Wireless Personal Communications (2021) 120:2653–2691
https://doi.org/10.1007/s11277-021-08552-7

A Secure and Privacy Friendly ECC Based RFID Authentication Protocol for Practical Applications

Atakan Arslan[1,2] · Sultan Aldırmaz Çolak[1] · Sarp Ertürk[1]

Accepted: 25 April 2021 / Published online: 15 May 2021

Abstract
Radio frequency identification (RFID) is a promising and widespread wireless communication technology for entity identification or authentication. By the emerging Internet of Things phenomenon, the use of RFID is densely augmenting in various daily life applications. However, RFID systems suffer from security and privacy issues. Recently, many researchers propose RFID authentication protocols based on elliptic curve cryptography (ECC) to efficiently mitigate the aforementioned concerns. In this work, we extensively examine the state-of-the-art RFID authentication protocols based on ECC in terms of security and performance. Some of these works claim that their protocols provide all general security and privacy properties. We revisit Vaudenay's formal privacy model and show that they do not provide forward and/or backward privacy under this model contrary to their claim. Then, we propose a secure, privacy-preserving and efficient ECC based RFID authentication protocol. We also present a security and performance analysis of our proposed protocol and compare it to the existing relevant schemes in detail. Furthermore, we implement our proposal in a real RFID system to demonstrate its practicability. To the best of our knowledge, our proposed scheme is the most efficient ECC based RFID authentication protocol realized in a real-world environment that satisfies all common security and privacy features including backward and forward privacy.

Keywords Privacy · Security · RFID · IoT · ECC · Implementation

✉ Atakan Arslan
atakan.arslan@kocaeli.edu.tr

Sultan Aldırmaz Çolak
sultan.aldirmaz@kocaeli.edu.tr

Sarp Ertürk
sertur@kocaeli.edu.tr

[1] Department of Electronics and Communication, Kocaeli University, 41380 İzmit, Kocaeli, Turkey

[2] TÜBİTAK BİLGEM Informatics and Information Security Research Center, 41470 Gebze, Kocaeli, Turkey

1 Introduction

Nowadays, many emerging technologies are widely used with advances in information and communication disciplines. The increasing demand for identification and authentication promotes Radio Frequency Identification (RFID) as a pervasive and promising wireless communication technology. The popularity of RFID has been rising day by day with the expeditious development of the Internet of Things (IoT) paradigm. In fact, the first idea of IoT was originated from a network of objects connected by RFID, and IoT tells us that "anything that can be connected, will be connected". It is predicted that by 2020, the number of daily life things that will be connected to each other will reach about 50 billion [1]. This means that RFID will continue to have a high impact on our daily activities and behaviors, and penetrate in our everyday lives rapidly by providing easy, efficient, cheap, secure and private connections of "*things*" which also includes people [2]. Although RFID technology is used in numerous real-world applications such as payment systems, healthcare system, e-passports, e-voting, national e-ID management, smart homes, access control, manufacturing, asset management, supply chain, etc. [3–7], RFID is still regarded to be its infancy today [8].

A basic RFID system includes a back-end server, a reader (verifier) and a tag (prover). An RFID tag is designed for wireless data transmission with a chip and an antenna. The tag uses the chip for processing and storing of information, and an antenna is used for wireless communication, respectively. The back-end server/database stores all data (keys, IDs, etc.) about tags. An RFID reader interrogates tags and relays the gathered information to the server.

RFID tags are classified into passive, semi-passive and active tags. Passive tags which are battery-free, solely use the back-scattered interrogation signal of the reader to energize their chips. Active tags are battery-supported and have their own power source. Semi-passive tags are triggered by the reader (need the reader's magnetic or electromagnetic field for transmitting data) and use their own power source for internal processing. In many application, battery-free low-cost RFID tags are preferred because of their smaller sizes and prices. However, passive low-cost tags have some difficulties such as computation, energy and size restrictions [9]. Furthermore, several standards are also published for the use of RFID systems in different ranges and operating frequencies [4]. For instance, ISO 15693, ISO 14443 and ISO 18000-3 standards are developed for high frequency (HF) and ISO 18000-6, ISO 18000-7 are intended for ultra high frequency (UHF). In fact, the development and improvement of NFC standards also increase the use of HF RFID tags.

Security and privacy concerns in RFID systems result from exchanging sensitive information (i.e. credit card data, personal healthcare data) of tags with a reader in an insecure wireless channel. An adversary might be able to catch and change the messages transmitted in the air. The adversary can cause security and privacy issues with performing various attacks such as tag impersonating, reader spoofing man-in-the-middle (MiTM), tracking, replay, denial of service (DoS) attacks, etc. Therefore, many authentication schemes have been designed for mitigating security and privacy problems in RFID systems [10]. In the RFID literature, all protocol designers claim that their own schemes are secure and privacy-friendly while providing some other requirements such as mutual authentication and scalability. However, it is shown in the literature that most of them are not resistant to every type of attacks and do not efficiently accomplish a least one of security and privacy properties such as forward privacy, backward privacy, impersonation resistance, desynchronization resistance etc [10–16]. Also, some RFID privacy models are presented to methodically

and formally analyze authentication schemes in terms of security and privacy. Although Vaudenay's model [17] is still successful and acceptable, in the course of time, several works have been proposed to improve and extend his model [11, 18–20].

Public-key (asymmetric) cryptography (PKC) can bring elegant solutions to security and privacy problems stated above. Especially nowadays, elliptic curve cryptography (ECC) is preferred in various RFID authentication schemes in order to reduce the key sizes, memory storage, and computation cost. Many protocol designers think that using ECC in their designs efficiently and achieve security and privacy properties (see Sect. 2). Although some researchers have doubts that PKC might not be affordable for constraint tags, the feasibility of using ECC in the tags is shown in [21–25]. Moreover, both privacy and scalability in RFID systems are more easily accomplished by using PKC rather than symmetric cryptographic blocks [25, 26].

On the other hand, the feasibility of using ECC in practice is important for real life RFID applications. Recent works show the implementations of their protocols in different environments such as Wireless Identification and Sensing Platform (WISP), Field Programmable Gate Array (FPGA) [27, 28]. There are also some RFID tags that are presented for implementations in Java cards, BasicCard, Mifare Card, and NFC cards. Especially HF RFID tags including NFC (Near Field Communications) tags have been densely preferred for IoT security applications [29]. In particular, the BasicCard environment [30] offers good opportunities for RFID systems as a powerful development tool in simulation and implementation.

2 Related Work

This section introduces previous works in ECC based RFID authentication protocols and outlines the contributions of this paper. To solve the various security and privacy problems in RFID systems, countless RFID protocols have been published.

A recent comprehensive survey of related work about these protocols is provided in Avoine's RFID Security and Privacy Lounge [10]. Among all researchers that used public key cryptography (PKC), nearly all preferred ECC-based protocols because of their ability to provide stronger security with smaller key sizes, as well as lighter and efficient computations.

Wolkerstorfer [31] asserts that ECC implementation in RFID tags was suitable. Tulys and Batina [32] firstly propose an ECC-based RFID identification scheme using the Schnorr identification protocol [33] by referring to the conclusion of Wolkerstorfer's work. They claim that their scheme is secure against cloning attacks. But, the implementation of this protocol is caused by security and privacy vulnerabilities. In the interactive phase, an adversary can obtain the information to calculate the ID-verifier and she can track the tag. The protocol has also a scalability problem. In the authentication phase, the verifier has to search the many public keys for each tag. Moreover, this protocol does not provide mutual authentication and anonymity [34].

Later, Batina et al. [24] propose a new scheme by applying Okamoto's identification protocol [35] to improve security and privacy. They also aim to discuss the feasibility of ECC based RFID identification protocols and present the implementation of Okamoto's protocol as an example. However, Batina et al.'s protocol does not solve the security, privacy and efficiency issues [36]. The adversary still can obtain the ID-verifier and track the

tag. In addition, the forward privacy is not provided in Batina et al.'s scheme similar to the situations in [32].

Lee et al. [34] show the weaknesses of Schnorr's and Okamoto's identification problems and propose a new RFID authentication protocol named EC-RAC using ECC to mitigate the security and privacy flaws mentioned above. But it is shown that this protocol has security and privacy issues, and is vulnerable to tracking attack, MiTM attack, algebraic attacks, etc. [25, 37–40]. Similarly, the protocol provides only one-way authentication.

Lee et al. [41] revise the EC-RAC protocol [34] and propose six different RFID authentication protocols by expanding the EC-RAC protocol. They state that their protocols are secure against common attacks, but each protocol provides different security properties. Lee et al. [39] address the existing vulnerabilities of EC-RAC protocols and present a new efficient searching scheme for the the RFID reader so as to query for a specific tag while protecting the tag's privacy.

Zhang et al. [42] present an ECC-based randomized key RFID authentication protocol to improve EC-RAC and Schnorr protocols to defeat their weaknesses. The proposal focuses on finding a way to solve the tracking attack effectively. However, this scheme is defenseless to active-tracking attack. Furthermore, updating tag information increases the computation complexity of the back-end server and causes scalability problems in this scheme. It also lacks mutual authentication [43]. Lv et al. [40], in 2012, show the weaknesses of EC-RAC protocols and propose three ECC-based RFID protocols which are the revision of EC-RAC protocols to overcome tracking attack. Later, An et al. [44] demonstrate that Lv et al.'s protocols are not secure against MiTM attack.

Liao and Hsiao [45] present an ECC-based RFID authentication scheme to satisfy the essential requirements of RFID systems including mutual authentication, anonymity, forward privacy, confidentiality, and scalability. But, it is shown that this scheme is inadequate in terms of computational cost and memory storage [43, 46–48]. Zhao [49] proposes a new protocol and shows that Liao and Hsiao's protocol suffers from the key compromise attack in which the adversary can obtain the private key stored in the tag. It is shown that Liao and Hsiao's protocol does not achieve any security and privacy properties in [50]. Chien [43] also proves that Liao and Hsiao's protocol is vulnerable to active tracking attack. Zhao's scheme does not provide tag anonymity, location privacy, data integrity, backward and forward privacy [27, 28, 51].

Later, Chou [36], designs a new and efficient RFID mutual authentication protocol based on ECC. Unfortunately, this scheme is defenseless against tag impersonation, cloning, and tracking attacks and it also does not satisfy tag anonymity, forward privacy and mutual authentication [28, 52, 53]. Zhang and Qi [53] point out that Chou's scheme does not provide tag information, backward and forward privacy. They also propose an enhanced new RFID scheme based on Chou's protocol to overcome the vulnerabilities of his scheme. But, it is shown that Zhang and Qi's scheme does not provide location privacy, backward and forward privacy [27, 28]. In the same year, He et al. [47] propose a new ECC RFID scheme that integrated with an ID verifier transfer. However, Jin et al. [54] state that He et al.'s scheme is not resistant to various attacks such as tag impersonation, server spoofing, replay, DoS, etc. On the other hand, in 2015, He and Zeadally [13] present a detailed survey of ECC based RFID authentication protocols up to that date.

Farash et al. [48] demonstrate the security and privacy vulnerabilities of [36, 45, 49, 53]'s schemes. In fact, it is shown that none of them provide forward privacy and provable security. Farash et al. also compare their performance and propose an efficient RFID authentication scheme to improve the security and privacy of previous protocols. However, their protocol does not fulfill tag anonymity and location privacy [27]. Jin et al. [55]

present an RFID mutual authentication scheme based on ECC to enhance patient privacy while achieving security requirements and overcoming various existing attacks. But, it is shown that their scheme does not provide data integrity and is vulnerable to key compromise problem [51, 56].

Chien [43] shows the attacks on [42, 45]'s schemes and proposes a new ECC-based RFID mutual authentication to defeat the security weaknesses. In the same year, Benssalah et al. [27] propose a secure RFID authentication scheme (we call BDD17) based on elliptic curve message recovery (ECMR) signature to provide significant security features and better performance compared to famous authentication protocols based on ECC in the RFID literature. They analyze their design using a formal security analysis with a random oracle model and claim that their protocol is provably secure. Besides, they implement ECMR in FPGA and validate its practical feasibility. However, it is shown in this paper that BDD17 scheme does not provide forward and backward privacy.

Ibrahim and Dalkılıç [28] propose an authentication scheme (we call ID17) for RFID tags based on both symmetric and asymmetric cryptographic algorithms such as ECC and advanced encryption standard (AES). They claimed that their protocol design is secure, private and provides mutual authentication only in two steps. Moreover, they implement their protocol in the wireless identification and sensing platform (WISP5) and present the performance results. However, it is shown again in this paper that their proposal does not provide forward and backward privacy, as they claim.

Alexander et al. [51] present a survey of the most promising ECC based RFID authentication protocols proposing a different methodology to evaluate recent RFID schemes. They develop a ranking method to compare several RFID protocols in terms of performance and security properties. However, in their evaluation, all ranking points in each category are equal. In other words, different ranks in different categories are weighted the same value. For instance, if a scheme is vulnerable to an attack (i.e. impersonation attack), it loses only a point and is classified into an appropriate rank order. We do not agree with their evaluation because we think that firstly security and performance of a scheme should be evaluated separately, and secondly ranking the security properties or performance of a scheme is not the proper approach to compare RFID protocols because it is hard to grade a certain security property among the others. Besides, Alexander et al. claim that Dinarvand and Barati's [56] scheme (we call DB19) provides all security and privacy requirements. But, we show that their scheme grade security does not provide backward privacy.

Liu et al. [57] propose a novel ECC based RFID authentication protocol (we call LZKZ18) establishing a key negotiation mechanism. They claim that their protocol design has higher security and privacy. However, it is shown in this paper that their scheme does not achieve forward and backward privacy.

Most recently, several ECC-based RFID authentication protocols have been proposed [58–64]. Kumar et al. present a framework called RSEAP for vehicular cloud computing [61]. They claim that their protocol provides security and privacy by using formal and information security analyses. However, Safkhani et al. [62] publish a new authentication scheme named RSEAP2 by showing the weaknesses of [61]. Kumari et al. [65] present a protocol called ESEAP and claim that it is more secure than the predecessors but [66, 67] state that their scheme is insecure. Also, Izza et al. [63] propose an ECC-based RFID authentication protocol to fulfill the security and privacy of healthcare applications by overcoming the security issues of Naeem et al. [60]. They also claim that their scheme provides better security than [56, 58, 59, 68]. However, their design has heavy computation, communication, and storage costs. Lastly, Agrahari and Varma [64] define a new scheme based on the EC Qu-Vanstone concept but

they present the performance of their design without implementing it. Moreover, they claim that their protocol satisfies only forward untraceability property but they do not mention backward untraceability.

To sum up, we present the following evaluations as an analysis of the literature review. Firstly, we realize that almost all of ECC-based RFID authentication protocols have not been implemented. Hence, the practical performance of these schemes in a real-world environment is questionable. Secondly, most of them suffer from many security and privacy vulnerabilities. The rest of them are not able to satisfy all common security and privacy requirements. Especially, both forward and backward secrecy properties have been overwhelmingly not mentioned in their security analysis. Finally, while many schemes claim to provide higher security and privacy, their performances are less efficient in terms of computation and communication costs. Motivation by this need, we focus on RFID authentication protocols using ECC mechanisms to lead our contributions to the literature. The contributions of this paper can be summarized as follows:

- We show that ID17 [28], BDD17 [27], DB19 [56] and LZKZ18 [57] do not provide forward and/or backward privacy, contrary to their claim. We reveal the vulnerabilities of these schemes under Vaudenay's [17] formal privacy model. For this purpose, we first revisit Vaudenay's model and then, we prove our attacks by utilizing privacy games.
- We propose a new secure and privacy-friendly ECC based authentication protocol by improving ID17. We elaborately analyze our proposed scheme in terms of security and performance and our analysis indicates that our scheme achieves all well-known security and privacy properties.
- We present implementation results of our proposed protocol in a real-world RFID system in order to show the practicability and feasibility of our proposal.
- We present detailed security and performance comparisons between our protocol and the related existing schemes. To the best of our knowledge, in the RFID literature, we claim that only our proposal is the up-to-the-minute implemented and tested protocol that can efficiently satisfy all essential security and privacy requirements for an RFID system

The rest of this paper is organized as follows. In Sect. 3, we give preliminary information about common cryptographic concepts used in this paper. In Sect. 4, we revisit Vaudenay's privacy model that will be used in our attacks. Then, in Sect. 5, we present the security analysis of recent works [27, 28, 56, 57] under Vaudenay's model to show privacy vulnerabilities. In Sect. 6, we describe our proposed protocol. In Sect. 7, we show the security analysis of our proposal. In Sect. 8, we compare our proposed protocol with existing schemes in terms of security and performance aspects, and also describe real world implementation. Finally, Sect. 9 concludes the paper.

