Wireless Personal Communications (2021) 117:2513–2539
https://doi.org/10.1007/s11277-020-07992-x

An Efficient RFID Authentication Scheme Based on Elliptic Curve Cryptography for Internet of Things

Mustapha Benssalah[1] · **Izza Sarah**[1] · **Karim Drouiche**[2]

Accepted: 11 November 2020 / Published online: 20 November 2020

Abstract
With the rapid development of microelectronics devices and the progress in communication and information technologies, many services and technologies are increasingly involved into our daily life. In fact, as the used systems are progressively interconnected and open, this introduce new threats such as more and more hacking, fraud and many other kinds of misuses. Consequently, the security and privacy of the exchanged data information tampering must be addressed most seriously. In this context, recently Elliptic Curve Cryptography (ECC) is widely used in many cryptosystems nowadays especially for those presenting challenging constraints in terms of power consumption, memory, computational cost, etc. It is well-known that the ECC provides high security level with much smaller key sizes. In this paper, we show that an inappropriate use of ECC cryptographic primitives, the lack of experience in designing secure protocols and the unsuitable choice of security verification tools can destroy the whole security of a given ECC-based scheme. Therefore, first we wreck efficient attacks on three most recent proposed ECC-based protocols published in three of well-known scientific journals. Then, an improved protocol that inherits the strengths of Dinarvand and Barati's protocol and takes into account the discovered flaws is proposed. Via formal and informal security models, we assess that the improved protocol could deliver all the virtues of Dinarvand and Barati's protocol and resists all known attacks.

Keywords ECC · Communication · Wireless networking · RFID · Security analysis

1 Introduction

With the fast development of wireless communication technologies and the Internet of Things (IoT) networks, many network services and wireless devices have been generated and introduced in favor of benefits of human well-being. In fact, we assist to a growth in connectivity and data traffic convoyed by innumerable information and communications

✉ Mustapha Benssalah
bensmusta@gmail.com

[1] Laboratoire Traitement du Signal, 'Ecole Militaire Polytechnique, BP 17 Bordj El Bahri, 16111 Algiers, Algeria

[2] LIK Neuville Sur Oise, Cergy Pontoise University, Cergy-Pontoise CEDEX, France

technologies (ICT), such as WiFi, sensors, Bluetooth, advanced mobile communications (3G/4G), Radio Frequency Identification (RFID), etc [1, 2]. We undoubtedly that one of the main issues that the scientific community may face will be the security and privacy of the exchanged personal and secret data. In other words, the security and privacy will be one of keys distinctive indicators that other relevant performance such as data rate, range, latency, etc. Consequently, the security and the privacy aspects must be guaranteed urgently.

In the recent years, numerous cryptographic solutions have been introduced in the literature to keep data safe over insecure public channels in ICT [3–6]. In fact, a variety of crypto-algorithms classes exist, including symmetric and asymmetric cryptosystems, hashing algorithms, etc. The Elliptic Curve Cryptography (ECC) cryptosystems similar to Rivest, Shamir and Adleman (RSA) belong to the asymmetric class that allows solving numerous problems such as the key management problem and the authentication issue for small devices with limited resources. Moreover, the ECC becomes as a crucial security mechanism for several common standards, services and authentication protocols such as Internet Key Exchange (IKE), Secure Internet Live Conferencing (SILC), Secure Multipurpose Internet Mail Extensions (SMIME), etc. [2, 7]. The ECC is widely implemented in many devices such as mobile phones, smart cards, biometric passports and some other important businesses [4, 5]. The asymmetric encryption approach is shown its imperative features comparing to its biggest competitors RSA by offering significantly lower computational workload, lower processing unit consumption, lower memory usage and tiny key sizes. In this context, for a comparable symmetric key length of 80 bits, the ECC requires only 160 bits for the same security level, which make it computationally lighter for longer keys. In addition, it is shown that the required processing time to encrypt/decrypt data using ECC is 400 times less than the needed time for an equivalent RSA key size [2, 8]. These overlap exactly with IoT and RFID devices limitations and challenges that make classical cryptography complicated to implement. The constraints for these tiny devices include computational workload, power consumption, memory and processor speed. In addition, the challenges include the identity management, devices and users registrations and the suitable use for IoT.

