \( \forall \) as given below:

- **Step 1** The challenger \( \forall \) chooses legitimate tag \( J \) and server \( J \).
- **Step 2** The adversary \( \forall \) calls oracle queries such as Execute, Send, and Query on a server \( J \) and a tag \( J \) for a polynomial time.
- **Step 3** After finishing Step 2, the adversary \( \forall \) notifies the challenger \( \forall \).
- **Step 4** The adversary \( \forall \) sends the Send query to impersonate a legitimate tag.
- **Step 5** If the adversary \( \forall \) authenticates his/herself as a legitimate tag, it means that \( \forall \) wins the game.

Now, if the backend server \( J \) interrogates into a session then adversary \( \forall \) will respond in order to prove his/her legitimacy. For that, \( \forall \) requires to send a legitimate identifier of the ith tag IDi and also \( \forall \) requires to generate a legitimate response Authi = IDi \oplus h(r_2, c_1, e, y). Here, \( \forall \) must able to know the random number a stored at the reader-server unit. However, the adversary \( \forall \) cannot disclose the value a, which means that \( \forall \) is unable to impersonate as a legitimate tag. Despite that, \( \forall \) requires to send a Query oracle in Step 4 and also \( \forall \) requires to sends a legitimate response C2 = h(ID2, r_2, c_1, e, y, Authi) to be authenticated as a legitimate server. Finally, \( \forall \) cannot even infer a, hence \( \forall \) cannot impersonate as a legitimate reader-server unit \( J \).

4.3. Informal security analysis

1) Confidentiality (SF_i): In SR^TP-DSC, the ith tag identifier IDi is embedded in the message Authi = IDi \oplus h(r_2, c_1, e, y). The adversary can compute the transmitted messages \{C1\} and \{x_i, x_2, Authi\}, public key v of the reader-server unit. However, the value of random number r_2 cannot be revealed. So, the adversary cannot compute the ith tag’s identifier IDi. Hence, SR^TP-DSC satisfies SF_i property.

2) Mutual authentication (SF_{ij}): In SR^TP-DSC, the adversary cannot create a legitimate messages \{x_i, x_2, Authi\} without knowing the ith tag’s identifier IDi, and the random numbers r_1 and r_2. However, the reader-server unit computes the message ID2 = Authi \oplus h(r_2, c_1, e, y) and verify the tag authenticity by searching for tag’s identifier IDi in the backend server.

Besides, \( \forall \) cannot compute the message C2 = h(ID2, r_2, c_1, e, y, Authi), since the parameters ID2 and r_2 are not known. Hence, SR^TP-DSC achieves SF_{ij}.

3) Tag privacy (SF_{II}): In SR^TP-DSC, the ith tag identifier IDi is embedded in the message Authi = IDi \oplus h(r_2, c_1, e, y). Moreover, the tag, and server generates fresh random numbers k, r_1, and r_2 in each session. In this way, the adversary cannot point out the exact tag’s location. Hence, SR^TP-DSC preserves a SF_{II} property.

4) Availability (SF_{I}): In SR^TP-DSC, the adversary cannot obtain the ith tag identifier IDi during the protocol execution session and also not required to change the value of IDi after the protocol execution phase. Hence, SR^TP-DSC preserves SF_{I} property.

5) Backward and forward traceability (SF_{III}): In SR^TP-DSC, the adversary cannot track the position of the tag, even if \( \forall \) disclose the identifier of the tag. Suppose that \( \forall \) can disclose the ith tag identifier ID_i and can intercept the transmitted messages \{C1\} and \{x_1, x_2, Authi\} between the tag and the server, where \( x_1 = g^v \mod p \), \( x_2 = (r_2 \cdot v^v) \mod p \), and Authi = IDi \oplus h(r_2, c_1, e, y). However, the adversary cannot obtain the secret key of the server, i.e. a and the values of random numbers r_1 and r_2. Hence the adversary cannot be sure that the messages that are being exchanged belong to the tag with identifier IDi. Thus, the proposed protocol preserves SF_{III} properties.

6) Rests tag impersonation attack (SF_{IV}): In SR^TP-DSC, consider an adversary eavesdrops on a session and tries to perform a tag impersonation to the server. The adversary intercepts the message \{C1\} which is transmitted by the reader-server unit and will try to produce a legitimate message \{x_1, x_2, Authi\}. However, the adversary cannot produce the legitimate messages \{x_1, x_2, Authi\} without knowing the ith tag’s identifier IDi and the random numbers r_1 and r_2. Hence, SR^TP-DSC is safe under tag impersonation attack.