3 Preliminaries

In this section, we introduce some preliminaries on main cryptographic topics utilized in this paper such as ECC cryptosystems, hash functions, and digital signatures.

3.1 Elliptic Curve Cryptography Cryptosystems

ECC is public-key cryptography based on elliptic curves over Galois or finite fields. More than 30 years ago, the use of EC in cryptography is firstly discussed by Koblitz [69] and Miller [70], independently. Today, more than billions of wireless communication systems prefer ECC based solutions to fulfill the security requirements in different sectors such as financial services, health care, government services, etc., because they need efficient and secure asymmetric cryptosystem for confidentiality, integrity, authentication, privacy, nonrepudiation (i.e. signature), etc. The advantages of ECC for wireless security is briefly overviewed in [71].

3.1.1 Theory of ECC

In this subsection, initially the theory of ECC is summarized, then security and benefits of ECC are discussed. An *elliptic curve* (E) used for cryptographical purposes can be generally defined over a prime finite field $\mathbb{F}_p$ (or Galois field) includes a group of points (x, y) that satisfies $y^2 \equiv x^3 + ax + b(modp)$ equation, where $(a, b) \in E$, $\triangle = 4a^3 + 27b^2 \implies \triangle \not\equiv 0(modp)$ and p is a large prime number. The EC cyclic group is formally defined as $E(\mathbb{F}_p) = \{(x, y) : x, y \in E(a, b)\} \cup \{\mathcal{O}\}$, where $\mathcal{O}$ denotes point at infinity and satisfies the following group properties. Let $\forall R, S \in E\left(\mathbb{F}_p\right)$ and $R = (x_1, y_1)$, $S = (x_2, y_2)$,

- Identity: $R + \mathcal{O} = \mathcal{O} + R$
- Negatives: $R + (-R) = \mathcal{O}$, where $-R = \left(x_1, -y_1\right)$. Also, $\mathcal{O} = -\mathcal{O}$
- Point addition: $R + S = \left(x_3, y_3\right) \in E\left(\mathbb{F}_p\right)$
- Point doubling: $P \neq (-P) \implies P + P = 2P = \left(x_3, y_3\right) \in E\left(\mathbb{F}_p\right)$

Order of an EC group refers to the number of points (elements) in that group and can be easily computed by Schoof's algorithm. Actually, ECC uses cyclic subgroups formed by EC with having cyclically repeated points. This type of groups has a base point (generator). Note that Schoof's algorithm cannot be used for finding the order of the subgroup. Let G be a cyclic subgroup of $E\left(\mathbb{F}_p\right)$ with order k and generator P, then $nP = \mathcal{O}$. Furthermore, a cofactor of G is $h = N/k$, where N is the order of $E\left(\mathbb{F}_p\right)$ and $h \in \mathbb{Z}$ because of Lagrange's theorem. In fact, cryptographers want a high order of an EC subgroup so before they find a generator, they first choose a large enough order then try to reach a suitable generator.

Lastly, the EC point multiplication is defined as $Q = aP$, where $Q, P \in E(F)$ and $a \in Z$. This operation corresponds to adding P by itself a times.

Domain parameters: ECC domain parameters are all the elements defining the EC (E) such as base point, prime order, cofactor of the base point, etc. Both tag and reader must agree on them to securely and efficiently use ECC. Their generation of them is out of the scope of this work. The are several secure standard curves and we prefer one of them, namely the ECC Brainpool [72].

3.1.2 Security and Benefits of ECC

The security of ECC-based schemes depends on the hardness of the EC discrete logarithm problem (ECDLP).

Definition 1 (*Elliptic curve discrete logarithm problem (ECDLP)*). Given $P, Q \in E(F_q)$ and $Q = kP$ where $k \in [1, q-1]$ and q is the point order. Then, it is hard to compute k by an algorithm in polynomial-time.

Using the EC scheme offers some advantages: smaller key sizes with respect to the other known PKC algorithms (at the same security level, see Table 1 [73]), higher speed, reduced memory storage as well as consumed power and bandwidth efficiency. Thus, these benefits make ECC desirable in many usage areas of asymmetric cryptographic schemes such as key agreement, encryption and digital signatures [71, 74–78].

3.2 Elliptic Curve Diffie-Hellman (ECDH)

Generally, ECDH (a variant of the DH scheme) is a secure key agreement scheme whereby two or more entities can agree on a secret key by using ECC over an insecure channel. Both entities already have pre-shared public keys of each other. They use their own private keys to recover the shared key, but an adversary cannot calculate the shared key from the public information.

Definition 2 (*Computational Elliptic Curve Diffie-Hellman Problem*). Given P, R $, S \in E(\mathbb{F}_q)$, $S = sP$ and $R = rP$ where $s, r \in [1, q-1]$ and q is the order of the base point P. Then, it is hard to compute srP by an algorithm in polynomial-time.

Evidently, these problems satisfy $DLP \Leftarrow DHP$. For specific groups, DHP is sometimes called Diffie-Hellman assumptions because DHP is assumed as a hard problem. Moreover, public keys in ECDH schemes might be either static or ephemeral (ECDHE).

3.3 Elliptic Curve Digital Signature Algorithm (ECDSA)

ECDSA is a variant of the Digital Signature Algorithm (DSA) that uses ECC. ECDSA is used for authentication, non-repudiation, and integrity. Therefore, the source of the message is authenticated, the entity that transmitted the message cannot deny it and the integrity of the message is ensured over an insecure channel. In this scheme, basically, the signer signs the message by using its own secret key and the verifier verifies the signature with a public key of the signer by using ECC. The security of ECDSA is based on ECDLP. Moreover, ECDSA is more effective than other known schemes such as RSA and DSA. It is accepted by ANSI, IEEE, and NIST.

Table 1 Key size (bits) comparisons for equivalent security levels [73]

Minimum strength (bits)	Symmetric algorithm	RSA and DL group (bits)	ECC (bits)
≤ 80	Two-key 3DES	1024	160
112	Three-key 3DES	2048	224
128	AES	3072	256
192	AES	7680	384
256	AES	15360	521

3.4 Cryptographic Hash Function

Cryptographic hash functions are used in many cryptographic schemes to provide integrity service as deterministic algorithms [79]. It can be formally defined as below.

Definition 3 (*Hash Function*). Hash function is a function that takes an arbitrary length of inputs and maps a fixed size outputs. Let H be a hash function $H(x) = y$, where x is arbitrary sized input string and y is fixed size output string. H should be deterministically computable in polynomial time. H should provide the following properties to be a cryptographic hash function.

- **Pre-image resistance:** This property means that any polynomial-time algorithm cannot find the input of a hash function for a given output. Finding x is hard for a given y, i.e. $H(x) = y$.
- **Second pre-image resistance:** This property means that any polynomial-time algorithm cannot find a new input for a given input and output pair of a hash function. Finding $x^{'}$, where $x^{'} \neq x$ is hard for a given x, y, i.e. $H(x) = y$.
- **Collision resistance:** This property means that any polynomial-time algorithm cannot find two different inputs to map the same output for a hash function. Finding $\left(x, x^{'}\right)$ that provides $y = y^{'}$ equality is hard for a hash function H, i.e. $H(x) = y$ and $H\left(x^{'}\right) = y^{'}$.

If H is resistant against collision attacks, it always provides second pre-image resistance, otherwise, but opposite implication might not be valid. This assumption is theoretically true, however, it is recommended that cryptographic hash functions need to satisfy all three requirements in practical applications. In practice, there are several known cryptographic hash functions with different digest sizes from 128 bits to 512 bits e.g. SHA family MD5, BLAKE, etc. Finally, all secure hash algorithms are published by NIST as a standard (FIPS).

4 Security and Privacy of RFID Schemes

In this section, we briefly introduce Vaudenay's model (VM) [17]. VM is an acknowledged, a well-defined and mature model that allows methodological security and privacy analysis of RFID schemes.

Basically, a simple RFID system consists of three entities such as a tag $\boldsymbol{T}$, a reader $\boldsymbol{R}$ and a back-end system/database $\boldsymbol{DB}$. A reader $\boldsymbol{R}$ interrogates a tag $\boldsymbol{T}$ and identifies/authenticates $\boldsymbol{T}$ by using its identifier $ID_{\boldsymbol{T}}$. Also, $\boldsymbol{DB}$ stores all information (secret keys, parameters, identifiers, etc.) of the valid tags. $\boldsymbol{R}$ has a secure communication with $\boldsymbol{DB}$ and accesses $\boldsymbol{DB}$ during authentication/identification phase of $\boldsymbol{T}$. $\boldsymbol{DB}$ might be constituted as a part of the reader. Furthermore, we assume that $\boldsymbol{R}$ is more powerful than the tag. $\boldsymbol{T}$ has a less computational capacity, memory storage, and a limited energy source. We also assume that an adversary $\boldsymbol{A}$ is able to corrupt $\boldsymbol{T}$ and utilize its internal sensitive data against the whole RFID system but she cannot corrupt $\boldsymbol{R}$ because it is considered as a tamper-resistant device. $\boldsymbol{A}$ always tends to attack the RFID system by investigating the vulnerabilities of an RFID protocol.

VM can be summarized in modeling an RFID scheme, adversary and privacy classes. For further detailed information on VM, [17] can be examined.

4.1 RFID Scheme Model

According to VM, an RFID scheme ***Sch*** can be constructed by the procedures of SetupReader, SetupTag, and IdentTag. By using SetupReader algorithm, public and private key pairs of $\boldsymbol{R}$ and a corresponding empty ***DB*** are generated and a public key is distributed to all entities. SetupTag is a polynomial-time probabilistic algorithm that generates a $\boldsymbol{T}$ with a secret key and updatable internal memory, assigns a unique identifier *ID* and inserts $\boldsymbol{T}$ into ***DB*** when it is valid. On the other hand, IdentTag is a polynomial-time algorithm that runs an interaction protocol between $\boldsymbol{R}$ and $\boldsymbol{T}$. Finally, it outputs identifier *ID* of $\boldsymbol{T}$ if $\boldsymbol{T}$ is valid. But if $\boldsymbol{T}$ is not legitimate, it outputs $\perp$. The result of this protocol execution might reveal some secondary information to an adversary.

4.2 Adversaries

An adversary $\boldsymbol{A}$ is known as a malicious party who aims to break the security and privacy of an RFID scheme by using its vulnerabilities. Formally, an adversary can be defined as a polynomial-time algorithm takes all public information of an ***Sch*** and is able to run the below oracles. Note that a polynomial-time adversary is just able to run polynomial time algorithms due to the fact that her computational abilities are asymptotically bounded. In general, $\boldsymbol{A}$ may act as a dishonest reader to communicate with a $\boldsymbol{T}$ but we assume that $\boldsymbol{T}$ is unaware of this interaction and is deceived that $\boldsymbol{A}$ is the legitimated $\boldsymbol{R}$. Therefore, there are three essential characteristics: (i) querying *oracles*, (ii) playing games to attack the system and reach her goal and (iii) interacting with the system using the *rules* of the game.

We consider that in an RFID scheme, there is solely one $\boldsymbol{R}$ and there might be more than one legitimate or illegitimate tags. We also consider that these tags have only one status free or are drawn for a certain time and this status can be changed by the related oracle. Moreover, we assume that $\boldsymbol{R}$ cannot be corrupted, whereas $\boldsymbol{T}$ cannot be tamper-resistant. Hence, the secret key K_s is kept secret. We also assume that at the beginning of each game, there are no tags. The adversary plays her game only using the following oracles:

- $\mathcal{O}^{CreateTag}(ID, l)$: Let *ID* is a unique identifier and $l \in \{0, 1\}$, this oracle generates a fresh $\boldsymbol{T}$ with *ID* and registers the tag by updating ***DB***, if $l = 1$. Else, the newly generated tag is invalid.
- $\mathcal{O}^{DrawTag}\left(p_d, n\right) \rightarrow (\psi_{\boldsymbol{T}_1}, l_1, \ldots, \psi_{\boldsymbol{T}_n}, l_n)$: This oracle randomly chooses n free tags from previously generated ones with a given probability distribution p_d by changing their status to *drawn* and dedicates an ephemeral pseudonym ψ_i (anonymously addressing $\boldsymbol{T}_i$) to the drawn i^{th} tag for each selection. The adversary can interact with *drawn* tags for only one individual session because the pseudonyms are refreshed for each session. The relations between pseudonyms and *IDs* $(\psi_{\boldsymbol{T}_i}, ID_i)$ are stored in a hidden table. This oracle also outputs a bit array $\left(l_1, l_2, \ldots, l_n\right)$ where l_i of the i^{th} tag shows whether it is valid or not. Moreover, the oracle may return $\perp$ if there are no existing tags or the querying tags are already *drawn*.
- $\mathcal{O}^{Free}\left(\psi_{\boldsymbol{T}}\right)$: This oracle changes the status of *drawn* $\psi_{\boldsymbol{T}}$ tag to *free* and $\boldsymbol{A}$ cannot reach $\boldsymbol{T}$ anymore using the $\psi_{\boldsymbol{T}}$
- $\mathcal{O}^{Corrupt}\left(\psi_{\boldsymbol{T}}\right) \rightarrow \mathfrak{M}$: This oracle allows to capture the whole memory $\mathfrak{M}$ of $\boldsymbol{T}$ taking $\psi_{\boldsymbol{T}}$.

- $\mathcal{O}^{Launch}() \rightarrow \pi$: This oracle runs an IDENTTAG by utilizing legitimate $\boldsymbol{R}$ and outputs a session identifier π of this protocol instance.
- $\mathcal{O}^{SendReader}(msg, \pi) \rightarrow msg'$: This oracle sends msg messages to $\boldsymbol{R}$ for π protocol session and outputs the responding message msg' by $\boldsymbol{R}$.
- $\mathcal{O}^{SendTag}(m, \pi) \rightarrow m'$: This oracle sends msg messages to $\boldsymbol{T}$ for π protocol session and outputs the responding message msg' by $\boldsymbol{T}$.
- $\mathcal{O}^{Execute}\left(\psi_{\boldsymbol{T}}\right) \rightarrow (\pi, transcript)$: This oracle runs an entire protocol between $\boldsymbol{R}$ and $\boldsymbol{T}$ taking $\psi_{\boldsymbol{T}}$ as an input and outputting a *transcript* that includes all successive messages of π .
- $\mathcal{O}^{Result}(\pi) \rightarrow z$: This oracle takes the protocol instance π of $\boldsymbol{T}$ and outputs $z \in \{0, 1\}$. If $\boldsymbol{T}$ is identified by $\boldsymbol{R}$ in its π during the IDENTTAG protocol $z = 1$. Otherwise, $\boldsymbol{T}$ is invalid and the oracle outputs $z = 0$.

4.2.1 Adversary Classes

VM mainly groups adversaries into four classes that restrict their attack capabilities. An adversary $\boldsymbol{A}$ within each class is only allowed to utilize certain oracles. *STRONG* $\boldsymbol{A}$ (meaning $\boldsymbol{T}$ is a member of *STRONG* adversary class) can freely use all aforementioned oracles without any limitation. *WEAK* $\boldsymbol{A}$ cannot call $\mathcal{O}^{Corrupt}$ oracle but she has permission to reach the others. *DESTRUCTIVE* $\boldsymbol{A}$ cannot utilize any oracle after querying the $\mathcal{O}^{Corrupt}$ oracle. The first query of $\mathcal{O}^{Corrupt}$ destroys the related $\boldsymbol{T}$. Finally, *FORWARD* $\boldsymbol{A}$ cannot reach any oracles except $\mathcal{O}^{Corrupt}$ after her first calling of the $\mathcal{O}^{Corrupt}$ oracle.

4.3 Security of RFID Schemes

We summarize some important security notions for an RFID scheme below.

Definition 4 (*Completeness*). An RFID scheme provides the *Completeness* property if the probability that an IDENTTAG protocol returns with $\perp$ result for each legitimate tags is negligible.

Definition 5 (*Soundness*). An RFID scheme provides the *Soundness* property if the probability that $\boldsymbol{A}$ impersonates a legitimate $\boldsymbol{T}$ is negligible.

4.4 Privacy of RFID Schemes

Vaudenay states that indistinguishability of a tag is closely related to *privacy* notion. In general, *privacy* refers to secretly keeping the relation of the identifier of a tag ID within its protocol messages. Put differently, if an adversary cannot find out any relation between the identifier of a tag *ID* and its obtained protocol instances, the protocol is private. The adversary performs her attack by playing a security game (an experiment) on an RFID scheme to see whether she finds a target RFID tag correctly in two phases; such as attack and analysis phases. First, the adversary uses the related oracles according to her own adversary class and puts forward a hypothesis. Second, she analyzes all pieces of information gathered and returns 1 if her hypothesis is true and 0 otherwise. An RFID scheme provides privacy if the success probability of this adversary is not negligible.