Nevertheless, a significant number of potential vulnerabilities on ECC can be operated in case judicious engineering practices and sanity recommendations to carefully follow are not cautiously performed. These attacks could include the twist-security, side channel attacks and so on [2, 9, 10]. In fact, these attacks threaten to reduce the provided high security level of ECC to secret keys. Side channel attacks usually based on information leaked from the physical implementation of the cryptosystem rather than mathematical flaws of the algorithm. This kind of attacks includes, simple power attacks, differential power analysis, simple timing attacks, electromagnetic attacks, fault analysis attacks, etc.

Recently, it has been shown that ECCs are now possible for securing RFID chips which is considered as one of the leading technologies alongside IoT [11, 12]. This suitability was considered as an important and open research issue in these past years due to the challenging constraints in terms of area, computational cost and power consumption. In this context, numerous RFID authentication protocols have been suggested in the literature to address the security and privacy problems in this technology.

In 2006, Tuyls and Batina [13] proposed an RFID anti-counterfeiting authentication protocol using ECC. In 2007, Batina et al. [2] suggested a similar authentication scheme for RFID using the public-key ECC based. However, in 2008, Lee et al. [14] showed that Tuyls and Batina's protocol [13] and Batina et al.'s [2] present privacy flaws. Then, Lee et al. suggested an improved version using ECC. In 2013, Liao and Hsiao [15] designed a secure RFID authentication scheme ECC-based combined with ID-verifier transfer

protocol. The authors of [15] claimed that their scheme could resist to various attacks. However, in 2014, Zhao [16] demonstrated that the Liao and Hsiao protocol [15] presents the key comprise problem where an attacker can reveal the tag's private key. Then, Zhao presented an enhanced version. In the same year, Chou et al. [17] designed a new authentication protocol ECC-based to improve the patient medication safety. The authors of [17] showed that their protocol can resist to the well-known attacks in healthcare environment. Unfortunately, Zhang and Qi [18] confirmed that Chou et al.'s protocol presents the tag's privacy information leakage and the forward and backward traceability problems. Then, the authors of [18] proposed an improved authentication protocol version ECC-based. In the same year, He et al. [19] designed a lightweight RFID authentication ECC-based integrated with an ID verifier transfer scheme and they showed that their protocol could overcome the flaws of the existing protocols. Elsewhere, Qu and Tan [20] presented a two-factor remote authentication and key agreement scheme where they pointed out that this scheme could resist to various attacks such as impersonation attack, off-line password guessing attack and smart card loss attack, etc. Unfortunately, Huang et al. [21] proved that Qu and Tan [20] scheme is vulnerable to the impersonation and off-line password guessing attacks. To address these flaws, Huang et al. proposed an improved scheme to simplify user authenticity, where they showed that this protocol is secure and practical as the secure universal access control mechanism. Nevertheless, Chaudhry et al. [22] showed that Huang et al. presents correctness problems and is vulnerable to impersonation and forgery attacks. To address these issues, an improved lightweight secure version is proposed. In 2015, Chen and Chou put forward an untraceable authentication scheme for large-scale active RFID tags ECC-based [23]. The authors of [23] claimed that their scheme had high performance and could resist to various attack. Unfortunately, Shen et al. [24] proved that Chen and Chou's scheme is vulnerable to replay attack and to server impersonation attack. Somewhere else, Jin et al. [25] proposed a secure RFID authentication protocol using ECC suitable for healthcare environments. Jin et al. used pre-computing method within tag's communication to get more efficiency. In 2017, Luo et al. [26] demonstrated that the dynamic ID-based remote user authentication ECC-based presented by Islam et al.'s [27] is prone to insider attack and off-line password guessing attack. Then, to overcome these imperfections, Luo et al. suggested an improved scheme that could defend various attacks in e-commerce services with mobile devices. In 2018, Madhusudhan et al. [28] observed that Troung et al. protocol [29] that was proposed earlier in 2014 does not provide perfect forward secrecy, replay attack, user anonymity and server's secret key security. Then, to fix these vulnerabilities, they put forward a new authentication scheme. Liu et al. [30] proposed first, a key negotiation mechanism followed by an authentication protocol ECC-based in mobile RFID system where, they showed that their scheme presents more efficient performance and its capacity to resist various attacks. Elsewhere, Adhikar et al. [31] suggested ECC-based secure efficient communication protocol for flexible content centric network (CCN) to protect the existing business policies. Later, in 2018, Naresh et al. [32] proposed a lightweight secure communication system using hyper elliptic curve (HEC) where they showed the possibility of implementing the HEC for wireless sensor network. Qi et al. [33] put forward also a new robust biometrics-based authentication scheme with key agreement phase using ECC. Unfortunately, Sahoo et al. [34] demonstrated that this scheme cannot resist to the off-line password guessing attack, the key compromise impersonation attack and to the known session-specific temporary information attack. To fix all these deficiencies, Sahoo et al. suggested an improved biometric based authentication scheme using ECC with more security features. Alamr et al. [5] put forward an RFID EC-Diffie-Hellman based key exchange scheme for IoT, where they claimed that their scheme has the ability to