7) Rests replay attack (SF_{V}): Consider that an adversary takes away the response messages C1 = k and C2 = h(ID2, r_2, c_1, e, y, Authi), and replay them to the tag. The tap can able to identify the existence of the adversary by verifying whether C1 \cdot \mathbb{Z} = h(ID2, r_2, c_1, e, y, Authi). The tag generates fresh random numbers r_1 and r_2 in each new authentication session. Moreover, the adversary intercepts the messages \{x_1, x_2, Authi\} and replay them to the server. The reader-server unit can observe the attack by verifying the authenticity of Authi. Therefore, the server generates a fresh random value k in each new authentication session. Hence, SR^TP-DSC is safe under replay attack.

8) Rests spoofing attack (SF_{VI}): In SR^TP-DSC, an adversary impersonate the server to the tag after generating a random value k \in \mathbb{Z}*, set C1 = k, and send response message C1 to tag. However, \( \forall \) cannot produce a legitimate response message C2 as \( \forall \) cannot disclose the secret key of the server i.e. a and cannot compute the ith tag’s identifier IDi. In this way, the adversary is unable to impersonate the server to the tag. Thus, SR^TP-DSC is safe under a server spoofing attack.

9) Rests denial-of-service attack (SF_{VII}): In SR^TP-DSC, the adversary tries to make it difficult for the target tag to receive the services. The adversary may try to disconnect the communication between tag and server by making them unrecognizable to each other, and can finally lead to disabling of the tag. In our protocol, the tag’s identifier IDi cannot be obtained by the adversary. Hence, it is not required to change the tag identifier IDi during the protocol execution. Therefore, SR^TP-DSC is secure against DoS attacks.

10) Rests location tracking attack (SF_{VIII}): In SR^TP-DSC, suppose that the adversary can track the exact tag’s location if \( \forall \) can find a connection of the intercepted message to its tag. The tag’s side transmitted messages should not be fixed because the adversary can trace the specific location of the tags. Now, we have considered that the adversary can reveal the identifier of the ith tag IDi and intercept the transmitted messages \{C1\} and \{x_1, x_2, Authi\} between the tag, and server. The tag, does not send the value of IDi directly in an authentication session. The
adversary cannot obtain the transmitted messages between tag and reader-server unit without knowing the values of a, r1, and r2. In this way, s/he cannot trace the location of the tag. Hence, SR2-AP-DSC is safe under tag location tracking attacks.

11) Resists cloning attack (SF11): In SR2-AP-DSC, the server generates a random secret (i.e. the identifier of the ith tag IDi such that IDi ∈ Z∗) for each tag. Now, suppose that an adversary can reveal only a tag’s identifier. On the contrary, s/he cannot reveal other identifiers of the tags, because of the lack of relation between the tags. Hence, SR2-AP-DSC is safe under cloning attack.

12) Resists de-synchronization attack (SF12): In SR2-AP-DSC, any adversary cannot obtain the value of the identifier of the ith tag IDi in a protocol execution session. Moreover, there is no need to synchronously change the value of IDi after our protocol execution phase. Hence, SR2-AP-DSC is safe under a de-synchronization attack.

13) Resists man-in-the-middle attack (SF13): It is not easy to launch man-in-the-middle attack even if the adversary knows response message {Cj}. Because if the adversary impersonates as a reader-server and receives the next valid messages {x1, x2, Authi}. In this regard, the adversary cannot reconstruc the next valid response message {Cj}, since s/he doesn’t know the parameters a, r1, and r2. Hence, SR2-AP-DSC is robust against MITM attacks.

5. Performance analysis

Here, the performance evaluation of our proposed protocol is carried out, and subsequently a comparison is made among several considerable RFID authentication protocols based on ECC for TMIS in terms of computational complexity of tags and server is shown in Fig. 3. As far as we know, our proposed protocol is first to use the concept of Digital Schnorr Cryptosystem to design, a secure and reliable RFID authentication protocol using DSC for IoT-Health in COVID-19 patients care that is more secure and robust on comparing with other existing ECC-based RFID protocols. More specifically, SR2-AP-DSC is highly efficient as well as it provides better performance in terms of computational cost is illustrated in Table 3. Therefore, these features make our protocol more preferable for a low-cost TMIS over other existing protocols.