Definition 6 (*Privacy*). Let consider that an adversary $\boldsymbol{A}$ is as a member of the adversary class, where $P \in \{STRONG, DESTRUCTIVE, FORWARD, WEAK\}$ and $pr_S^{\boldsymbol{A}}$ is the probability of $\boldsymbol{A}$ to successfully prove her hypothesis by playing a security game. If $\forall \boldsymbol{A} \in P$, $pr_S^{\boldsymbol{A}}$ is negligible, then an RFID system is *P-private*.

Vaudenay also defines an *untraceability* property related to the notion of privacy. In general, *untraceability* means indistinguishability of two different tags in an RFID scheme. In the RFID literature, the notion of untraceability is categorized into two types: *backward untraceability* and *forward untraceability* [80–82]. Sometimes they are also called *forward secrecy/privacy* and *backward secrecy/privacy*, respectively. Notably, these terms express the opposite meaning of their word meaning. For instance, *backward privacy* means keeping the privacy of an RFID scheme, even if the tags in the scheme had been corrupted in the past. Actually, "*backward*" and "*forward*" terms are originated from the certain time that an adversary can obtain the internal privileges of an RFID tag (i.e. tampering or having ownership transfer) [81]. She is also able to record both a set of backward and forward protocol interactions so that she can destroy the tag privacy.

Let $prb_s^{\boldsymbol{A}}\left(t, \Phi_{t_0}^T\right) \rightarrow y$ be a function that outputs the probability of an adversary $\boldsymbol{A}$ to successfully trace a tag $\boldsymbol{T}$ at time t knowing $\Phi_{t_0}^T$, where $\Phi_{t_0}^T$ denotes the whole internal knowledge (e.g. secret keys and parameters) of $\boldsymbol{T}$ at time t_0 (i.e. A can obtain $\Phi_{t_0}^T$ by corrupting $\boldsymbol{T}$ at time t_0) and $0 \leq y \leq 1$.

Definition 7 (*Backward Untraceability/Forward Privacy*). An RFID scheme satisfies *backward untraceability* property, if $prb_s^A\left(t, \Phi_{t_0}^T\right)$ is negligible, where $t < t_0$.

Definition 8 (*Forward Untraceability/Backward Privacy*). An RFID scheme satisfies *forward untraceability* property, if $prb_s^A\left(t, \Phi_{t_0}^T\right)$ is negligible, where $t > t_0$.

It has been considered that both *backward* and *forward* privacy are the essential security requirements for an RFID authentication scheme. Lim and Kwon [81] introduce forward untraceability property and argue that in general, providing this property for an RFID scheme is harder than accomplishing backward untraceability. They focus on the importance of forward untraceability and state that it is at least as crucial as backward untraceability, for RFID authentication schemes. Moreover, Vaudenay says that only a STRONG adversary can break the forward untraceability of an RFID scheme since she can call other oracles after corrupting the tag. Vaudenay also shows in his paper that the ultimate privacy level for RFID systems can be ensured by using PKC [17].

5 Analysis of Previous Authentication Schemes

In this section, we first briefly introduce four recent and relatively popular RFID protocols, namely ID17 [28], BDD17 [27], DB19 [56] and LZKZ18 [57]. Then, we present that these schemes do not ensure forward and/or backward privacy as they claimed.

5.1 Analysis of ID17 RFID Authentication Scheme

In this subsection, we first briefly describe ID17 [28] and then show our proposed attacks.

5.1.1 The Protocol Description

ID17 scheme (illustrated in Fig. 1) has two phases called initialization and authentication, respectively. In the initialization phase, reader and tag make an agreement on EC. Later on, the reader generates a random number as its private key, k_r' and calculates its public key $k_R' = k_r' P$, where P is base point of EC. Meanwhile, the tag executes a similar process so it possesses the public k_t' and private k_T' keys. At the end of the phase, both reader and tag share their public keys with each other.

In the authentication phase, the reader first randomly generates an ephemeral private key k_r and calculates its own ephemeral public key, where $k_R = k_r P$. Then, the reader signs k_R with its private key k_r' using ECDSA. The signature of k_R is (x, y), where $(x, y) = ECDSA_{k_r'}(k_R)$. After signing, the reader sends k_R and its signature (x, y) to the tag.

When the tag receives the messages of the reader (k_R, x, y), the tag checks that (x, y) are integers in the range $[1, n-1]$. If not, the tag terminates the session. Else, it continues the verification process and verifies k_R using the public key of the reader k_R'. If the verification is succeeded, the tag authenticates the reader. Otherwise, it rejects the session. In case of authentication, the tag also randomly picks k_t as an ephemeral private key and computes its own ephemeral public key, $k_{TT} = k_t P$. Then, the tag signs k_{TT} and gets the signature pair (w, v) using ECDSA, where $[w, v] = ECDSA_{k_t'}(k_{TT})$. After signing, the tag calculates $K_{TR} = k_t k_R$ as an ephemeral shared secret key. Later on, the tag encrypts its ID with K_{TR} and obtains message C, where $C = AES_{K_{TR}}(ID)$. The tag sends k_{TT}, w, v, C to the reader.

Upon receiving the tag's response, the reader also checks that w, v are integers in the range $[1, n-1]$. If not, the reader drops the session. Else, the reader continues the

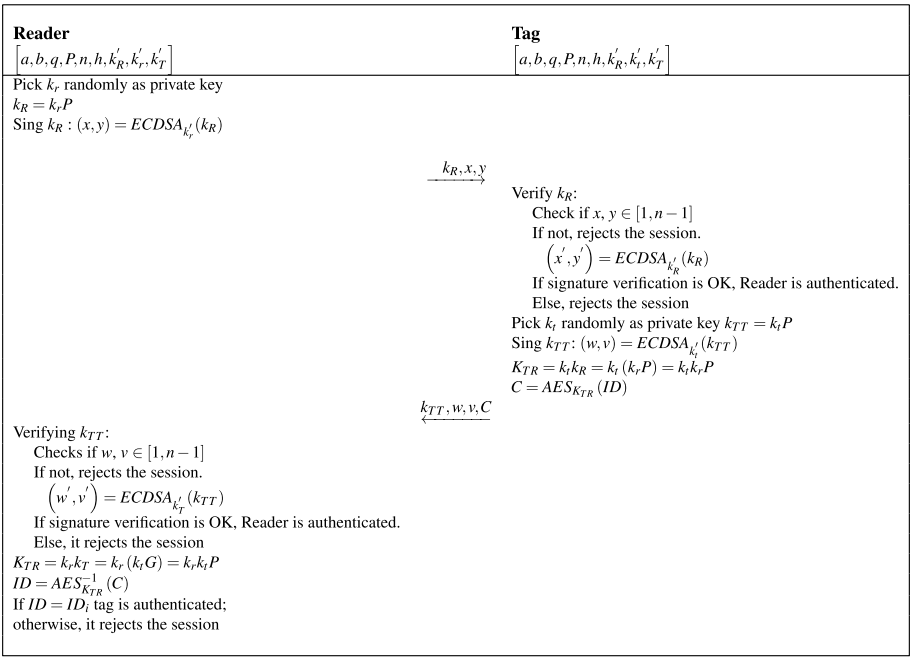

Fig. 1 ID17 RFID authentication scheme proposed in [28]

verification process and verifies k_{TT}. If the verification is succeeded, the reader also computes the ephemeral shared secret key, where $K_{TR} = k_r k_{TT}$. The reader decrypts C using K_{TR} and obtains the ID of the tag. If the reader finds that the ID belongs to the tag registered in the database, the tag is authenticated, too. This scheme is shown in Fig. 1.

5.1.2 Proposed Attacks on the Protocol

The authors claim that their protocol (ID17) provides forward and backward security but we prove that when an adversary corrupts a tag, she can distinguish the tag among the others using its past and future transactions [28]. The authors, in their analysis, state that an adversary cannot perform these attacks because all the transmitted messages are updated for each protocol session. We show that their design does not fulfill randomization in each session to prevent the untraceability since the adversary can verify every signature of the tag if she obtains the private key of the tag once. Therefore, the adversary can violate the backward and forward privacy. Formally, the adversary plays the following games to show how to break the forward and backward privacy properties.

Theorem 1 *ID17 scheme does not provide backward privacy.*

Proof : Let $\boldsymbol{A}$ be a STRONG adversary that plays a security game as below.

1. $\boldsymbol{A}$ calls $\mathcal{O}^{CreateTag}(ID_0, 1)$ and $\mathcal{O}^{CreateTag}(ID_1, 1)$ to create two valid tags T_0 and T_1, respectively.
2. $\boldsymbol{A}$ randomly picks one tag T_i by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseu-donym ψ_{T_i}, where $i \in_R \{0, 1\}$.
3. $\boldsymbol{A}$ chooses a time interval I_0. During I_0,she calls $\mathcal{O}^{Corrupt}\left(\psi_{T_i}\right)$ and obtains the internal values of the tag with pseudonym ψ_{T_i}. These are $a, b, q, P, n, h, k'_{TT_i}, k_{t'_i}, ID_i, k_{R'_i}$.
4. $\boldsymbol{A}$ frees the tag by calling $\mathcal{O}^{Free}\left(\psi_{T_i}\right)$ oracle.
5. $\boldsymbol{A}$ chooses another time interval I_1, where $I_1 > I_0$. During I_1, $\boldsymbol{A}$ calls$\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and receives two pseudonyms ψ_{T_0} and ψ_{T_1}.
6. $\boldsymbol{A}$ arbitrarily selects one of the drawn tags $\left(\text{e.g. } \psi_{T_1}\right)$ and calls $\mathcal{O}^{Execute}\left(\psi_{T_1}\right)$ oracle. She gets $k_{R_1}^{I_1}, x_1^{I_1}, y_1^{I_1}, k_{TT_1}^{I_1}, w_1^{I_1}, v_1^{I_1}, C_1^{I_1}$ as a protocol transcript.
7. $\boldsymbol{A}$ frees the tags by calling $\mathcal{O}^{Free}\left(\psi_{T_0}\right)$ and $\mathcal{O}^{Free}\left(\psi_{T_1}\right)$ oracle.
8. Then, $\boldsymbol{A}$ tries to verify the signature $\left(w_1^{I_1}, v_1^{I_1}\right)$ of the ephemeral public key $k_{TT_1}^{I_1}$ of the tag with the pseudonym ψ_{T_1} by using the corrupted static key k'_{TT_i} of the tag.
9. If the signature is valid, she claims that $i = 1\left(\psi_{T_i} = \psi_{T_1}\right)$. Otherwise, she claims that $i = 0\left(\psi_{T_i} = \psi_{T_0}\right)$.

Obviously, the success probability of this adversary is 1 and she wins the game. This means that $\boldsymbol{A}$ can distinguish the future transactions of the tag. Therefore, this scheme does not provide backward privacy. □

Theorem 2 *ID17 scheme does not provide forward privacy.*

Proof Let $\boldsymbol{A}$ be a STRONG adversary that plays a security game as below.

1. $\boldsymbol{A}$ calls $\mathcal{O}^{CreateTag}(ID_0, 1)$ and $\mathcal{O}^{CreateTag}(ID_1, 1)$ to create two valid tags T_0 and T_1, respectively.
2. $\boldsymbol{A}$ randomly picks two tags by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and gets two pseudonyms ψ_{T_0} and ψ_{T_1}.
3. $\boldsymbol{A}$ chooses a time interval I_0. During I_0, she arbitrarily selects one of the drawn tags $(e.g.\, \psi_{T_1})$ and calls $\mathcal{O}^{Execute}(\psi_{T_1})$ oracle. Then, $\boldsymbol{A}$ gets $k_{R_1}^{I_0}, z_1^{I_0}, s_1^{I_0}, k_{TT_1}^{I_0}, g_1^{I_0}, h_1^{I_0}, C_1^{I_0}$ as a protocol transcript for ψ_{T_1}.
4. $\boldsymbol{A}$ frees the tags by calling $\mathcal{O}^{Free}(\psi_{T_0})$ and $\mathcal{O}^{Free}(\psi_{T_1})$ oracle.
5. $\boldsymbol{A}$ chooses another time interval I_1, where $I_1 > I_0$. During I_1, $\boldsymbol{A}$ randomly chooses a tag T_i by calling$\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseudonym ψ_{T_i}, where $i \in_R \{0, 1\}$.
6. $\boldsymbol{A}$ calls $\mathcal{O}^{Corrupt}(\psi_{T_i})$ oracle and gets $a, b, q, P, n, h, k'_{TT_i}, k_{t'_i}, ID_i$ and $k_{R'_i}$
7. $\boldsymbol{A}$ frees the tag by calling $\mathcal{O}^{Free}(\psi_{T_i})$ oracle.
8. Then, $\boldsymbol{A}$ tries to verify the signature $\left(w_1^{I_0}, v_1^{I_0}\right)$ of the ephemeral public key $k_{TT_1}^{I_0}$ of the tag with the pseudonym ψ_{T_1} by using the corrupted static key k'_{TT_i} of the tag.
9. If the signature is valid, she claims that $i = 1\,(\psi_{T_i} = \psi_{T_1})$. Otherwise, she claims that $i = 0\,(\psi_{T_i} = \psi_{T_0})$.

The success probability of this adversary is 1 and she wins the game. This means that $\boldsymbol{A}$ has stored some past transcripts. Then, when she obtains the internal values of the tag, thereby she can verify the signature of the ephemeral public key and identify the tag using a previous transcript. Therefore, this scheme does not provide forward privacy. □

5.2 Analysis of BDD17 RFID Authentication Scheme

Tn this subsection, we first briefly describe the BDD17 [27] scheme and then show our proposed attacks.

5.2.1 The Protocol Description

BDD17 scheme shown in Fig. 2 has three phases: setup phase, authentication phase, and update phase. In the setup phase, a trusted issuer generates the system parameters $< Z_{BS_j}, ID_{T_i}, x_{T_i}, SID_j, P_s, m >$, $< Z_{BS_j}, x_{R_i}, SID_j, P_s, V_k, W_k >$ and $< Z_{T_i}, Z_{R'_i}, ID_{T_i}, RID_{i'}, SID_j, x_{BS_j}, x_{R_{i'}}, P_s, m, IDs_{T_i}^{old}, IDs_{T_i}^{new} >$ to be stored by all involved entities (tags, readers and the back-end server, respectively).

In the mutual authentication and an updating phase, reader ($R_{i'}$) controls the user's password and checks whether $V'_k = V_k$. If it is held, then $R_{i'}$ generates a random number r_R and broadcast the request (r_R, *auth*) to tag T_i. When the tag receives the request, it signs r_R with a pre-shared message m and a random scalar k using elliptic curve message recovering signature algorithm (ECMR). Then, the tag responds with an anonymous identity $ID_{s_{T_i}}$ and an ECMR signature (r, s). Upon receiving this response, the reader gets the current timestamp T_{r_1} and computes the message $V = h\left(x_{R_{i'}} \| r_R \| T_{r_1}\right)$. Then, the reader sends $r, s, r_R, V, T_{r_1}, ID_{s_{T_i}}$ to BS_j. BS_j firstly checks the validity of the timestamp and authenticates the reader checking the value V. BS_j finds the related tag's parameter using $ID_{s_{T_i}}$ in $O(1)$ time. Then, BS_j recovers message m' and verifies its validity by calculating $\gamma = h(r \| r_R)\left(Z_{T_i} + \left(\left(Z_{T_i}\right)_x + ID_{T_i}\right)P_s\right)$ and $\left(r_R \oplus m'\right) = r - \left((sG + \gamma)x_{BS_j}\right)_x mod(n)$.