defend against various security attacks. However, most recently Naeem et al. [35] showed that the scheme of Alamr et al. is not scalable and can satisfy only one tag. Then, they introduced an improved scalable scheme suitable for IoT environment.

Despite of the excellent performance of the ECC in terms of security properties and computation cost, we find that many ECC-based protocols have critical weaknesses caused by several factors such as design immaturity of some authors, non-rigorous security verification using appropriate security tools, a little efforts in security verification process, etc. In this paper, we pay attention on three recently published protocols in well-known journals by Liu et al. [30], Naeem et al. [35] and Dinarvand and Barati [3]. First, we show an efficient impersonation attack on Liu et al. [30] authentication protocol that exploits design typos in tag's response messages caused basically on little efforts in security verification process. Then, through efficient secret identifier disclosure attack and impersonation attack, we demonstrate that Naeem et al. protocol [35] has a serious security issues that are related to lack of rigorous design verification process. Moreover, we prove via an efficient twist attack that inappropriate use of cryptographic primitives ECC-based and a non-meticulous validation of the EC domain parameters at each step of the protocol execution can destroy the security of a given scheme. Consequently, we present an efficient invalid curve attack on a most recently proposed RFID authentication protocol using ECC proposed by Dinarvand and Barati [3]. Through simulation analysis, we will show how to extract the tag's identifier and then impersonate the legitimate reader to any communication partner. As a remedy, we give solutions for each discovered flaw for Liu et al. and Naeem et al. protocols' and a complete improved version for Dinarvand and Barati [3] protocol.

The organization of the rest of paper is as follows; the next section summarizes the ECC background. The security analysis of Liu et al., Naeem et al., and Dinarvand and Barati protocols are given in Sects. 3, 4 and 5, respectively. Section 6 is devoted to the improved protocol. Section 7 gives the security analysis of the improved protocol. In Sect. 8, the performance of the improved protocol is evaluated with comparison to some related works. Finally, we conclude this paper in Sect. 9.

2 Background

ECC schemes are public-key mechanisms proposed independently by Koblitz and Miller [8]. The ECC are built on the elliptic curves algebraic construction over finite fields. The elliptic curve cryptography provides the same functionality as the conventional asymmetric cryptography such as RSA schemes [8]. Let q be a large prime number. An elliptic curve (E) over a prime finite field $GF(q)$ is the set of solutions of the plane curve given by [8]:

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

where a and b in $GF(q)$ satisfying $4a^3 + 27b^2 \neq 0 \pmod{q}$.

The set of points (x, y), where $x, y \in GF(q)$ that satisfies the Eq. (1) form the Abelian group G with an additional point at infinity denoted by (∞), i.e. $G = \{(x, y) : x, y \in GF(q); (x, y) \in E\} \cup \{\infty\}$b [7, 8].

In the following, we give some group law for the curve E:

– Identity element, the point (∞) works as the identity element of G: $P + \infty = \infty + P = P$ for all points $P \in G$.