In Salem and Amin (2020), the execution times for all cryptographic functions are as follows: Hash functions T hashes ≈ 0.0004 ms, modular multiplications Tmult ≈ 0.015 ms, modular exponentiation Texp ≈ 1.83 ms, addition over elliptic curve Tadd ≈ 0.009 ms, and scalar multiplications Tmult ≈ 7.35 ms. On the other hand, in the Multiplicative Inverse (MI), there is only a modular operation with some other operations. So, it can be ignored as it will take very less time to execute the operation. As compared to other functions, the computation overhead of some other lightweight primitives like concatenation, XOR, and comparison operators can be neglected from the calculation because of their insignificant impact.

6. Simulation and verification using AVISPA and scyther tool

6.1. SF17: AVISPA Tool verification

This section confirms that the proposed protocol has been formally verified using a widely accepted tool called AVISPA (Automated Validation of Internet Security Protocols and Application) Armando et al. (2005). The High Level Protocol Specification Language (HLPSL) is used to write the code or specification of the designed protocol. The HLPSL specification for basic roles namely, tag and reader-server unit of the proposed protocol is shown in Fig. 4 and Fig. 5, respectively. In addition, the composite roles namely, session, goal, and environment are also shown in Fig. 6. However, AVISPA tool contains four different back-ends such as OFMC, CL-AtSe, SATMC, and TA4SP. Fig. 7 shows that the obtained outcomes of our protocol are “SAFE” simulated on AVISPA with OFMC and CL-AtSe back-ends. Besides, the OFMC back-end produces outputs as “SAFE” after visiting 16 nodes with a depth of 4 plies in 0.00s parseTime and 0.50s searchTime respectively. Likewise, the CL-AtSe back-end simulation took 0.13s to confirm the protocol as “SAFE”. Due to unsupported bitwise XOR operations on SATMC and TA4SP back-ends, the results have shown inconclusive on them. Hence, our protocol is robust and secure under active and passive attacks including MITM and replay attacks.

6.2. SF18: Scyther tool verification

Scyther has become a prominent and widely accepted tool used for checking the correctness of security protocols Cremers (2008). The adversary model used in Scyther works under Dolev-Yao model, which means that the adversary can eavesdrop, tamper, delete, or modify messages on the communication channel in Scyther and can learn from the messages. In order to simulate the proposed protocol, we use Linux operating system with Ubuntu version 20.04, an i7 processor of 3.60 GHz, and 12.0 GB of RAM. The Scyther uses Security Protocol Description Language (SPDL) to write the code or specification of the designed protocol. Moreover, a set of roles are designed by a sequence of events such as declarations, send, receive, and claim events. The event match is used for logical comparison and checks the equality of two identical messages. The protocol specification defines a sequence of roles of tag as shown in Fig. 8, reader and server as a single unit as shown in Fig. 9. The Scyther verification result is shown in Fig. 10, which indicates that there is no attack possible within bounds. Thus, the proposed protocol is safe under all known security attacks.

7. Conclusion

This paper puts forward a secure and reliable RFID authentication protocol using Digital Schnorr Cryptosystem for IoT-Health in COVID-19 patient care. On comparing with the similar existing protocols, the formal and informal security analysis followed by the performance evaluation of our proposed protocol demonstrates that it has low computational overhead as well as provides resistance to various well-known security attacks. Based on the AVISPA and Scyther simulation results, the proposed protocol strongly resists various known active and passive security attacks including MITM and replay attacks. Furthermore, the proposed protocol provides better security and performance with higher efficiency and is well-suited for IoT-enabled low-cost TMIS systems. The future work can be analyzed under some of the limitations of the proposed protocol. The backed server can obtain all the RFID tag’s communications if it is too powerful. Besides, an adversary can also obtain all the secret parameters and then can perform some forgery attacks. We wish to resolve such security issues in our future research.

Declaration of Competing Interest

All authors have participated in (a) conception and design, or analysis and interpretation of the data; (b) drafting the article or revising it critically for important intellectual content; and (c) approval of the final version. This manuscript has not been submitted to, nor is under review at, another journal or other publishing venue. The authors have no affiliation with any organization with a direct or indirect financial interest in the subject matter discussed in the manuscript.