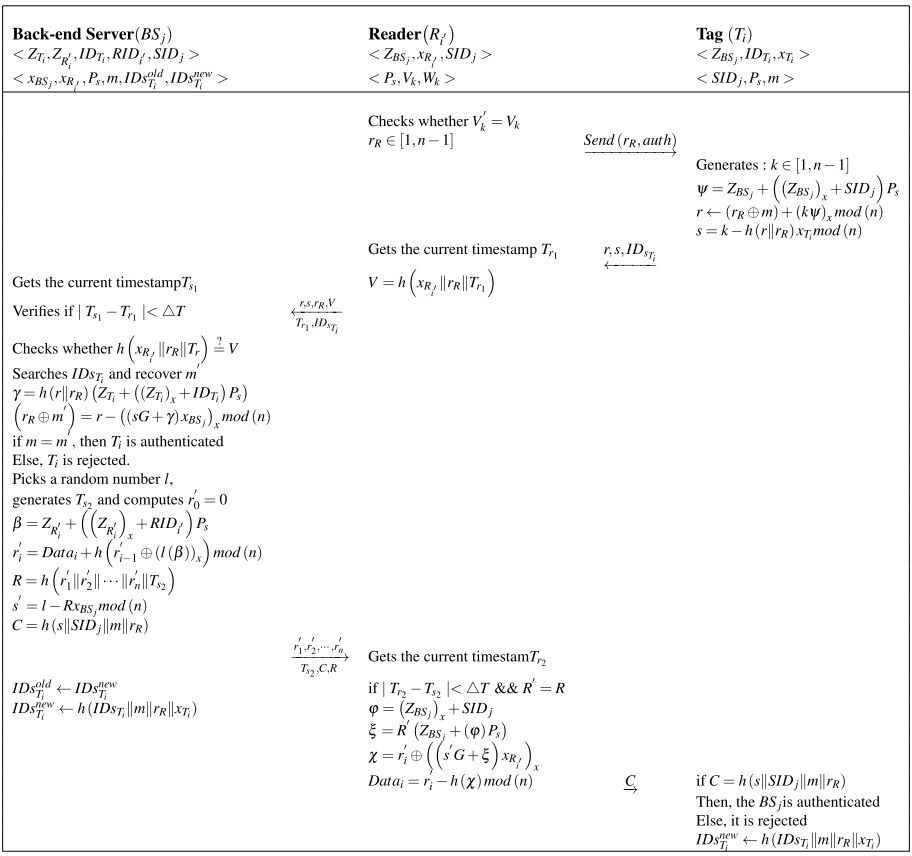

Fig. 2 BDD17 RFID authentication scheme proposed in [27]

If the signature is not correct, it rejects T_i. Otherwise, BS_j server gets the current timestamp T_{s_2} and performs the following calculations: $\beta = Z_{R_i'} + \left(\left(Z_{R_i'}\right)_x + RID_{i'}\right)P_s$, $r_i^{'} = Data_i + h\left(r_{i-1}^{'} \oplus (l(\beta))_x\right)\ mod(n)$, $R = h\left(r_1^{'} \| r_2^{'} \| \cdots \| r_n^{'} \| T_{s_2}\right)$, $s^{'} = l - R x_{BS_j} mod(n)$ and $C = h\left(s \| SID_j \| m \| r_R\right)$.

After BS_j sending the message $(C, R, r_1^{'}, r_2^{'}, \ldots, r_n^{'}, T_{s_2})$ to the reader, it updates $IDs_{T_i}^{new} \leftarrow h\left(IDs_{T_i} \| m \| r_R \| x_{T_i}\right)$. When $R_{i'}$ receives the response of the server, it firstly verifies the validity of the timestamp, $\mid T_{r_2} - T_{r_1} \mid < \bigtriangleup T$. It also verifies the validity and integrity of the transmitted message by calculating: $\varphi = \left(Z_{BS_j}\right)_x + SID_j$, $\xi = R^{'}\left(Z_{BS_j} + (\varphi) P_s\right)$, $\chi = r_i^{'} \oplus \left(\left(s^{'} G + \xi\right) x_{R_{i'}}\right)_x$ and $Data_i = r_i^{'} - h(\chi) mod(n)$. If the verifications are succeeded, then the reader $R_{i'}$ relays the message C to the tag T_i for mutual authentication. When the tag receives C, it checks $C = h\left(s \| SID_j \| m \| r_R\right)$. If succeeded, T_i authenticates BS_j; else, it rejects. Finally, T_i updates its pseudonym $IDs_{T_i}^{new} \leftarrow h\left(IDs_{T_i} \| m \| r_R \| x_{T_i}\right)$ and terminates the session.

5.2.2 Proposed Attacks on the Protocol

The authors claim that their protocol provides the forward security but we prove that when an adversary corrupts a tag, she can distinguish backward and forward transactions of the tag and destroy its privacy [27]. The authors, in their analysis, state that even if an attacker discovers the tag secret parameters, she cannot track the tag's past positions because she does not reach the timestamps and random values. However, we show that their scheme does not provide backward and forward privacy since an adversary can check the updates of the anonymous identifier IDs_{T_i} and break the tag's privacy. Formally, the adversary can perform the following attack.

Theorem 3 *BDD17 protocol does not provide backward privacy.*

Proof Let $\boldsymbol{A}$ be a STRONG adversary that plays a security game as below.

1. $\boldsymbol{A}$ calls $\mathcal{O}^{CreateTag}(ID^0_{T_0}, 1)$ and $\mathcal{O}^{CreateTag}(ID^0_{T_1}, 1)$ to create two valid tags T_0 and T_1 with initial identifiers (the tags update their own identifier after authenticating the reader.), respectively.
2. $\boldsymbol{A}$ randomly picks one tag T_i by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseu-donym ψ_{T_i}, where $i \in_R \{0, 1\}$.
3. $\boldsymbol{A}$ chooses a time interval I_0. During I_0, she calls a $\mathcal{O}^{Corrupt}\left(\psi_{T_i}\right)$ oracle and gets $< Z^{I_0}_{BS_j}, ID^{I_0}_{\psi_{T_i}}, x^{I_0}_{\psi_{T_i}}, SID^{I_0}_j, P^{I_0}_s, m^{I_0} >$.
4. $\boldsymbol{A}$ frees the tag by calling $\mathcal{O}^{Free}\left(\psi_{T_i}\right)$ oracle.
5. $\boldsymbol{A}$ chooses another time interval I_1, where $I_1 > I_0$. During I_1, $\boldsymbol{A}$ calls$\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and receives two pseudonyms ψ_{T_0} and ψ_{T_1}.
6. $\boldsymbol{A}$ arbitrarily selects one of the drawn tags $\left(e.g.\ \psi_{T_1}\right)$ and calls $\mathcal{O}^{Execute}\left(\psi_{T_1}\right)$ oracle. She gets $r^{I^1_1}_R, auth^{I^1_1}, r^{I^1_1}, s^{I^1_1}, IDs^{I^1_1}_{\psi_{T_1}}, C^{I^1_1}$as a protocol transcript.
7. $\boldsymbol{A}$ calls $\mathcal{O}^{Execute}\left(\psi_{T_1}\right)$ oracle again and gets $r^{I^2_1}_R, auth^{I^2_1}, r^{I^2_1}, s^{I^2_1}, IDs^{I^2_1}_{\psi_{T_1}}, C^{I^2_1}$.
8. $\boldsymbol{A}$ frees the tags by calling $\mathcal{O}^{Free}\left(\psi_{T_0}\right)$ and $\mathcal{O}^{Free}\left(\psi_{T_1}\right)$ oracle.
9. Then, $\boldsymbol{A}$ tries to verify the message $IDs^{I^2_1}_{\psi_{T_1}}$ for the tag ψ_{T_i} by computing $IDs^{I^2_1}_{\psi_{T_1}} \stackrel{?}{=} h\left(IDs^{I^1_1}_{\psi_{T_1}} \| m^{I_0} \| r^{I^2_1} \| x^{I_0}_{\psi_{T_i}}\right)$.
10. If the verification is succeeded, she claims that $i = 1\left(\psi_{T_i} = \psi_{T_1}\right)$. Otherwise, she claims that $i = 0\left(\psi_{T_i} = \psi_{T_0}\right)$.

The success probability of this adversary is 1 and she wins the game. This means that $\boldsymbol{A}$ can distinguish the future interactions of T_i checking the updates of the anonymous identifier IDs_{T_i}. Therefore, this scheme does not provide backward privacy. □

Theorem 4 *BDD17 protocol does not provide forward privacy.*

Proof Let $\boldsymbol{A}$ be a STRONG adversary that plays a security game as below.

1. $\boldsymbol{A}$ calls $\mathcal{O}^{CreateTag}(ID^0_{T_0}, 1)$ and $\mathcal{O}^{CreateTag}(ID^0_{T_1}, 1)$ to create two valid tags T_0 and T_1 with initial identifiers (the tags update their own identifier after authenticating the reader), respectively.
2. $\boldsymbol{A}$ randomly picks two tags by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and gets two pseudonyms ψ_{T_0} and ψ_{T_1}.
3. $\boldsymbol{A}$ chooses a time interval I_0. During I_0, she arbitrarily selects one of the drawn tags $\left(e.g.\, \psi_{T_1}\right)$ and calls $\mathcal{O}^{Execute}\left(\psi_{T_1}\right)$ oracle. Then, $\boldsymbol{A}$ gets $r_R^{I^1_0}, auth^{I^1_0}, r^{I^1_0}, s^{I^1_0}, IDs^{I^1_0}_{\psi_{T_1}}, C^{I^1_0}$ as a protocol transcript for ψ_{T_1}.
4. $\boldsymbol{A}$ calls $\mathcal{O}^{Execute}\left(\psi_{T_1}\right)$ oracle again and gets $r_R^{I^2_0}, auth^{I^2_0}, r^{I^2_0}, s^{I^2_0}, IDs^{I^2_0}_{\psi_{T_1}}, C^{I^2_0}$
5. $\boldsymbol{A}$ frees the tags by calling $\mathcal{O}^{Free}\left(\psi_{T_0}\right)$ and $\mathcal{O}^{Free}\left(\psi_{T_1}\right)$ oracle.
6. $\boldsymbol{A}$ chooses another time interval I_1, where $I_1 > I_0$. During I_1, $\boldsymbol{A}$ randomly chooses a tag T_i by calling$\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseudonym ψ_{T_i}, where $i \in_R \{0, 1\}$.
7. $\boldsymbol{A}$ calls $\mathcal{O}^{Corrupt}\left(\psi_{T_i}\right)$ oracle and gets $< Z^{I_1}_{BS_j}, ID^{I_1}_{\psi_{T_i}}, x^{I_1}_{\psi_{T_i}}, SID^{I_1}_j, P^{I_1}_s, m^{I_1} >$
8. $\boldsymbol{A}$ frees the tag by calling $\mathcal{O}^{Free}\left(\psi_{T_i}\right)$ oracle.
9. Then, $\boldsymbol{A}$ tries to verify the message $IDs^{I^2_0}_{\psi_{T_1}}$ for the tag ψ_{T_i} by computing $IDs^{I^2_0}_{\psi_{T_1}} \stackrel{?}{=} h\left(IDs^{I^1_0}_{\psi_{T_1}} \| m^{I_1} \| r^{I^2_0} \| x^{I_1}_{\psi_{T_i}}\right)$.
10. If the verification is succeeded, she claims that $i = 1\left(\psi_{T_i} = \psi_{T_1}\right)$. Otherwise, she claims that $i = 0\left(\psi_{T_i} = \psi_{T_0}\right)$.

The success probability of this adversary is 1 and she wins the game. This means that $\boldsymbol{A}$ has stored some past transcripts. Then, when she obtains the internal values of the tag, she can identify the tag using previous transcripts by verifying the message $IDs^{new}_{T_i}$. Therefore, this scheme does not provide forward privacy as claimed. □

5.3 Analysis of DB19 RFID Authentication Scheme

In this subsection, we first briefly describe the DB19 [56] scheme and then show our proposed attacks.

5.3.1 The Protocol Description

DB19 scheme (illustrated in Fig. 3) consists of 3 phases: setup phase, authentication phase, and updating phase. Before the authentication, public and private key pairs, ECC domain and some system parameters are securely shared to the readers and the tags in the system. The authentication and updating phases are described below.

Authentication Phase. In this phase, mutual authentication is provided. Firstly, the reader picks a random number r_1, computes R_1 and sends it to the tag. When the tag receives the challenge of the reader, the tag also picks a random number r_2, computes R_2 and sends R_2, and pseudonym IDS back to the reader. When the reader receives the response of the tag, it searches IDS in the database. If the reader does not find it, the reader terminates the protocol. Otherwise, the reader obtains the corresponding identifier (x_t) and key (k) corresponding to IDS^{new} or IDS^{old}. Then, the reader computes $TK_{s_1} = r_1 kR_2$, $TK_{S_2} = x_S kR_2$ and $Auth_s = TK_{s_1} \oplus TK_{s_2} \oplus x_t$. After receiving $Auth_s$, the tag computes $TK_{t_1} = r_2 kR_2$, $TK_{s_2} = x_s kR_2$ and checks if $x'_t = Auth_s \oplus TK_{t_1} \oplus TK_{t_2}$. If the obtained identifier x'_t does not match, the tag terminates the session. Otherwise, the tag authenticates the reader, computes

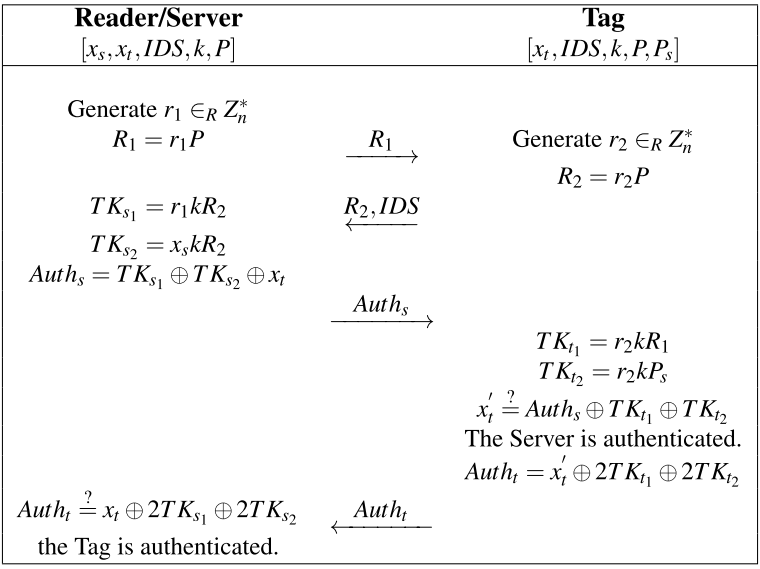

Fig. 3 DB19 RFID authentication scheme proposed in [56]

$Auth_t = x_t^{'} \oplus 2TK_{t_1} \oplus 2TK_{t_2?}$and sends $Auth_t$ to the reader. When the reader receives the message, it checks if $Auth_t = x_t \oplus 2TK_{s_1} \oplus 2TK_{s_2}$. If checking succeeds, the reader authenticates the tag. Otherwise, the reader rejects the $Auth_t$ and terminates the protocol.

Updating Phase. When the authentication is successfully accomplished, both the reader and the tag refresh their secret keys k and the pseudonyms (IDS). The reader also keeps old and new IDS. The tag performs the following updates: $IDS^* = X\left(TK_{t_1}\right) \oplus IDS \oplus k$, $k^* = X\left(TK_{t_2}\right) \oplus 2k$ and $IDS = IDS^*, k = k^*$.

The reader performs the following updates: If IDS^{old} is received, the reader computes $IDS^{new} = X\left(TK_{S_1}\right) \oplus IDS^{old} \oplus k$ and $k^{new} = X\left(TK_{S_2}\right) \oplus 2k^{old}$. If IDS^{new} is received, the reader updates $IDS^{old} = IDS^{new}$ and $k^{old} = k^{new}$. The reader, then, computes $IDS^{new} = X\left(TK_{S_1}\right) \oplus IDS^{old} \oplus k$ and $k^{new} = X\left(TK_{S_2}\right) \oplus 2k^{old}$

5.3.2 Proposed Attacks on the Protocol

Dinarvand and Barati [56] claim that their protocol (BD17) provides forward privacy. However, they do not mention backward privacy in their paper. In this subsection, we show that their scheme does not achieve backward privacy which is one of the well-known privacy requirements. In other words, we prove that when an adversary obtains the secrets of a tag once, she can distinguish the tag with using its future transactions. An adversary can directly reveal the identifier of the tag x_t with sending P_s to the tag instead of R_1 after obtaining the secrets of the tag. Formally, the adversary plays the following game to show how to break the forward untraceability property.

Theorem 5 *BD17 scheme does not provide backward privacy.*

Proof Let $\boldsymbol{A}$ be a STRONG adversary that plays a security game as below.