- If $P = (x, y) \in G$, then $(x, y) + (x, -y) = (\infty)$. The negative point of P is $(x, -y)$ denoted by $-P$.
- Point addition and doubling: let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2) \in G$, where $P_1 \neq \pm P_2$. Then $P_1 + P_2 = R = (x_3, y_3)$, where $x_3 = \lambda^2 - x_1 - x_2$ and $y_3 = \lambda(x_1 - x_2) - y_1$, where $\lambda = \frac{(y_2 - y_1)}{(x_2 - x_1)}$ if $P_1 \neq P_2$ and $\lambda = \frac{3x_1^2 + a}{2y_1}$ if $P_1 = P_2$.
- Let $P(x, y)$ a point $\in G$ and k is an integer, then the scalar point multiplication operation is defined as follows: $kp = \underbrace{P + P + \ldots + P}_{(k \text{ times})}$. Thus, the problem which consists to extract the integer k given the product kP and the point P is known as the ECDLP (elliptic curve discrete logarithm problem) [8]. The assumed hardness of numerous problems related to ECDLP in a subgroup of G allows the cryptographic use of ECC.

On the other hand, in a given ECC cryptosystem, all contributing entities have to share a set of elements known as the elliptic curve domain parameters given by (a, b, q, P, n, h), where P is the base point of the cyclic subgroup, n is the order of the point P ($nP = \infty$) and h is the co-factor [8].

Nowadays, the ECC is used in numerous applications and standards thanks to its benefits such as smaller parameters with higher security level compared with conventional public key crypto-systems. Smaller key sizes allow faster computations and smaller certificates and less complexity of the cryptosystem.

Definition 1 *ECDLP*: The ECDLP is defined as follows: Given points P and $Q \in G$, find the scalar l such that $Q = lp$, which is computationally difficult. If we denote $Adv_{\hat{A}}^{ECDLP}(t)$ as the advantage of the adversary $\hat{A}$ to find $l \in [1, n-1]$, given Q and P such that $Q = lP$ for a specified time interval t. So, the ECDLP is an intractable problem, if the $Adv_{\hat{A}}^{ECDLP}(t) = Pr[l \in [1, n-1] | Q = lP] < \epsilon$, for any sufficiently small $\epsilon > 0$ [10].

Definition 2 *Hash function collision resistance property*: A one way collision resistant hash function is mathematical function that maps a string of any length to another of fixed length, denoted the hashed value [36]. $h : X \rightarrow Y$, where $X = \{0, 1\}^*$ and $Y = \{0, 1\}^n$. This property is expressed as follows:

$Adv_{\hat{A}}^{Hash}(t) = Pr[(x, x') \Leftarrow_R \hat{A} : x \neq x' \text{ and } h(x) = h(x')]$, where $Pr[e]$ is the random event e probability, $(x, x') \Leftarrow_R \hat{A}$ is the pair message (x, x') arbitrarily chosen by adversary $\hat{A}$ and $Adv_{\hat{A}}^{Hash}(t)$ signifies the probability advantage, over a random choices, made by the $\hat{A}$ for a time t. Therefore, this function is collision-resistant, if $Adv_{\hat{A}}^{Hash}(t) < \epsilon$ for a very small values of $\epsilon > 0$ [37].

3 Security Analysis of Liu et al. Protocol

Recently, Liu et al. [30] proposed a new key negotiation and authentication RFID protocol based on ECC divided in two separate schemes; one for the key establishment mechanism and the second for the authentication protocol. In this section, we focus only on authentication protocol and interested reader can refer to original paper [30] for more detail. The main steps of this protocol are depicted in Table 1 and summarized in the following two phases:

Table 1 Liu et al. [30] authentication protocol

Server $\{T_D, R_D, k_{AB}\}$ $\{k_{AC}, a, P_s = aP\}$	Insecure channel	Reader $\{R_D, k_{AB}\}$ $\{b, P_R = bP\}$	Insecure channel	Tag $\{T_D, k_{AC}\}$ $\{c, P_T = cP\}$
$x_S \in_R Z_q, S_1 = x_S P$		$x_R \in_R Z_q$	$\xrightarrow{Query, R_1}$	$x_T \in_R Z_q$
$S_2 = H(R_1 \parallel k_{AB} \parallel t_R)$		$R_1 = x_R P$		$T_1 = x_T P$
Judge:$S_2 \stackrel{?}{=} R_3$		$R_2 = H(x_R T_1)$		$T_2 = H(x_T R_1)$
	$\xleftarrow{T_1, T_3, R_1}$	Judge:$R_2 \stackrel{?}{=} T_2$	$\xleftarrow{T_1, T_2, T_3}$	$T_3 = T_D + (x_T + c)P_s$
$S_3 = R_4 - aR_1 - k_{AB}$	R_3, R_4, t_R	$R_3 = H(R_1 \parallel k_{AB} \parallel t_R)$		
Judge:$S_3 \stackrel{?}{=} R_D$		$R_4 = R_D + (x_R + b)P_s$		
$S_4 = T_3 - aT_1 - k_{AC}$				
Judge:$S_4 \stackrel{?}{=} T_D$	$\xrightarrow{S_1, S_5, S_6}$	$R_5 = x_R S_1 + k_{AB}$	$\xrightarrow{S_1, S_6}$	$T_4 = x_T S_1 + k_{AC}$
$R_5 = x_S R_1 + k_{AB}$		Judge:$R_5 \stackrel{?}{=} S_5$		Judge:$T_4 \stackrel{?}{=} S_6$
$R_6 = x_S T_1 + k_{AC}$				

Initialization phase

1. The mobile reader, the background database and the electronic tags share a set of system domain parameters (q, a, b, P, n, h).
2. Through the key negotiation mechanism, the reader and the database share the secret key k_{AC} and in addition, the reader share k_{AB} with the tag.
3. A random point (x_1, y_1) is selected from the elliptic curve E as the identity identifier of the ith tag T_D and another point (x_2, y_2) is selected as the identity identifier of the jth reader R_D.
4. The database stores the tag and reader identifiers T_D and R_D and its own public/private keys $\langle P_S = aP, a \rangle$.
5. The tag stores its own identifier T_D and its public/private keys $\langle P_T = cP, c \rangle$.
6. The reader stores its own identifier R_D and its public/private keys $\langle P_R = bP, b \rangle$.

Authentication phase

- First, the reader generates a random number $x_R \in_R Z_q$ and calculates $R_1 = x_R P$ and sends it to the tag.
- Upon receiving the request, the tag generates a random number $x_T \in_R Z_q$ and computes $T_1 = x_T P$, $T_2 = H(x_T R_1)$ and $T_3 = T_D + (x_T + c)P_s$ and sends T_1, T_2, T_3 to the reader.
- After the reception of T_1, T_2, T_3, the reader computes $R_2 = H(x_R T_1)$ and verifies if $R_2 \stackrel{?}{=} T_2$. If not, it rejects the request. If the two are equal, the reader authenticates the tag and it continues to calculate $R_3 = H(R_1 \parallel k_{AB} \parallel t_R)$, $R_4 = R_D + (x_R + b)P_s$ and sends $T_1, T_3, R_1, R_3, R_4, t_R$ to the server.
- After receiving $T_1, T_3, R_1, R_3, R_4, t_R$, first the server verifies the validity of the timestamp. If t_R is valid, the server continues the authentication by generating a random number $x_S \in_R Z_q$ and computes $S_1 = x_S P$, $S_2 = H(R_1 \parallel k_{AB} \parallel t_R)$ and verifies if $S_2 \stackrel{?}{=} R_3$. If not, the authentication fails else; it authenticates the reader (as a legal reader). The server continues the authentication process by calculating $S_3 = R_4 - aR_1 - k_{AB}$ and verifying if $S_3 \stackrel{?}{=} R_D$. If the equality does not hold, the authentication fails; else the reader's R_D is the authorization identifier. Then, the server continues the process by computing

$S_4 = T_3 - aT_1 - k_{AC}$ and checking if $S_4 \stackrel{?}{=} T_D$. If it does not hold, the authentication is not valid; else the tag's T_D is the authorization identifier. Finally, the server calculates $R_5 = x_S R_1 + k_{AB}$, $R_6 = x_S T_1 + k_{AC}$ and sends them to the reader.
- Upon receiving the message, the reader computes $R_5 = x_R S_1 + k_{AB}$ and checks whether $R_5 \stackrel{?}{=} S_5$. If it does not hold, the authentication fails; else it authenticates the server. Then, the reader sends S_1 and S_6 to the tag.
- Upon the reception of S_1 and S_6, the tag calculates $T_4 = x_T S_1 + k_{AC}$ and checks whether $T_4 \stackrel{?}{=} S_6$. If not, the authentication fails; else the server and the reader are valid.