References

Afroz, T., Ihsan, M. N. U., & Uddin, M. N. (2019). A secure mutual authentication protocol for IoT using id verifier based on ecc. 2019 international conference on sustainable technologies for industry 4.0 (siti) (pp. 1-6). IEEE.
Alamr, A. A., Kausar, F., Kim, J., & Seo, C. (2018). A secure ecc-based rfid mutual authentication protocol for internet of things. The Journal of supercomputing, 74(9), 4281–4294.
Armando, A., Bastin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuéllar, J., Dreihua, P. H., Heam, P.-C., Kouchanenko, O., Mantovani, J., et al. (2005). The avigma tool for the automated validation of internet security protocols and applications. International conference on computer aided verification (pp. 281–285). Springer.

Bernal-Mahal, M., Djedjou, M., & Drouiche, K. (2017). Security analysis and enhancement of the most recent rfid authentication protocol for telecare medicine information system. Wireless Personal Communications, 96(4), 6221–6238.

Chauhan, V., Galwankar, S., Arquilla, B., Garg, M., Di Somma, S., El-Menayar, A., Krishnan, V., Gerber, J., Hollund, R., & Starwick, S. P. (2020). Novel coronavirus (covid-19): Leveraging telemedicine to optimize care while minimizing exposures and viral transmission. Journal of emergencies, trauma, and shock, 13(1), 20.

Chen, X., Geng, D., Zhai, J., Liu, W., Zhang, H., & Zhu, T. (2020). Security analysis and enhancement of the most recent rfid protocol for telecare medicine information system. Wireless Personal Communications, 114(2), 1371–1387.

Chen, Y., & Chou, J.-S. (2015). Epc-based untraceable authentication for large-scale active-tag rfid systems. Electronic Commerce Research, 15(1), 97–126.

Chou, J.-S. (2014). An efficient mutual authentication rfid scheme based on elliptic curve cryptography. The Journal of supercomputing, 70(1), 75–94.

Cremers, C. J. (2008). The scyther tool: Verification, falsification, and analysis of security protocols. International conference on computer aided verification (pp. 414–418). Springer.

Dharmander, D., Mishra, D., & Li, X. (2020). Construction of rsa-based authentication scheme in authorized access to healthcare services. Journal of medical systems, 44(1), 1–9.

Dolev, D., & Yao, A. (1983). On the security of public key protocols. IEEE Transactions on Information Theory, 29(2), 198–208.

Fan, K., Jiang, W., Li, H., & Yang, Y. (2018). Lightweight rfid protocol for medical privacy protection in iot. IEEE Transactions on Industrial Informatics, 14(4), 1656–1665.

Farah, M. S., Nawaz, O., Mahmood, K., Chaudhry, S. A., & Khan, M. K. (2016). A provably secure rfid authentication protocol based on elliptic curve for healthcare environments. Journal of medical systems, 40(7), 165.

Fouladi, S. M. J. E., Ali, A., Mobd, Y. B., & Ali, A. (2021). Efficient deep neural networks for classification of covid-19 based on ct images: Virtualization via software defined radio. Computer communications, 176, 234–248.

Guo, C., Tian, P., & Choo, K. K. R. (2020). Enabling privacy-protected fog-based data aggregation in e-healthcare systems. IEEE Transactions on Industrial Informatics, 17 (3), 1948–1957.

Gupta, V., Nikita, J., Piyush, K., Adarsh, K., Senhilkumar, M., Ali, A., & Massimiliano, F. (2021). An emotion care model using multimodal textual analysis on covid-19. Chaos, Solitons & Fractals, 144(3), 110708.

Haddara, M., & Staaby, A. (2018). Rfid applications and adoptions in healthcare: A review on patient safety. Procedia computer science, 138, 80–88.

Jin, C., Xu, C., Zhang, X., & Li, F. (2016). A secure ecc-based rfid mutual authentication protocol to enhance patient medication safety. Journal of medical systems, 40(1), 1–6.

Kaul, S. D., & Awasthi, A. K. (2013). Rfid authentication protocol to enhance patient medication safety. Journal of medical systems, 37(6), 1–6.

Kundu, R., Hrimat, B., Pawan, K. S., Ali, A., Massimiliano, F., & Ram, S. (2021). Fuzzy rank-based fusion of cnn models using gompertz function for screening covid-19 ct-scans. Scientific reports, 11(1), 1–12.

Li, C.-T., Weng, C.-Y., & Lee, C.-C. (2015). A secure rfid authentication protocol with enhanced security for tmis. Journal of medical systems, 39(8), 1–8.

Liao, Y.-P., & Hsiao, C.-M. (2014). A secure ecc-based rfid authentication scheme integrated with id-verifier transfer protocol. Ad hoc networks, 18, 133–146.