1. $\boldsymbol{A}$ calls $\mathcal{O}^{CreateTag}(x_t^{T_0}, 1)$ and $\mathcal{O}^{CreateTag}(x_t^{T_1}, 1)$to create two valid tags T_0 and T_1, respectively, where $x_t^{T_0}$ denotes the identifier of a tag.
2. $\boldsymbol{A}$ randomly picks one tag T_i by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseu-donym ψ_{T_i}, where $i \in_R \{0, 1\}$.
3. $\boldsymbol{A}$ chooses a time interval I_0. During I_0,calls a $\mathcal{O}^{Corrupt}\left(\psi_{T_i}\right)$ oracle. She obtains all internal values of the tag with pseudonym ψ_{T_i}. These are $x_t^{T_i}, IDS_i, k_i, P$ and P_s.
4. $\boldsymbol{A}$ frees the tag by calling $\mathcal{O}^{Free}\left(\psi_{T_i}\right)$ oracle.
5. $\boldsymbol{A}$ chooses another time interval I_1, where $I_1 > I_0$. During I_1, $\boldsymbol{A}$ calls$\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and receives two pseudonyms ψ_{T_0} and ψ_{T_1}.
6. $\boldsymbol{A}$ arbitrarily selects one of the drawn tags $\left(e.g.\ \psi_{T_1}\right)$ and calls $\mathcal{O}^{Launch}()$ oracle. She starts a new protocol execution with π_1
7. $\boldsymbol{A}$ calls $\mathcal{O}^{SendTag}\left(P_s, \pi_1\right)$ oracle and she sends P_s message instead of R_1 message. The tag ψ_{T_1} respones with $R_2^{I_1}, IDS^{I_1}$but $\boldsymbol{A}$ does not need these messages.
8. $\boldsymbol{A}$ sends $x_t^{T_i}$ to the tag instead of $Auth_s^{I_1}$ message (in step-3) by calling $\mathcal{O}^{SendTag}(x_t^{T_i}, \pi_1)$ oracle and waits for the response of the tag.
9. If the tag ψ_{T_1} responds with $Auth_t^{I_1}$, $\boldsymbol{A}$ directly gets $x_t^{T_1}$ and claims that $i = 1\left(\psi_{T_i} = \psi_{T_1}\right)$, since the response means that the tag authenticates $\boldsymbol{A}$. In fact, $Auth_t^{I_1} = x_t^{T_1}$ because of $Auth_t^{I_1} = x_t^{T_1} \oplus 2\left(r_2 k_1 P_s\right) \oplus 2\left(r_2 k_1 P_s\right)$.
10. If the tag ψ_{T_1} does not respond, this means that the tag does not authenticate and terminates the session. Therefore, $\boldsymbol{A}$ claims that $i = 0\left(\psi_{T_i} = \psi_{T_0}\right)$.

Obviously, the success probability of this adversary is 1 and she wins the game. Therefore, BD17's scheme does not provide forward untraceability property. □

5.4 Analysis of LZKZ18 RFID Authentication Scheme

In this subsection, we first describe LZKZ18 [57] scheme and then show our proposed attacks.

5.4.1 The Protocol Description

LZKZ18 scheme (illustrated in Fig. 4) includes two processes: a setup process and an implementation process. In the setup process which is also divided into initialization and bidirectional authentication phases, the server and the reader securely share and store the needed keys. The reader, server and tag agree on the ECC domain parameters, too.

In the implementation process, at first, the reader picks a random x_R, computes R_1 and initiates a new protocol session sending the query request *Query* and R_1. When the tag receives a request, the tag picks a random x_T and computes T_2 and T_3. The tag sends T_1, T_2 and T_3 to the reader. When the reader receives the response of the tag, the reader computes R_2 and checks if $R_2 \stackrel{?}{=} T_2$. If the checking is false, the authentication fails and the session drops. Otherwise, the reader considers that the tag is legitimate and computes R_3 and R_4. Then, the reader sends R_1, R_2, R_3, T_1, T_3 and t_R to the server. After receiving the message, the server firstly checks the timestamp t_R. If the t_R exceed the time limit, the server finishes the authentication. Otherwise, the server picks a random number x_S and computes S_1 and S_2. If $S_2 \neq R_3$, then the authentication fails. Otherwise, the server authenticates the reader and calculates S_3. If $S_3 \neq R_D$, then the authentication fails again. Otherwise, the server obtains the reader's authorization identifier and computes S_4 and checks if $S_4 \stackrel{?}{=} T_D$.

Server $[T_D, R_D, k_{AB}, k_{AC}, a, P_S = aP]$		**Reader** $[R_D, k_{AB}, b, P_R = bP]$		**Tag** $[T_D, k_{AC}, c, P_R = cP, P_S]$
		$x_R \in_R Z_q$	(1) $\xrightarrow{Query, R_1}$	$x_T \in_R Z_q$
		$R_1 = x_R P$		$T_1 = x_T P$
				$T_2 = H(x_T R_1)$
$x_S \in_R Z_q$, $S_1 = x_S P$		$R_2 = H(x_R T_1)$	(2) $\xleftarrow{T_1, T_2, T_3}$	$T_3 = T_D + (x_T + c) P_S$
$S_2 = H(R_1 \| k_{AB} \| t_R)$	(3) $\xleftarrow[T_1, T_3, t_R]{R_1, R_3, R_4}$	Judge: $R_2 \stackrel{?}{=} T_2$		
Judge: $S_2 \stackrel{?}{=} R_3$		$R_3 = H(R_1 \| k_{AB} \| t_R)$		
$S_3 = R_4 - aR_1 - k_{AB}$		$R_4 = R_D + (x_R + b) P_S$		
Judge: $S_3 \stackrel{?}{=} R_D$				
$S_4 = T_3 - aT_1 - k_{AC}$				
Judge: $S_4 \stackrel{?}{=} T_D$				
$S_5 = x_S R_1 + k_{AB}$				
$S_6 = x_S T_1 + k_{AC}$	(4) $\xrightarrow{S_1, S_5, S_6}$	$R_5 = x_R S_1 + k_{AB}$	(5) $\xrightarrow{S_1, S_6}$	$T_4 = x_T S_1 + k_{AC}$
		Judge: $R_5 \stackrel{?}{=} S_5$		Judge: $T_4 \stackrel{?}{=} S_6$

Fig. 4 LZKZ18 RFID authentication scheme proposed in [57]

If $S_4 \neq T_D$, then the authentication fails. Otherwise, the server obtains the tag's authorization identifier and calculates S_5 and S_6. Later on, the server sends S_1, S_5 and S_6 to the reader. When the reader gets this, it computes R_5 and checks if $R_5 \stackrel{?}{=} S_5$. If $R_5 = S_5$, the reader authenticates the server. Otherwise, the authentication fails. After the successful authentication, the reader sends S_1, S_6 to the tag. The tag then computes T_4. Finally, the tag checks if $T_4 \stackrel{?}{=} S_6$. If $T_4 \neq S_6$, the tag rejects the authentication and terminates the session. Otherwise, the tag authenticates the reader and the back-end server, too.

5.4.2 Proposed Attacks on the Protocol

The authors claim that LZKZ18 protocol provides forward security without presenting any analysis [57]. In this paper, we show that their scheme does not fulfill both backward and forward privacy because an adversary can destroy the privacy of a tag by sending P_s instead of R_1 and checking if $T_2 \stackrel{?}{=} H(T_3 - T_D - cP_S)$.

Theorem 6 *LZKZ18 scheme does not provide backward privacy.*

Proof Let $\boldsymbol{A}$ is a STRONG adversary that plays a security game as below.

1. $\boldsymbol{A}$ calls $\mathcal{O}^{CreateTag}(T_{D_0}, 1)$ and $\mathcal{O}^{CreateTag}(T_{D_1}, 1)$ to create two valid tags T_0 and T_1, respectively.
2. $\boldsymbol{A}$ randomly picks one tag T_i by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseu-donym ψ_{T_i}, where $i \in_R \{0, 1\}$.
3. $\boldsymbol{A}$ chooses a time interval I_0. During I_0,calls a $\mathcal{O}^{Corrupt}(\psi_{T_i})$ oracle. She obtains all internal values of the tag with pseudonym ψ_{T_i}. These are $T_{D_i}, k_{AC}, c_i, P_{T_i}$ and P_S.
4. $\boldsymbol{A}$ frees the tag by calling $\mathcal{O}^{Free}(\psi_{T_i})$ oracle.
5. $\boldsymbol{A}$ chooses another time interval I_1, where $I_1 > I_0$. During I_1, $\boldsymbol{A}$ calls$\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and receives two pseudonyms ψ_{T_0} and ψ_{T_1}.
6. $\boldsymbol{A}$ arbitrarily selects one of the drawn tags $(e.g.\ \psi_{T_1})$ and calls $\mathcal{O}^{Launch}()$ oracle. She starts a new protocol execution with π_1

7. $\boldsymbol{A}$ calls $\mathcal{O}^{SendTag}\left(P_S, \pi_1\right)$ oracle so she sends P_S message instead of R_1 message. The tag ψ_{T_1} responses with $T_1^{I_1}, T_2^{I_1}, T_3^{I_1}$.
8. $\boldsymbol{A}$ checks $T_2^{I_1} \stackrel{?}{=} H\left(T_3^{I_1} - T_{D_i} - c_i P_S\right)$. If succeeds, $\boldsymbol{A}$ claims that $i = 1\left(\psi_{T_i} = \psi_{T_1}\right)$. Otherwise, $\boldsymbol{A}$ claims that $i = 0\left(\psi_{T_i} = \psi_{T_0}\right)$.

Obviously, the success probability of this adversary is 1 and she wins the game. Therefore, LZKZ's scheme does not ensure backward privacy. □

Theorem 7 *LZKZ18 scheme does not provide forward privacy.*

Proof Let $\boldsymbol{A}$ be a STRONG adversary that plays a security game as below.

1. $\boldsymbol{A}$ calls $\mathcal{O}^{CreateTag}(T_{D_0}, 1)$ and $\mathcal{O}^{CreateTag}(T_{D_1}, 1)$ to create two valid tags T_0 and T_1, respectively.
2. $\boldsymbol{A}$ randomly picks two tags by querying $\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 2\right)$ oracle and gets two pseudonyms ψ_{T_0} and ψ_{T_1}.
3. $\boldsymbol{A}$ chooses a time interval I_0. During I_0, she arbitrarily selects one of the drawn tags $\left(e.g.\, \psi_{T_1}\right)$ and calls $\mathcal{O}^{Launch}()$ oracle. $\boldsymbol{A}$ starts a new protocol execution with π_1
4. $\boldsymbol{A}$ calls $\mathcal{O}^{SendTag}\left(P_S, \pi_1\right)$ oracle so she sends P_S message instead of R_1message. The tag ψ_{T_1} responses with $T_1^{I_0}, T_2^{I_0}, T_3^{I_0}$.
5. Then, $\boldsymbol{A}$ frees the tags by calling $\mathcal{O}^{Free}\left(\psi_{T_0}\right)$ and $\mathcal{O}^{Free}\left(\psi_{T_1}\right)$ oracle.
6. $\boldsymbol{A}$ chooses another time interval I_1, where $I_1 > I_0$. During I_1, $\boldsymbol{A}$ randomly chooses a tag T_i by calling$\mathcal{O}^{DrawTag}\left(\frac{1}{2}, 1\right)$ oracle and gets a pseudonym ψ_{T_i}, where $i \in_R \{0, 1\}$.
7. $\boldsymbol{A}$ calls $\mathcal{O}^{Corrupt}\left(\psi_{T_i}\right)$ oracle and gets $T_{D_i}, k_{AC}, c_i, P_{T_i}$ and P_S.
8. $\boldsymbol{A}$ frees the tag by calling $\mathcal{O}^{Free}\left(\psi_{T_i}\right)$ oracle.
9. *Then,* $\boldsymbol{A}$ checks if $T_2^{I_0} \stackrel{?}{=} H\left(T_3^{I_0} - T_{D_i} - c_i P_S\right)$. If succeeds, $\boldsymbol{A}$ claims that $i = 1\left(\psi_{T_i} = \psi_{T_1}\right)$. Otherwise, $\boldsymbol{A}$ claims that $i = 0\left(\psi_{T_i} = \psi_{T_0}\right)$.

Obviously, the success probability of this adversary is 1 and she wins the game. Therefore, LZKZ's scheme does not provide forward privacy. □

6 Our Improved Protocol

We propose a new privacy-friendly ECC based RFID authentication protocol depicted in Fig. 5 by enhancing ID17 scheme [28]. Our focus is to overcome the privacy weaknesses of their protocol. We consider that transmitting the ephemeral public key and its signature in an insecure channel causes privacy issues. Therefore, we claim that if an ephemeral public key is broadcasted with an indistinguishable encrypted signature, then an attacker cannot track the past and future interactions of any tag so that both forward and backward untraceability properties are provided.

We consider that both the reader and the back-end server (BS) are trusted entities but a tag might be corrupted, compromised or illegitimate. For the sake of simplicity, we also consider both BS and reader as a single entity and the tag is the second entity of our scheme. Note that this does not affect the generality since most of the applications accept that the communication of tag−reader is not secure but the communication of reader−BS

is secure (as shown in Fig. 6). Before describing the protocol, we present the following notations in Table 2 to improve the intelligibility.

6.1 Protocol Description

We present a brief description of our scheme below. Figure 5 also elaborately shows the details. Our proposed protocol consists of a setup phase and an authentication phase.

6.1.1 Setup Phase

Reader and tags must agree on ECC domain parameters of the scheme to use elliptic curve cryptosystem. Hence, in the setup, both tags and readers firstly agree on a curve with ECC domain parameters. In our scheme, we prefer ECC brainpoolP160r1, a standard curve, to be used for the domain parameter values [72]. In this phase, all unique identifiers ID_i of the tags are stored in BS. An integer $k_t^{'}$ is randomly chosen as the private key of the tag, where $1 \leq k_t^{'} \leq n-1$ and its public key is computed as $k_T^{'} = k_t^{'}G$. Then, the key pairs are stored in the tag. $k_T^{'}$ is shared with the reader. On the other hand, an integer $k_r^{'}$ is randomly chosen as the private key of the reader, where $1 \leq k_r^{'} \leq n-1$ and its public key is computed as $k_R^{'} = k_r^{'}G$. Then, the key pairs are stored in BS, while $k_R^{'}$ is shared with all tags.

6.1.2 Authentication Phase

In this phase, the mutual authentication is accomplished in two rounds with the following steps. Note that Fig. 5 also depicts each step of our protocol execution.

Step-1: First, the reader randomly generates an ephemeral private key k_r and calculates its own ephemeral public key, where $k_R = k_rG$.
Step-2: The reader signs k_R with its private key $k_r^{'}$ using ECDSA, where $(z, s) = ECDSA_{k_r^{'}}(k_R)$
Step-3: The reader sends k_R and the signature (z, s) to the tag.
Step-4: The tag firstly verifies k_R using the public key of the reader $k_R^{'}$.
Step-5: If the verification is succeeded, the tag will authenticate the reader. Otherwise, it rejects the session. In case of authentication, the tag also randomly picks

Table 2 Notations of our proposed protocol

p, a, b, G, n, h	ECC domain parameters
$k_R^{'}, k_r^{'}$	Static key pairs (public, private) of the reader
$k_T^{'}, k_t^{'}$	Static key pairs (public, private) of the tag
ID_i	Unique identifier of i^{th} tag
k_R, k_r	Ephemeral key pairs (public, private) of the reader
(z, s)	Signature of the ephemeral public key of the reader
k_{TT}, k_t	Ephemeral key pairs (public, private) of the tag
(g, f)	Signature of the ephemeral public key of the tag
k_{TR}	Established ephemeral shared secret key after ECDH key agreement protocol
$Hash(.)$	A secure cryptographic hash function

Reader $\left[a,b,p,G,n,h,k_R^{'},k_r^{'},k_T^{'}\right]$		**Tag** $\left[a,b,p,G,n,h,k_R^{'},k_t^{'},k_T^{'},ID\right]$
Pick k_r randomly as private key (1) $k_R = k_r G$ (2) Sing k_R: $\{(z,s) = sign(k_R)\}$ (2.1) $e = Hash(k_R)$ (2.2) Select k randomly (2.3) $z = x_1 mod\,(n)$; if $z = 0$ then go to (2.2) (2.4) $s = k^{-1}\left(e + k_r^{'} z\right) mod(n)$; if $s = 0$ then go to (2.2)		
	(3) $\xrightarrow{k_R, z, s}$	
		(4) Verifying k_R (4.1) Check if $z,s \in [1,n-1]$ If not, rejects the session. (4.2) $e = Hash(k_R)$ (4.3) $w = s^{-1}\ mod\,(n)$ (4.4) $u_1 = ew\ mod\,(n)$ and $u_2 = zw\ mod\,(n)$ (4.5)$(x_1, x_2) = u_1 G + u_2 k_R^{'}$ (4.6) If $x_1 \stackrel{?}{=} z\ mod\,(n)$, Reader is authenticated; else, it is not and rejects the session (5) Pick k_t randomly as private key $k_{TT} = k_t G$ (6) Sing k_{TT}: $\{(g,f) = sign(k_{TT})\}$ (6.1) $e = Hash(k_{TT})$ (6.2) Select γ randomly (6.3) $g = x_2 mod\,(n)$; if $g = 0$ then go to (6.2) (6.4) $f = \gamma^{-1}\left(e + k_t^{'} g\right) mod(n)$; if $f = 0$ then go to (6.2) (7) $K_{TR} = k_t k_R = k_t\,(k_r G) = k_t k_r G$ (8) $C = AES_{K_{TR}}\,(ID \| g \| f)$
	(9) $\xleftarrow{k_{TT}, C}$	
(10) $K_{TR} = k_r k_{TT} = k_r\,(k_t G) = k_r k_t G$ (11) $[ID \| g \| f] = AES_{K_{TR}}^{-1}\,(C)$ (12) Verifying k_{TT} (12.1) Check if $g,f \in [1,n-1]$ If not, rejects the session. (12.2) $e = Hash(k_{TT})$ (12.3) $i = f^{-1}\ mod\,(n)$ (12.4) $j_1 = ei\ mod\,(n)$ and $j_2 = gi\ mod\,(n)$ (12.5)$(x_2, y_2) = j_1 G + j_2 k_T^{'}$ (12.6) If $x_2 \neq g\ mod\,(n)$, reader rejects the session. (13) If $x_2 \stackrel{?}{=} g\ mod\,(n)$ and $ID = ID_i$ tag is authenticated; else, it is not and rejects the session		

Fig. 5 Our proposed scheme

k_t as an ephemeral private key and computes its own ephemeral public key, $k_{TT} = k_t G$.