3.1 Tag Impersonation Attack

In this section, we show that Liu et al. [30] has critical weakness. Then, the proposed attack is in light of a flaw of the protocol related to tag's response which is not carefully scrutinized. Therefore, we illustrate how an attacker could exploit this kind of vulnerability to generate a fake tag's response that could pass the reader authentication process. The tag impersonation attack is given as follows:

1. In the absence of the legitimate reader, the attacker interrogates the tag by sending the request message $\texttt{Query}, R_1 = P$.
2. The tag proceeds as follows: it generates $x'_T \in_R Z_q$ and calculates $T'_1 = x_{T'} P, T'_2 = H(x'_T P)$ and $T'_3 = T_D + (x'_T + c)P_s$ and returns T'_1, T'_2 and T'_3 to the attacker.
3. Upon receiving the tag response, the attacker saves: $T_3 = T_D + (x'_T + c)P_s$ and $T_1 = P$.
4. Now, when a legitimate reader initiates a new session by sending a message query $R_1 = x_R P$ to the tag, the attacker intercepts it and responds by putting: $T_1 = P$, $T_3 = T_D + (x'_T + c)P_s$ and calculates $T_2 = H(R_1) = H(x_R P)$.
5. Upon the reception of this fake tag response from the attacker, the reader computes $R_2 = H(x_R T_1) = H(x_R P)$ and verifies if $R_2 \stackrel{?}{=} T_2$. In this case, we have the equality and then the reader authenticates the attacker as the legitimate tag and continues the protocol steps.

This attack could be avoided whether well-known principals for designing secure cryptographic schemes would have been seriously valued and followed. In addition, we found that there is a lack of security design maturity in this field for the authors which require a lot of experience. The problem in this protocol is that the tag response $\{T_1, T_2 \texttt{ and } T_3\}$ did not incorporate something related to the message T_1 in the hash function which is used here to guarantee the integrity. To fix this pitfall, we suggest to change the tag response as follows: $T_1 = x_T P, T_2 = H(x_T R_1 \parallel T_1)$ and $T_3 = T_D + (x_T + c)P_s$.

4 Security Weaknesses of Naeem et al. Protocol

Most recently in 2019, Naeem et al. [35] suggested an enhanced RFID authentication protocol for Internet of things environment claiming that it provides a high security level and low computation and communication costs. This authentication scheme is

subdivided into two phases: initialization and authentication phases as given as follows and summarized in Table 2:

4.1 Initialization Phase

1. The server produces a set of system parameters and it chooses the different tags identities X_{T_i}.
2. The server chooses P_{r_R} as a random number that represents the reader's secret key and calculates the public key $P_{u_R} = P_{r_R}P$. It stores $\{P_{r_R}, P_{u_R}\}$ in the reader memory.
3. The reader operates on the database in which the server stores the tags secret identities.
4. The server inserts each reader's public key and tag's identity in the corresponding tag's memory.

4.2 Authentication Phase

- First, the reader generates a random number r_1 and calculates $R_1 = r_1 P$ and sends it to the tag.
- Upon receiving the message, it generates a random number t_1 and calculates $T_1 = t_1 P$, $C_1 = t_1 R_1$ and $C_2 = X_{T_i} + h(T_1, R_1, C_1)$. Then, it sends back C_1 and C_2 to the reader.
- After the reception of C_1 and C_2, the reader computes $T_1 = C_1 r_1^{-1}$ and $X_{T_i} = C_2 - h(T_1, R_1, C_1)$ and compares it with X_{T_i} in its database. If X_{T_i} is not found then, the reader ignores the request else; the reader authenticates the tag and calculates $C_3 = P_{r_R} T_1$ and $C_4 = h(C_3, T_1, R_1, C_1)$. Then, it sends C_4 to the tag.
- Upon receiving C_4, the tag computes $Y = P_{u_R} t_1$ and authenticates the reader only if $C_4 \stackrel{?}{=} h(Y, T_1, R_1, C_1)$.
- Finally, the tag calculates the shared session key $TK_{ag} = X_{T_i} t_1 R_1$ and in the other side, the reader computes the same session key $RK_{ag} = X_{T_i} r_1 T_1$.