Moazzami, B., Razavi-Khorasani, N., Moghadam, A. D., Farokhi, E., & Rezaei, N. (2020). Covid-19 and telemedicine: Immediate action required for maintaining healthcare providers well-being. Journal of Clinical Virology, 126, 104345.

Noori, D., Shakeri, H., & Torshiz, M. N. (2020). Scalable, efficient, and secure rfid with elliptic curve cryptosystem for internet of things in healthcare environment. EURASIP Journal on Information Security, 2020(1), 1–11.

Ohanessian, R., Duong, T. A., & Odone, A. (2020). Global telemedicine implementation and integration within health systems to fight the covid-19 pandemic: A call to action. JMIR public health and surveillance, 6(2), e18810.

Ouali, K., & Phan, R. C.-W. (2008). Privacy of recent rfid authentication protocols. International conference on information security practice and experience (pp. 263–277). Springer.

Peeters, R., & Hermans, J. (2013). Attack on liu and hsiao’s secure ecc-based rfid authentication scheme integrated with id-verifier transfer protocol. IACR Cryptol. ePrint Arch., 2013, 399.

Prakash Pokala, J., Reddy, M. C., Bapana, S., Vorugunti, C. S., et al. (2016). A secure rfid protocol for telecare medicine information systems using ecc. 2016 international conference on wireless communications, signal processing and networking (wispnet) (pp. 2295–2300). IEEE.

Salkhani, M., & Vasilakos, A. (2019). A new secure authentication protocol for telecare medicine information system and smart campus. IEEE Access, 7, 23514–23526.

Saha, P., Debadyuti, M., Pawan, K. S., Ali, A., Massimiliano, F., & Ram, S. (2021). Graphcovidnet: A graph neural network based model for detecting covid-19 from ct scans and x-rays of chest. Scientific reports, 11(1), 1–16.

Salem, F. M., & Amin, R. (2020). A privacy-preserving rfid authentication protocol based on el-ganamal cryptosystem for secure mts. Information sciences, 527, 382–393.

Shariq, M., & Singh, K. (2021). A novel vector-space-based lightweight privacy-preserving rfid authentication protocol for iot environment. The Journal of supercomputing, 1–31.

Shariq, M., Singh, K., Moursy, P. K., Ahmadian, A., & Ariffin, M. R. K. (2021). Utrust: An ultralightweight rfid authentication scheme using permutation operation. Peer-to-Peer Networking and Applications, 1–21.

Shen, H., Shen, J., Khan, M. K., & Lee, J.-H. (2017). Efficient rfid authentication using elliptic curve cryptography for the internet of things. Wireless Personal Communications, 96(4), 5253–5266.

Smith, A. C., Thomas, E., Snowsell, C. L., Haydon, H., Mehrotra, A., Clemenson, J., & Caffery, L. J. (2020). Telehealth for global emergencies: Implications for coronavirus disease 2019 (covid-19). Journal of telemedicine and telecare, 26(5), 309–313.

Srivastava, R., Awasthi, A. K., Kaul, S. D., & Mittal, R. (2015). A hash based mutual rfid tag authentication protocol in telecare medicine information system. Journal of medical systems, 39(1), 1–5.

Tu, Y.-J., Kapoor, G., & Piramuthu, S. (2020). Security of lightweight mutual authentication protocols. The Journal of supercomputing, 1–17.

Vidal-Alaball, J., Acosta-Roja, R., Hernandez, N. P., Luque, U. S., Morrison, D., Perez, S. N., Perez-Llano, J., Verges, A. S., & Seguit, F. L. (2020). Telemedicine in the face of the covid-19 pandemic. Atencion primaria, 52(6), 418–422.

Xiao, L., Xie, S., Han, D., Liang, W., Guo, J., & Chou, W.-K. (2021). A lightweight authentication scheme for telecare medical information system. Connection science, 1–17.

Xie, S., Zhang, F., & Cheng, R. (2020). Security enhanced rfid authentication protocols for healthcare environment. Wireless Personal Communications, 1–16.

Zhang, Z., & Qi, Q. (2014). An efficient rfid authentication protocol to enhance patient medication safety using elliptic curve cryptography. Journal of medical systems, 38 (5), 1–7.

Zhou, Z., Wang, P., & Li, Z. (2019). A quadratic residue-based rfid authentication protocol with enhanced security for tmis. Journal of ambient intelligence and humanized computing, 10(9), 3603–3615.