Step-6: The tag signs k_{TT} and gets its signature (g, f) by computing $(g,h) = ECDSA_{k_t^{'}}\left(k_{TT}\right)$.

Step-7: The tag calculates $K_{TR} = k_t k_R$ as an ephemeral shared secret key.

Step-8: The tag encrypts ID and the signature (g, f) together as a plaintext using the ephemeral key K_{TR}, where $C = AES_{K_{TR}}(ID \| g \| f)$.

Step-9: The tag transmits only C and k_{TT} messages to the reader.

Step-10: When the reader receives the message of the tag, the reader computes the ephemeral shared secret key $K_{TR} = k_r k_{TT}$, where $k_r k_{TT} = k_r(k_t G) = k_r k_t G$.

Step-11: Then, the reader can meaningfully decrypt message C using K_{TR}, obtain ID and signature (g, f) if the shared key is valid. Otherwise, the reader has a garbage message.

Step-12: The reader verifies the decrypted messages. It checks if g and f are integers in the range $[1, n-1]$. If not, it rejects the session. After that, the reader also verifies the k_{TT} with using the decrypted signature (g, f). If the verification is not succeeded, it rejects the session.

Step-13: The reader checks if $x_2 \stackrel{?}{=} g \; mod(n)$ and searches if the ID belongs to a tag registered in the database ($ID = ID_i$), the tag is authenticated. Otherwise, the reader rejects the session.

7 Security Analysis of Our Proposed Protocol

In this section, we give the security and privacy analysis of our proposed protocol and prove that our scheme provides all essential security and privacy properties.

Theorem 8 *Our protocol provides confidentiality.*

Proof In our protocol, the sensitive information is the identity of tag ID and the private keys of the reader and the tag. The private keys are protected well and are not transmitted. Furthermore, ID is transmitted as ciphertext encrypted by AES. The key of AES is ephemerally derived using elliptic-curve Diffie–Hellman mechanism by both the reader and the tag. Therefore, an adversary $\boldsymbol{A}$ who collects k_R, z, s, k_{TT} and C transcripts, cannot obtain any confidential information without breaking AES or ECDHE in polynomial time. □

Theorem 9 *Our protocol provides integrity.*

Proof In our protocol, we use the ECDSA signatures that are basically used to provide the integrity of k_R and k_{TT} messages. An adversary $\boldsymbol{A}$ cannot change the content of the protocol transcripts because both the reader and the tag verify the transmitted signatures (z, s) and (g, f). $\boldsymbol{A}$ can modify the transmitted message or forge the related signatures if she solves the elliptic curve discrete logarithm problem (ECDLP) but ECDSA is computationally secure and it is a hard problem for polynomial time attackers. Consequently, the protocol guarantees the integrity of transmitted messages. □

Theorem 10 *Our protocol provides availability.*

Proof In our protocol, the tag identifier ID and the pre-shared keys are securely stored and protected well. Hence, it is not needed to synchronously refresh these values for our scheme. In fact, there is no update mechanism between the tag and reader. Therefore, the protocol can be executed all the time between the reader and the tag. Hence, our scheme provides availability. □

Theorem 11 *Our protocol provides tag anonymity.*

Proof In the authentication phase, the tag responds when it receives challenges from the reader. Hence, anonymity is becoming one of the utmost important and imperative security requirement for privacy. In our protocol, an adversary $\boldsymbol{A}$ collects the only k_R, z, s, k_{TT} and C transcripts and cannot reach the tag identifier ID because $\boldsymbol{A}$ is not able to ECDLP in polynomial time and gain k_{TR}. $\boldsymbol{A}$ also cannot break C without having k_{TR} because AES-128 is considered computationally secure. In fact, Theorem 8 also shows that $\boldsymbol{A}$ can never obtain any confidential information. Moreover, if k_{TT} and C messages were not randomly generated for each session, the adversary can ruin the anonymity. However, all messages of the tag in our scheme are randomized for each protocol session and $\boldsymbol{A}$ cannot even distinguish any tag's messages sent in different sessions. Therefore, the protocol achieves tag's anonymity property and the adversary cannot attain any indicator to point out a tag anymore.

□

Theorem 12 *Our protocol provides mutual authentication.*

Proof Mutual authentication (two-way authentication) is an important property in which both entities in a protocol link authenticate each other. In the authentication phase of our protocol, the reader sends randomly generated k_{TR} and its signature z, s by using ECDSA. The tag can verify k_{TR} using the pre-shared public key of the reader $k_R^{'}$, herewith the reader can be authenticated. Likewise, the reader authenticates the tag after verifying k_{TT}. For this authentication, the reader firstly decrypts C, gets the unique tag identifier ID and the ECDSA signature of k_{TT} which is g, f. Secondly, the reader verifies g, f using the pre-shared public key of the tag $k_T^{'}$. If the verification is successful, the reader finally searches ID in its database. If the reader finds it (matches $ID = ID_i$), the tag is authenticated, too. Therefore, the proposed protocol provides mutual authentication. □

Theorem 13 *Our protocol provides scalability.*

Proof The scalability is a crucial property that reduces the computational cost, searching time of a tag in the database and authentication time. In most cases, the searching time linearly increases proportionally proliferating the registered tags in the database with search complexity $O(N)$, where N is the number of valid tags. In our protocol, the reader decrypts C and gets the ID_i (where $1 \leq i \leq N$). Then, the reader searches the matched entry in the database with search complexity $O(1)$ because each entry ID_i matches only one tag in DB. Therefore, our proposed protocol is scalable.

□

Theorem 14 *Our protocol provides forward privacy.*

Proof Forward security is explained in Definition 7. In our proposed protocol, the reader freshly sends k_R and its signature z, s for each protocol session. The tag also generates a new fresh k_{TT} and C messages. The ephemeral K_{TR} ensures that C is randomized. Because of randomization of all session messages, if a probabilistic polynomial-time (ppt) adversary $\boldsymbol{A}$ corrupts a tag $\boldsymbol{T}$, discloses the secrets $ID, k_t^{'}$ and collects the past protocol transcripts, $\boldsymbol{A}$ can distinguish the corrupted tag and its transactions with a negligible probability. $\boldsymbol{A}$ never gets any advantage to overcome the previous indistinguishable transactions of our scheme. Therefore, our protocol provides backward untraceability property. □

Theorem 15 *Our protocol provides backward privacy.*

Proof As mentioned in the proof of Theorem 14, if the same adversary $\boldsymbol{A}$ collects the future protocol transcripts, $\boldsymbol{A}$ can distinguish the corrupted tag and its transactions with a negligible probability. $\boldsymbol{A}$ never gets any advantage to overcome the future indistinguishable transaction of our scheme. Therefore, our protocol provides forward untraceability property. □

Theorem 16 *Our protocol provides location, traceability privacy and withstands the tracking attack.*

Proof In Theorems 14 and 15, we prove that future and backward untraceability property of our protocol. An adversary $\boldsymbol{A}$ cannot destroy the privacy of a tag $\boldsymbol{T}$, even if $\boldsymbol{A}$ has the secrets of $\boldsymbol{T}$ and the past/future protocol transcripts in related protocols. Hence, $\boldsymbol{A}$ certainly cannot ruin location privacy of $\boldsymbol{T}$ without any confidential information of the tag and track $\boldsymbol{T}$. In other words, untraceability properties imply this result. Therefore, our protocol provides location, traceability privacy and it is resistant against the tracking attack. □

Theorem 17 *Our protocol withstands the tag impersonation and reader spoofing attacks.*

Proof An adversary $\boldsymbol{A}$ can impersonate a tag $\boldsymbol{T}$ only by obtaining ID and $k_t^{'}$ but solving ECDLP is computationally infeasible in polynomial time. Hence, $\boldsymbol{A}$ cannot impersonate $\boldsymbol{T}$. Similarly, $\boldsymbol{A}$ can never produce valid C, z, s messages without having K_{TR}, ID and $k_t^{'}$ because of the aforementioned computational infeasibilities. Thus, $\boldsymbol{A}$ cannot spoof the reader. □

Theorem 18 *Our protocol withstands the replay attack.*

Proof In a replay attack, an adversary $\boldsymbol{A}$ imitates a tag $\boldsymbol{A}$ or a reader $\boldsymbol{R}$ by reusing the intercepted past protocol messages. In our proposed protocol, $\boldsymbol{A}$ cannot generate and reuse valid k_R, z, s messages because they are randomly changed for each session. Similarly, $\boldsymbol{A}$ cannot generate and reuse valid k_{TT}, C messages because they are ephemerally generated random transcripts. This attack can be succeeded, only if $\boldsymbol{A}$ reveals the tag secrets $k_t^{'}, ID$ and reader private key $k_r^{'}$. Therefore, the proposed protocol is resistant against the replay attack. □

Theorem 19 *Our protocol withstands the denial-of-service (DoS) and de-synchronization attack.*

Proof We prove that our proposed protocol provides availability in Theorem 10 which shows that both a tag and a reader always remain synchronized during each protocol execution. An adversary cannot desynchronize both entities and execute DoS attack. Thus, the scheme is resistant against the DoS and de-synchronization attack. □

Theorem 20 *Our protocol withstands MiTM attack.*

Proof Our scheme provides mutual authentication property (see Theorem 12). Therefore, our design is resistant to MiTM attack. □

Theorem 21 *Our protocol withstands the cloning attacks .*

Proof In our proposed protocol, each tag has its own identity ID_i and the secret key t'. Even if an adversary can obtain some tags' *IDs* and their private keys, she cannot reach the other tags' *IDs* and their secret keys. Thus, the protocol is resistant to the cloning attack. □

Theorem 22 *Our protocol provides unforgeability.*

Proof In our scheme, only the valid tag and the reader can generate a legitimate signature. An adversary can never perform a forgery attack without having the private keys as their security leans to the hardness of ECDLP. Therefore, the proposed protocol provides unforgeability. □

Theorem 23 *Our protocol withstands modification attack*

Proof According to Theorem 9, our proposed protocol provides integrity property. Therefore, it is resistant to any modification attack. □

8 Comparison and Implementation

In this section, we compare our proposed scheme with other existing ECC-based RFID authentication works in terms of security and performance.

8.1 Security Comparison

We enumerate the security and privacy comparison of our proposed scheme and related protocols in Table 3. It can be obviously seen that our scheme provides all essential security and privacy requirements of an RFID system and is more secure than the previously proposed protocols [27, 28, 36, 45, 49, 53, 56, 57]. Furthermore, we proved in Sect. 5 that the state-of-the-art protocols [27, 28, 56, 57] cannot provide forward and/or backward privacy . Our scheme not only guarantees the related security and privacy requirements but also provides additional properties such as mutual authentication and efficiency in search of the tags during the identification process.

8.2 Performance Comparison

Although security and privacy properties are indispensable for RFID schemes, the performance of these schemes is vital to effectively use RFID systems in real applications. While our priority is proposing an authentication scheme that solves all essential security and privacy issues existing in RFID systems; we target to design an efficient scheme for practical applications. In this section, we first analyze our protocol and compare it to previous prominent ECC-based RFID authentication protocols [27, 28, 36, 45, 49, 53, 56, 57] in terms of computational and communication costs. A detailed performance comparison (including computation and communication costs) in the literature are summarized in Table 4.

Table 3 Security and privacy comparison (✓:provide, x:do not provide, –:not mentioned)

Security and privacy properties	LH14 [45]	Z14 [49]	C14 [36]	ZQ14 [53]	BDD17 [27]	ID17 [28]	DB19 [56]	LZKZ18 [57]	Our Protocol
Mutual authentication	x	✓	x	✓	✓	✓	✓	✓	✓
Confidentiality	x	x	✓	✓	✓	✓	✓	✓	✓
Integrity	–	–	–	–	✓	✓	✓	✓	✓
Availability	✓	✓	✓	✓	✓	✓	✓	✓	✓
Tag anonymity	x	x	x	✓	✓	✓	✓	✓	✓
Location privacy	x	x	x	x	✓	✓	✓	✓	✓
Scalability	✓	✓	✓	✓	✓	✓	✓	✓	✓
Forward privacy	x	x	x	x	x	x	✓	x	✓
Backward privacy	x	x	x	x	x	x	x	x	✓
Tag impersonation att. res.	x	✓	x	✓	✓	✓	✓	✓	✓
Reader spoofing att. res.	x	✓	✓	✓	✓	✓	✓	✓	✓
Replay attack res.	✓	✓	✓	✓	✓	✓	✓	✓	✓
DoS attack res.	✓	✓	✓	✓	✓	✓	✓	✓	✓
MiTM attack resistance	–	–	x	✓	✓	✓	✓	✓	✓
Desynchronization att. res.	–	–	✓	✓	✓	✓	✓	✓	✓
Cloning attack resistance	x	✓	x	✓	✓	✓	✓	✓	✓

Table 4 Performance comparison

	Protocol rounds	comm. Overhead (B)	Scalability	Tag's Comp. cost	Reader's comp. cost	Total Comp. cost
LH14 [45]	3	168	$O(1)$	$5T_{ecm}$	$5T_{ecm}$	$10T_{ecm}$
Z14 [49]	3	168	$O(1)$	$5T_{ecm}$	$5T_{ecm}$	$10T_{ecm}$
C14 [36]	3	160	$O(1)$	$2T_{ecm}$	$2T_{ecm}$	$4T_{ecm}$
ZQ14 [53]	3	140	$O(1)$	$2T_{ecm}$	$2T_{ecm}$	$4T_{ecm}$
BDD17 [27]	3	> 255	$O(1)$	$2T_{ecm}$	$7T_{ecm}$	$9T_{ecm}$
ID17 [28]	2	176	$O(1)$	$4T_{ecm}$	$4T_{ecm}$	$8T_{ecm}$
DB19 [56]	4	180	$O(1)$	$3T_{ecm}$	$3T_{ecm}$	$6T_{ecm}$
LZKZ18 [57]	3	> 220	$O(1)$	$4T_{ecm}$	$9T_{ecm}$	$13T_{ecm}$
Our Protocol	2	168	$O(1)$	$4T_{ecm}$	$4T_{ecm}$	$8T_{ecm}$

8.2.1 Communication Cost Comparison

Communication cost is crucial because of determining availability delays. Increase in delays usually obstacles the effective usage of the entire system. In terms of communication cost, there are two dominant factors to determine the effects, i.e. the number of protocol rounds and communication overhead. According to our analysis, only our protocol and the inspired ID17 have two rounds. DB19 scheme has four rounds and the other protocols require three rounds to provide authentication. Furthermore, the communication overhead of our protocol from reader-to-tag is 80 bytes (the public key and its signature), and 88 bytes (the public key and 3 blocks of AES encryption) transmitted from tag-to-reader. Hence, the total overhead of our protocol is 168 bytes. As seen in Table 4, ZQ14's scheme achieves the lowest communication overhead. However, they use SHA-1 algorithm for hashing the messages but SHA-1 is cryptographically insecure [83, 84]. Their communication overhead will be greater if they prefer a secure alternative hash function in their scheme. In fact, two works [13, 28] evaluate that CH14's and ZQ14's schemes have 184–186 bytes and 160–165 bytes communication overhead, respectively. Therefore, we deduce that our protocol provides the minimum communication cost considering the aforementioned factors.

Moreover, ECC point compression methods could be applied during sending public keys in the channel so that the communication efficiency might be increased in terms of communication overhead. For instance, our protocol can gain roughly 38 bytes in transmission and the communication overhead will be only 130 bytes in this case. However, this compression causes extra computations on both tag and reader sides. Note that, this point compression load might be delegated to only the reader since it has higher computational capabilities.