Table 2 Naeem et al. [35] authentication protocol

Tag $\{P_{u_R}, X_{T_i}, n, P\}$	Insecure channel	Reader $\{P_{r_R}, P_{u_R}, X_{T_i}, n, P\}$
		Generates r_1
Generates t_1	$\xleftarrow{R_1}$	Computes $R_1 = r_1.P$
$T_1 = t_1.P, C_1 = t_1.R_1$		
$C_2 = X_{T_i} + h(T_1, R_1, C_1)$	$\xrightarrow{C_1, C_2}$	
		$T_1 = C_1.r_1^{-1}$
		$X_{T_i} = C_2 - h(T_1, R_1, C_1)$
$Y = t_1.P_{u_R}$		Checks with X_{T_i} in database
Authenticates the reader if	$\xleftarrow{C_4}$	$C_3 = P_{r_R}.T_1$
$C_4 \stackrel{?}{=} h(Y, T_1, R_1, C_1)$		$C_4 = h(C_3, T_1, R_1, C_1)$
$TK_{ag} = X_{T_i}.t_1.R_1$		$RK_{ag} = X_{T_i}.r_1.T_1$

4.3 Secret Identifier Disclosure Attack

In this subsection, we show that Naeem et al. [35] has a serious security issues that are related to lack of rigorous design verification process. We found out this protocol is vulnerable to secret identifier disclosure attack and tag impersonation attack. In fact, the tag's identity is assumed to be a shared secret parameter between the reader and the tag only, because any reveal of this parameter will allow to the adversary to track, to localize and even to impersonate the reader. The disclosure of this secret identifier is given as follows:

1. In the absence of the legitimate reader, the attacker interrogates the tag (pretending to be the legitimate reader) by putting the random number $r_1 = 1$, calculating and sending $R_1 = r_1 P = P$.
2. Upon receiving R_1, the tag generates a random number t_1 and calculates $T_1 = t_1 P$, $C_1 = t_1 R_1 = t_1 P = T_1, C_2 = X_{T_i} + h(C_1, P, C_1)$, and it sends C_1 and C_2 to the attacker.
3. The attacker uses the tag response to calculate $X_{T_i} = C_2 - h(C_1, P, T_1) = C_2 - h(C_1, P, C_1)$ and to disclose the secret tag identity X_{T_i}. Consequently, Naeem et al. protocol is vulnerable to the secret tag identity disclosure attack.

4.4 Tag Impersonation Attack

Now, once the attacker has the tag secret identifier X_{T_i}, it can impersonate the reader (the user) as follows:

1. When the legitimate reader initiates a new session by sending a message query $R_1^{'} = r_1^{'} P$ to the tag, the attacker intercepts this message.
2. Then, the attacker generates a random number $t_1^{'}$ and computes: $T_1^{'} = t_1^{'} P$, $C_1^{'} = t_1^{'} R_1^{'}$, $C_2^{'} = X_{T_i} + h(T_1^{'}, R_1^{'}, C_1^{'})$, then it sends $C_1^{'}$ and $C_2^{'}$ to the reader.
3. Upon the reception of $C_1^{'}$ and $C_2^{'}$, the reader computes $T_1^{'}$ and extracts X_{T_i}, then it compares this latter with the tag's secret identity stored in its database. Then, the reader authenticates the attacker believing that it is the legitimate tag. Consequently, Naeem et al. protocol is vulnerable to the tag impersonation attack.

These attacks are in light of a flaw related to the tag response message which is not carefully scrutinized $\langle T_1, C_1, C_2 \rangle$. The problem is that we can evaluate the hashing function of the message C_2 which is used to mask the tag identity X_{T_i}. In other words, we can easily deduce the only unknown message T_1 for a specific request ($R_1 = P$), i.e. $T_1 = C_1 = t_1.P = t_1.R_1$ for $R_1 = P$. Finally, this flaw can be fixed by redesigning the tag response using the public key of the reader P_{u_R} as summarized in Table 3. In this case, it is difficult to an attacker to construct a valid tag response using only the public exchanged messages without knowing the reader secret key P_{r_R} which is linked to the public key P_{u_R} by the ECDLP problem.