8.2.2 Computational Cost Comparison

In this section, we compare the computational cost of our protocol with existing ECC-based RFID authentication protocols [27, 28, 36, 45, 49, 53, 56, 57]. Table 6 summarizes the results in more detail. At first, to make an appropriate comparison, we will consider the primary operations which directly affects and determine the computation efficiency of an authentication protocol such as $T_{ecm}, T_{eca}, T_{inv}, T_{mul}, T_h$ and T_{aes}. Kobliz et al. [85] and Wu and Chen [86] analyze the time complexity of various operations in terms of T_{mul}. Also, these metrics are accepted by [13, 27]. Table 5 depicts their running time comparison of these primary operations.

We calculate the computation cost of our proposed protocol and the related works based on the above analysis in terms of T_{mul}. The tag and reader computational cost of our protocol are separately around $4T_{ecm} + 1T_{eca} + 2T_{inv} + 4T_{mul} + 2T_h + 2AES = 4817T_{mul}$, so the

Table 5 Notations and running time of primary operations in terms of T_{mul} [86]

T_{mul}	Modular multiplication in $\mathbb{F}_{2^{163}}$	1
T_{add}	Modular addition in $\mathbb{F}_{2^{163}}$	*negligible*
T_{aes}	AES-128 encryption	$\approx 0.15T_{mul}$
T_h	SHA (512-bit) hashing	$\approx 0.36T_{mul}$
T_{inv}	Modular inversion in $\mathbb{F}_{2^{163}}$	$\approx 3T_{mul}$
T_{eca}	EC point addition in $E(\mathbb{F}_{2^{163}})$	$\approx 5T_{mul}$
T_{ecm}	EC point multiplication in $E(\mathbb{F}_{2^{163}})$	$\approx 1200T_{mul}$

Table 6 Computational cost comparison

Protocols	Tag's computations	Reader's computations	Total cost
LH14 [45]	$5T_{ecm} + 3T_{eca} \backsimeq 6015T_{mul}$	$5T_{ecm} + 3T_{eca} \backsimeq 6015T_{mul}$	$\backsimeq 12030T_{mul}$
Z14 [49]	$5T_{ecm} + 3T_{eca} + 2T_{mul} \backsimeq 6017T_{mul}$	$5T_{ecm} + 3T_{eca} + 2T_{mul} \backsimeq 6017T_{mul}$	$\backsimeq 12034T_{mul}$
C14 [36]	$2T_{ecm} + 3T_{mul} + 2T_h \backsimeq 2403T_{mul}$	$2T_{ecm} + 2T_{inv} + 1T_{mul} + 2T_h \backsimeq 2408T_{mul}$	$\backsimeq 4811T_{mul}$
ZQ14 [53]	$2T_{ecm} + 1T_{eca} + 2T_h \backsimeq 2405T_{mul}$	$2T_{ecm} + 1T_{eca} + 2T_h \backsimeq 2405T_{mul}$	$\backsimeq 4810T_{mul}$
BDD17 [27]	$2T_{ecm} + 1T_{eca} + 3T_h \backsimeq 2406T_{mul}$	$7T_{ecm} + 6T_{eca} + 9T_h \backsimeq 8433T_{mul}$	$\backsimeq 10839T_{mul}$
ID17 [28]	$4T_{ecm} + 1T_{eca} + 2T_{inv} + 4T_{mul} + 2T_h + 1AES \backsimeq 4817T_{mul}$	$4T_{ecm} + 1T_{eca} + 2T_{inv} + 4T_{mul} + 2T_h + 1AES \backsimeq 4817T_{mul}$	$\backsimeq 9634T_{mul}$
DB19 [56]	$3T_{ecm} + 5T_{mul} \backsimeq 3605T_{mul}$	$3T_{ecm} + 5T_{mul} \backsimeq 3605T_{mul}$	$\backsimeq 7210T_{mul}$
LZKZ18 [57]	$4T_{ecm} + 3T_{eca} + 1T_h \backsimeq 4810T_{mul}$	$9T_{ecm} + 6T_{eca} + 1T_h \backsimeq 10819T_{mul}$	$\backsimeq 15629T_{mul}$
Our Protocol	$4T_{ecm} + 1T_{eca} + 2T_{inv} + 4T_{mul} + 2T_h + 1AES \backsimeq 4817T_{mul}$	$4T_{ecm} + 1T_{eca} + 2T_{inv} + 4T_{mul} + 2T_h + 1AES \backsimeq 4817T_{mul}$	$\backsimeq 9634T_{mul}$

total cost is roughly $8T_{ecm} + 2T_{eca} + 4T_{inv} + 8T_{mul} + 4T_h + 2AES = 9634T_{mul}$. According to Table 6, it is clearly seen that our scheme performs an acceptable computational cost. The schemes [36, 53] have better computational efficiency, however, they have serious security and privacy issues. In fact, these results show us that EC point multiplication T_{ecm} is a dominant and decisive operation to determine the computational cost of a protocol. Hence, we claim that calculating T_{ecm} is enough for evaluating the computational cost of an ECC based authentication protocol, in general. We presented this interpretation in Table 4 to intelligibly demonstrate our performance comparison.

8.2.3 Our Implementation Environment and Results

To explore the practical usage of our proposed design, we implemented our scheme in a real-world RFID system. the overwhelming majority of authors [13, 27, 36, 45, 49, 51, 53, 56], except [28], present computational cost of their protocols by referencing previous simulation results [87, 88] in their performance evaluations. Hence, a real world implementation is valuable.

We implemented our proposed scheme in ZeitControl's BasicCard environment [89]. We use a personal computer as a back-end server which has Intel Core i5 CPU processor @2.5GHz, 6GB RAM and 64-bit Windows 7 operating system to run simulations, develop codes (P-Code) and download codes to RFID tag. We can test our codes even if we do not have RFID reader and tag, by simulating the BasicCard environment in the computer. This feature is quite useful for protocol designers before testing their schemes in real-wold applications. The computer basically controls the reader and stores the system data. Also, the software of the BasicCard environment, which is free and functional, supports a higher level language such as Java or ZC-Basic (dialect of Basic). We write our own codes with ZC-Basic because using ZC-Basic is easy to program and there is a detailed library about its usage. Additionally, the heart of a BasicCard processor is its P-Code (like Java

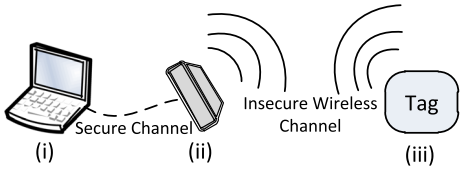

Fig. 6 Overview of our RFID system [i. Back-end server, ii. Reader (OMNIKEY 5321), iii. Tag (BasicCard ZC7.5)]

Table 7 Time-memory cost of our proposed protocol implementation in BasicCard

Code sizes (B)	Data sizes (B)	EEPROM usage (B)	RAM usage (B)	Total running time (ms)
488	3278	567	1510	442

programming language) interpreter and written codes are compiled into a machine-independent language called P-Code which is similar to machine code [89].

Moreover, we use OMNIKEY 5321 device as an RFID reader. The reader complies with ISO 15693 and ISO 14443 standards and can communicate 13.56 MHz RFID tags. We implement our proposed scheme in professional version BasicCard ZC7.5 cards supporting ISO-14443 standard as RFID tag. The tag contains $32K$ of EEPROM and $4.3K$ RAM. In the tag, there are also three processors such as CPU, RSA/ECC, and DES/AES co-processors. The overview of our RFID system is depicted in Fig. 6. Finally, our implementation considers the following parameters: a binary 160-bit elliptic curve over of the form $y^2 = x^3 + ax + b$ complying with brainpoolP160r1 standard [72].

We first simulate an RFID system and run several simulations to accelerate and mature our implementation using BasicCard development environment (v8.55). Then, we run tens of realizations and take the average time of all. In the end, we obtain the results presented in Table 7. According to the table, our proposal uses 488 bytes as code size and 3278 bytes as data size on the reader side. Besides, it has a 567 bytes EEPROM usage and 1510 bytes RAM usage on the tag side. Also, the running time of our protocol is on average 442 ms. Actually, we realize that a remarkable amount of the time is consumed for wireless channel communication.

At this point, we would like to emphasize that implementers might obtain different realization results because of some reasons: implementation platform and implementation approach (pipelining the algorithms in FPGA or using processors, etc.). For instance, the running time of ID17 scheme, in WISP platform, is roughly 12,742 ms. Its FLASH/FRAM usage is 29,450 bytes for code size and 3296 bytes for data size. The RAM usage is 1595 bytes. Thus, our implementation has better results than their WISP realization.

In our implementation, an EC point multiplication T_{ecm} takes on average 27 ms. But, this running time includes some extra operations that are used to prevent the RFID tag against side channel attacks.

On the other hand, it is obtained that T_{ecm} takes on average 1, 471 ms in the implementation of ID17 scheme [28]. The authors implement only the main components units (ECMR signature unit and ECMR recovery unit) in FPGA but they do not give any numerical results about the running time of BDD17 scheme [27]. They just present the usage hardware resources for these units such as number of flip flops, slice registers, and LUTs. Finally, the other related papers use Gódor et al.'s [87, 88] simulation results in their works. They accept that T_{ecm} takes averagely 64 ms which is slower than our result, too.

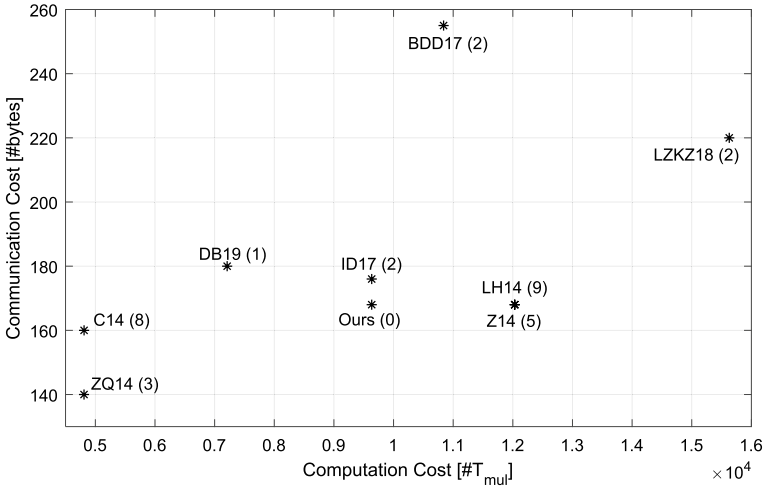

Fig. 7 Computation cost, communication cost and securiy-privacy analysis of the compared works

Figure 7 summarizes the security and performance analysis of aforementioned protocols in Tables 3, 4 and 6. It depicts the computation and communication costs in terms of number of transmitting bytes and number of the executing modular multiplication operations equivalence. It is interpreted that the performance of a scheme increases with approaching the origin point of the figure (0, 0). The more efficient protocols are located on the left-bottom of the figure. The figure also indicates the number of security and privacy vulnerabilities of a protocol within the parenthesis. Smaller value means providing more security and privacy requirements. None of the protocols, except ours, provides all the security and privacy properties mentioned in Table 3. DB19 and ours are only schemes that satisfy forward privacy. Moreover, the mean computation cost of the schemes is 9, 626 T_{mul} equivalence. ID17 and our design is very close the average. ZQ14 and C14 have higher computation efficiency with roughly 4, 800 T_{mul}. Also, the mean of byte transmission is approximately 183 bytes. Meanwhile, BDD17 and LZKZ18 are suffering from high communication cost.

9 Conclusion

In this paper, we mainly focused on both theoretical and practical aspects of ECC based RFID authentication protocols. First, we investigated vulnerabilities of the existing protocols and showed that ID17 [28], BDD17 [27], DB19 [56] and LZKZ18 [57] schemes did not provide forward and/or backward privacy. We presented our attacks against these schemes under Vaudenay's privacy model. Then, we enhanced ID17 scheme and proposed a new and practical ECC based authentication RFID protocol to efficiently satisfy all essential security and privacy properties. Thereafter, we analyzed our improved protocol in terms of security and performance perspectives. We also compared it with recent ECC-based assertive schemes and give an in-depth comparison.

Considering the practicality, we explored the realization of the existing protocols. To the best of our knowledge, the overwhelming majority of ECC based RFID protocols have not yet been implemented and tested so far in a real-word RFID system. Among the previous

protocols, the conservative approach for evaluating the performance was utilizing only previous simulation results [87, 88]. Contrary to this approach, we implemented and tested our proposed protocol in ZeitControl's BasicCard environment, and presented the implementation results. Finally, we evaluated our realization outcomes especially in terms of communication and computational cost to show the performance of our proposed scheme in practice. We demonstrated that our proposed scheme had higher performance providing all common security and privacy features including backward and forward privacy rather than the ECC based RFID authentication protocols implemented in a real-world environment. Also, we believe that this work will shed light on future designs and evaluations of ECC based RFID protocol designers.

Author Contributions Conceptualization: AA, Methodology: AA, Formal analysis and investigation: AA, Writing—original draft preparation: AA, Review and editing: SE and SAÇ, Supervision: SE, Validation: SE and SAÇ.

Funding No funds received for this research.

Availability of Data and Material Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Code Availability Not applicable.

Ethics Approval This article does not contain any studies with animals performed by any of the authors.

Consent to Participate Not applicable.

Consent for Publication Not applicable.

References

1. Bello, O., Zeadally, S., & Badra, M.. (2017). Network layer inter-operation of device-to-device communication technologies in internet of things (IoT). *Ad Hoc Networks, 57*, 52–62. Special Issue on Internet of Things and Smart Cities security, privacy and new technologies.
2. Eteng, A. A., Rahim, S. K. A., & Leow, C. Y. (2018). *RFID in the internet of things* (pp. 135–152). London: Wiley (**chapter 5**).
3. Priyanka, D. D., Jayaprabha, T., Florance, D. D., Jayanthi, A., & Ajitha, E. (2016). A survey on applications of RFID technology. *Indian Journal of Science and Technology, 9*(2), 1–5.
4. Finkenzeller, K. (2003). *RFID handbook: Fundamentals and applications in contactless smart cards and identification* (2nd ed.). New York: Wiley Publishing.
5. Zhang, D., Huang, H., & Jo, M. (2015). Future RFID technology and applications: Visions and challenges. *Telecommunication Systems, 58*(3), 193–194.
6. Kardas, S., Celik, S., Bingöl, M. A., & Levi, A. (2013). A new security and privacy framework for RFID in cloud computing. In *IEEE 5th international conference on cloud computing technology and science, CloudCom 2013, Bristol, United Kingdom, December 2–5, 2013, Volume 1* (pp. 171–176).
7. Bingöl, M. A., Birinci, F., Kardaş, S., & Kiraz, M. S. (2012). Anonymous RFID authentication for cloud services. *International Journal of Information Security Science, 1*(2), 32–42.
8. Roberti, M. (2017). When RFID becomes obsolete. *RFID Journal Blog*. Accessed on 17 March, 2018.
9. Avoine, G., Bingöl, M. A., Carpent, X., & Kardaş, S. (2013). *Deploying OSK on low-resource mobile devices* (pp. 3–18). Berlin, Heidelberg: Springer.
10. Avoine, G. (2018). RFID lounge. http://www.avoine.net/rfid/. Accessed on 26 February 2018.

11. Arslan, A., Kardaş, S., Çolak, S. A., & Ertürk, S. (2018). Are RNGs Achilles' heel of RFID security and privacy protocols? *Wireless Personal Communications, 100*(4), 1355–1375.
12. Avoine, G., Bingöl, M. A., Carpent, X., & Yalcin, S. B. O. (2013). Privacy-friendly authentication in RFID systems: On sublinear protocols based on symmetric-key cryptography. *IEEE Transactions on Mobile Computing, 12*(10), 2037–2049.
13. He, D., & Zeadally, S. (2015). An analysis of RFID authentication schemes for internet of things in healthcare environment using elliptic curve cryptography. *IEEE Internet of Things Journal, 2*(1), 72–83.
14. Ibrahim, A., & Dalkılıç, G. (2019). Review of different classes of RFID authentication protocols. *Wireless Networks, 25*(3), 961–974.
15. Avoine, G., Bingöl, M. A., Kardaş, S., Lauradoux, C., & Martin, B. (2011). A framework for analyzing rfid distance bounding protocols. *Journal of Computer Security, 19*(2), 289–317.
16. Kardaş, S., Çelik, S., Arslan, A., & Levi, A. (2013). An efficient and private RFID. In G. Avoine & O. Kara (Eds.), *Lightweight cryptography for security and privacy* (pp. 130–141). Berlin, Heidelberg: Springer.
17. Vaudenay, S. (2007). On privacy models for RFID. In Kurosawa, K. (Ed.), *Advances in cryptology ASIACRYPT 2007*, volume 4833 of *Lecture notes in computer science* (pp. 68–87). Berlin, Heidelberg: Springer.
18. Kardaş, S., Çelik, S., Bingöl, M. A., Kiraz, M. S., Demirci, H., & Levic., A. (2014). k-Strong privacy for radio frequency identification authentication protocols based on physically unclonable functions. *Wireless Communications and Mobile Computing, 15*(18), 2150–2166.
19. Avoine, G., Coisel, I., & Martin, T. (2010). Time measurement threatens privacy-friendly RFID authentication protocols. In SB Ors Yalcin (Eds.), *Workshop on RFID security—RFIDSec'10*, volume 6370 of *lecture notes in computer science* (pp. 138–157). Istanbul: Springer.
20. Hermans, J., Peeters, R., & Preneel, B. (2014). Proper RFID privacy: Model and protocols. *IEEE Transactions on Mobile Computing, 13*(12), 2888–2902.
21. Hein, D., Wolkerstorfer, J., & Felber, N. (2009). ECC is ready for RFID–A proof in silicon. In M. A. Roberto, K. Liam, & F. Sica (Eds.), *Selected areas in cryptography* (pp. 401–413). Berlin, Heidelberg: Springer.
22. Hutter, M., Feldhofer, M., & Plos, Thomas. (2010). An ECDSA processor for RFID authentication. In S. Berna & O. Yalcin (Eds.), *Radio frequency identification: Security and privacy issues* (pp. 189–202). Berlin, Heidelberg: Springer.
23. Lee, Y. K., Sakiyama, K., Batina, L., & Verbauwhede, I. (2008). Elliptic-curve-based security processor for RFID. *IEEE Transactions on Computers, 57*(11), 1514–1527.
24. Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., & Verbauwhede, I. (2007). Public-key cryptography for RFID-tags. In *International workshop on pervasive computing and communication security—PerSec 2007* (pp. 217–222). New York City: IEEE, IEEE Computer Society.
25. Bringer, J., Chabanne, H., & Icart, T. (2008). Cryptanalysis of EC-RAC, a RFID identification protocol. In Franklin, M. K, Chi, L., Hui, K., & Wong, D. S. (Eds.), *7th International conference on cryptology and network security—CANS'08, volume 5339 of lecture notes in computer science* (pp. 149–161). Hong Kong: Springer.
26. Altop, D. K., Bingöl, M. A., Levi, A., & Savaş, E. (2017). DKEM: Secure and efficient distributed key establishment protocol for wireless mesh networks. *Ad Hoc Networks, 54*(C), 53–68.
27. Benssalah, M., Djeddou, M., & Drouiche, K. (2017). A provably secure RFID authentication protocol based on elliptic curve signature with message recovery suitable for m-health environments. *Transactions on Emerging Telecommunications Technologies, 28*(11), e3166.
28. Ibrahim, A., & Dalkılıç, G. (2017). An advanced encryption standard powered mutual authentication protocol based on elliptic curve cryptography for RFID, proven on WISP. *Journal of Sensors, 2017,* 2367312.
29. A White Paper from CoreRFID. (2017). The internet of things: Practical thoughts for bussiness. http://www.corerfid.com/wp-content/uploads/2017/12/The-IoT-White-Paper.pdf. Accessed on 19 November 2018.
30. Gueulle, P. (2012). BasicCard goes contactless a discreet alternative. http://www.basiccard.com/elektor_zc75rfid.pdf. Accessed on 19 November, 2018.
31. Wolkerstorfer, J. (2005). Is elliptic-curve cryptography suitable to secure RFID tags. In *E-CRYPT workshop RFID and lightweight crypto* (pp. 78191). Graz, Austria.
32. Tuyls, P., & Batina, L. (2006). RFID-tags for anti-counterfeiting. In Pointcheval, D., (Eds.), *Topics in cryptology—CT-RSA 2006* (pp. 115–131). Berlin, Heidelberg: Springer.
33. Schnorr, C. P. (1990). Efficient identification and signatures for smart cards. In Brassard, G., (Ed.), *Advances in cryptology—CRYPTO' 89 proceedings* (pp. 239–252). New York, NY: Springer.

34. Lee, Y. K., Batina, L., & Verbauwhede, I. (2008). EC-RAC (ECDLP based randomized access control): Provably secure RFID authentication protocol. In *2008 IEEE international conference on RFID* (pp. 97–104).
35. Okamoto, T. (1993). Provably secure and practical identification schemes and corresponding signature schemes. In Brickell, E. F. (Ed.), *Advances in cryptology—CRYPTO' 92* (pp. 31–53). Berlin, Heidelberg: Springer.
36. Chou, J.-S. (2014). An efficient mutual authentication RFID scheme based on elliptic curve cryptography. *The Journal of Supercomputing, 70*(1), 75–94.
37. van Deursen, T., & Radomirović, S. (2009). Algebraic attacks on RFID protocols. In Markowitch, O., Bilas, A., Hoepman, J.-H., Mitchell, C. J., & Quisquater, J.-J. (Eds.), *Workshop on information security theory and practice—WISTP'09, volume 5746 of lecture notes in computer science* (pp. 38–51), Brussels, Belgium: Springer.
38. van Deursen, T., & Radomirović, S. (2010). EC-RAC: Enriching a capacious RFID attack collection. In Ors Yalcin, S. B. (Eds.), *Workshop on RFID security—RFIDSec'10, volume 6370 of lecture notes in computer science* (pp. 75–90). Istanbul: Springer.
39. Lee, Y. K., Batina, L., Singelee, D., Preneel, B., & Verbauwhede, I. (2010). *Anti-counterfeiting, untraceability and other security challenges for RFID Systems: Public-key-based protocols and hardware* (pp. 237–257). Berlin, Heidelberg: Springer.
40. Lv, C., Li, H., Ma, J., & Zhang, Y. (2012). Vulnerability analysis of elliptic curve cryptography-based RFID authentication protocols. *Transactions on Emerging Telecommunications Technologies, 23*(7), 618–624.
41. Lee, Y. K., Batina, L., & Verbauwhede, I. (2009). Untraceable RFID authentication protocols: Revision of EC-RAC. In *2009 IEEE international conference on RFID* (pp. 178–185).
42. Zhang, X., Li, L., Wu, Y., & Zhang, Q. (2011). An ECDLP-based randomized key RFID authentication protocol. In *2011 International conference on network computing and information security*, (Vol. 2, pp. 146–149).
43. Chien, H.-Y. (2017). Elliptic curve cryptography-based RFID authentication resisting active tracking. *Wireless Personal Communications, 94*(4), 2925–2936.
44. An, R., Feng, H., Liu, Q., & Li, L. (2017). Three elliptic curve cryptography-based RFID authentication protocols for internet of things. In L. Barolli, F. Xhafa, & K. Yim (Eds.), *Advances on broad-band wireless computing, communication and applications* (pp. 857–878). Cham: Springer.
45. 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.
46. Moosavi, S. R., Nigussie, E., Virtanen, S., & Isoaho, J. (2014). An elliptic curve-based mutual authentication scheme for RFID implant systems. *Procedia Computer Science*, 32, 198–206. The 5th international conference on ambient systems, networks and technologies (ANT-2014), the 4th international conference on sustainable energy information technology (SEIT-2014).
47. He, D., Kumar, N., Chilamkurti, N., & Lee, J.-H. (2014). Lightweight ECC based RFID authentication integrated with an ID verifier transfer protocol. *Journal of Medical Systems, 38*(10), 116.
48. Farash, 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.
49. Zhao, Z. (2014). A secure RFID authentication protocol for healthcare environments using elliptic curve cryptosystem. *Journal of Medical Systems, 38*(5), 46.
50. Peeters, R., & Hermans, J. (2013). Attack on Liao and Hsiao's secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol. Cryptology. Report 2013/399. https://eprint.iacr.org/2013/399.
51. Alexander, P., Baashirah, R., & Abuzneid, A. (2018). Comparison and feasibility of various RFID authentication methods using ECC. *Sensors, 18*(9), 2902.
52. Farash, M. S. (2014). Cryptanalysis and improvement of an efficient mutual authentication RFID scheme based on elliptic curve cryptography. *The Journal of Supercomputing, 70*(2), 987–1001.
53. 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), 47.
54. Jin, C., Chunxiang, X., Zhang, X., & Zhao, J. (2015). A secure RFID mutual authentication protocol for healthcare environments using elliptic curve cryptography. *Journal of Medical Systems, 39*(3), 24.
55. Jin, C., Chunxiang, X., 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.
56. Dinarvand, N., & Barati, H. (2019). An efficient and secure RFID authentication protocol using elliptic curve cryptography. *Wireless Networks, 25*(1), 415–428.

57. Liu, G., Zhang, H., Kong, F., & Zhang, L. (2018). A novel authentication management RFID protocol based on elliptic curve cryptography. *Wireless Personal Communications, 101*(3), 1445–1455.
58. 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.
59. Kumar, D., Grover, H. S., & Adarsh. (2019). A secure authentication protocol for wearable devices environment using ECC. *Journal of Information Security and Applications, 47*(8), 15.
60. Naeem, M., Chaudhry, S. A., Mahmood, K., Karuppiah, M., & Kumari, S. (2020). A scalable and secure RFID mutual authentication protocol using ECC for internet of things. *International Journal of Communication Systems, 33*(13), e3906.
61. Kumar, V., Ahmad, M., Mishra, D., Kumari, S., & Khan, M. K. (2020). RSEAP: RFID based secure and efficient authentication protocol for vehicular cloud computing. *Vehicular Communications, 22,* 100213.
62. Safkhani, M., Camara, C., Peris-Lopez, P., & Bagheri, N. (2021). RSEAP2: An enhanced version of RSEAP, an RFID based authentication protocol for vehicular cloud computing. *Vehicular Communications, 28,* 100311.
63. Izza, S., Benssalah, M., & Drouiche, K. (2021). An enhanced scalable and secure RFID authentication protocol for WBAN within an IoT environment. *Journal of Information Security and Applications, 58,* 102705.
64. Agrahari, A. K., & Varma, S. (2021). A provably secure RFID authentication protocol based on ECQV for the medical internet of things. *Peer-to-Peer Networking and Applications*, *14*, 1277–1289. https://doi.org/10.1007/s12083-020-01069-z.
65. Kumari, A., Jangirala, S., Abbasi, M. Y., Kumar, V., & Alam, M. (2020). ESEAP: ECC based secure and efficient mutual authentication protocol using smart card. *Journal of Information Security and Applications, 51,* 102443.
66. Kamil, I. A., & Ogundoyin, S. O. (2021). A lightweight mutual authentication and key agreement protocol for remote surgery application in tactile internet environment. *Computer Communications, 170,* 1–18.
67. Braeken, A. (2021). Public key versus ssymmetric key cryptography in client—Server authentication protocols. *International Journal of Information Security*. https://doi.org/10.1007/s10207-021-00543-w.
68. Vijayakumar, P., Obaidat, M. S., Azees, M., Islam, S. H., & Kumar, N. (2020). Efficient and secure anonymous authentication with location privacy for IoT-based WBANs. *IEEE Transactions on Industrial Informatics, 16*(4), 2603–2611.
69. Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation, 48*(177), 203–209.
70. Miller, V. S. (1986). Use of elliptic curves in cryptography. In Williams, H. C. (Eds.), *Advances in cryptology—CRYPTO '85 proceedings* (pp. 417–426). Berlin, Heidelberg: Springer.
71. Lauter, K. (2004). The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless Communications, 11,* 62–67.
72. Merkle, J., & Lochter, M. (2010). Elliptic curve cryptography (ECC) brainpool standard curves and curve generation. RFC 5639. https://rfc-editor.org/rfc/rfc5639.txt.
73. Recommendation for Key Management, Special Publication 800-57 Part 1 Rev. 4. NIST, 01/2016.
74. Harkanson, R., & Kim, Y. (2017). Applications of elliptic curve cryptography: A light introduction to elliptic curves and a survey of their applications. In *Proceedings of the 12th annual conference on cyber and information security research*, CISRC '17 (pp. 6:1–6:7). New York: ACM.
75. Ravikumar, K., & Udhayakumar, A. (2014). Secure multiparty electronic payments using ECC algorithm: A comparative study. In *2014 World congress on computing and communication technologies* (pp. 132–136).
76. Bingöl, M. A., Biçer, O., Kiraz, M. S., & Levi, A. (2018). An efficient 2-party private function evaluation protocol based on half gates. *The Computer Journal* (bxy136). https://doi.org/10.1093/comjnl/bxy136.
77. Bicer, O., Bingöl, M. A., Kiraz, M. S., & Levi, A. (2020). Highly efficient and re-executable private function evaluation with linear complexity. *IEEE Transactions on Dependable and Secure Computing*. https://doi.org/10.1109/TDSC.2020.3009496.
78. Bingöl, M. Ali. (2019). Efficient and secure schemes for private function evaluation. Ph.d thesis, Sabanci University, Istanbul. http://research.sabanciuniv.edu/36861/.
79. Schoenmakers, B. (2018). Lecture notes cryptographic protocols version 1.32. http://www.win.tue.nl/~berry/2DMI00/LectureNotes.pdf. Accessed on 14 November, 2018.
80. Song, B., & Mitchell, C. J. (2008). RFID authentication protocol for low-cost tags. In *Proceedings of the first ACM conference on wireless network security*, WiSec '08 (pp. 140–147). New York, NY: ACM.

81. Lim, C. H., & Kwon, T.. (2006). Strong and robust RFID authentication enabling perfect ownership transfer. In Ning, P., Qing, S., Li, N., (Eds.), *Information and communications security* (pp. 1–20). Berlin, Heidelberg: Springer.
82. Phan, R.C.-W., Wu, J., Ouafi, K., & Stinson, D. R. (2011). Privacy analysis of forward and backward untraceable RFID authentication schemes. *Wireless Personal Communications, 61*(1), 69–81.
83. Stevens, M., Bursztein, E., Karpman, P., Albertini, A., & Markov, Y. (2017). The first collision for full SHA-1. In Katz, J., Shacham, H., (Eds.), *Advances in cryptology—CRYPTO 2017* (pp. 570–596). Cham: Springer.
84. Wang, X., Yin, Y. L., & Yu, H. (2005). Finding collisions in the full SHA-1. In Victor S, (Eds.), *Advances in cryptology—CRYPTO 2005* (pp. 17–36). Berlin, Heidelberg: Springer.
85. Koblitz, N., Menezes, A., & Vanstone, S. (2000). The state of elliptic curve cryptography. *Designs Codes and Cryptography, 19*(2–3), 173–193.
86. Shuhua, W., & Chen, K. (2012). An efficient key-management scheme for hierarchical access control in E-medicine system. *Journal of Medical Systems, 36*(4), 2325–2337.
87. Gódor, G., Giczi, N., & Imre, S. (2010). Elliptic curve cryptography based mutual authentication protocol for low computational capacity RFID systems—Performance analysis by simulations. In *2010 IEEE international conference on wireless communications, networking and information security* (pp. 650–657).
88. Gódor, G., & Imre, G. (2011). Elliptic curve cryptography based authentication protocol for low-cost RFID tags. In *2011 IEEE international conference on RFID-technologies and applications* (pp. 386–393).
89. ZeitControl cardsystems GmbH. (2018). BasicCard Developer Manual V8.15. http://www.basiccard.com/index.html. Accessed on 15 November.

Atakan Arslan received the B.S degree in Control Engineering and the M.S. degree in Telecommunication Engineering from Istanbul Technical University, Istanbul, Turkey in 2008 and 2012. He received the Ph.D. degree in Electronics and Telecommunication Engineering from Kocaeli University, Kocaeli, Turkey, in 2019. His primary research interests include information security, privacy, RFID systems and cryptographic protocols.

Sultan Aldırmaz Çolak received the B.S degree in Electronics and Communications Engineering from Kocaeli University, Kocaeli 2004 and the M.S. and PhD degrees in Yildiz Technical University (YTU), Istanbul in 2006 and 2012, respectively. She was a visiting research scholar in the Department of Electrical and Computer Engineering of University of South Florida for the spring and summer of 2009. She is currently an Assoc. Professor in the Electronics and Communications Engineering Department of Kocaeli University, Kocaeli, Turkey. Her primary research interests include 5G systems, heterogeneous networks, MIMO systems, index modulation, and visible light communications.

Sarp Ertürk (M'99) received the B.Sc. degree in electrical and electronics engineering from Middle East Technical University, Ankara, Turkey, in 1995 and the M.Sc. degree in telecommunication and information systems and the Ph.D. degree in electronic systems engineering from the University of Essex, Colchester, U.K., in 1996 and 1999, respectively. From 1999 to 2001, he carried out his compulsory service at the Army Academy, Ankara. He is currently a Full Professor with the Department of Electronics and Telecommunication Engineering, Kocaeli University, where he was an Assistant Professor between 2001 and 2002 and an Associate Professor between 2002 and 2007. His research interests include digital signal and image processing, video coding, remote sensing, and digital communications.