5 Security Analysis of Dinarvand and Barati Protocol

Most recently in 2019, Dinarvand and Barati [3] suggested a new RFID authentication protocol based on ECC to overcome flaws of the existing authentication schemes published earlier. The authors of [3] showed that their protocol presents distinguished security

Table 3 The improved tag response to fix the discovered flaw

Tag $\{P_{u_R}, X_{T_i}, n, P\}$	Insecure channel	Reader $\{P_{r_R}, P_{u_R}, X_{T_i}, n, P\}$
Generates t_1	$\xleftarrow{R_1}$	Generates r_1
$T_1 = t_1.P_{u_R}, C_1 = t_1.R_1$		Computes $R_1 = r_1.P$
$C_2 = X_{T_i} + h(T_1, R_1, C_1)$	$\xrightarrow{C_1, C_2}$	$T_1 = P_{r_R}.C_1.r_1^{-1}$
		$X_{T_i} = C_2 - h(T_1, R_1, C_1)$
		Checks with X_{T_i} in database

requirements such as mutual authentication, forward security, scalability, data integrity, availability and tag anonymity. Moreover, Dinarvand and Barati [3] showed that their protocol could prevent different attacks such as replay attack, cloning attack, Denial of Service (DoS) attack, de-synchronization attack, tag masquerade attack and server spoofing attack. Dinarvand and Barati's protocol is composed of two steps as depicted in the Table 4. For more detail, interested readers can consult the original paper [3]. In this protocol, (q, a, b, P, n) are EC domain parameters. x_S and $P_S = x_S P$ are the server's private/public keys. x_t is a random point that represents the unique identifier for each tag. ID_S is a random number as a unique pseudonym for each tag. K represents the shared secret key between the server and the tag. $\langle x_t, P_S, ID_S \texttt{ and } K \rangle$ represents the tag's memory EC domain parameters and $\langle ID_S, K, x_t \rangle$ are the database's stored domain parameters. In Dinarvand and Barati protocol, the tag and the server are mutually authenticated by the subsequent exchanged messages:

- First, the tag generates a random number $r_2 = Z_n^*$ and calculates $R_2 = r_2 P$, then forwards the messages $\langle R_2, ID_S \rangle$ to the server.
- Upon the reception of this message, the server uses ID_S as an index to get a matching entry in it. If ID_S is not in the database, then the server aborts the session, other-

Table 4 Dinarvand and Barati [3] RFID authentication protocol

Reader/server $\{Server(x_S), Tag(x_t, ID_S, K)\}$	Insecure channel	Tag $\{x_t, ID_S, K, P, P_S\}$
Generates r_1		Generates r_2
Computes $R_1 = r_1 P$	$\xrightarrow{R_1 = r_1 P}$	Computes $R_2 = r_2 P$
$TK_{s1} = r_1 K R_2$		
$TK_{s2} = x_S K R_2$	$\xleftarrow{R_2 = r_2 P, ID_S}$	
$Auth_s = TK_{s1} \oplus TK_{s2} \oplus x_t$		
		$TK_{t1} = r_2 K R_1$
	$\xrightarrow{Auth_s}$	$TK_{t2} = r_2 K P_s$
		$x_t' \stackrel{?}{=} Auth_s \oplus TK_{t1} \oplus TK_{t2}$
		Server is authenticated
$Auth_t \stackrel{?}{=} x_t \oplus 2TK_{s1} \oplus 2TK_{s2}$		if the equality holds
Tag is authenticated	$\xleftarrow{Auth_t}$	$Auth_t = x_t' \oplus 2TK_{t1} \oplus 2TK_{t2}$
if this equality holds		

wise, it extracts the corresponding parameters $\langle K, x_t \rangle$. Here, the key K could be K^{old} or K^{new} according to the received value of ID_S (ID_S^{old} or ID_S^{new}). Next, the server calculates $TK_{s1} = r_1 KR_2$, $TK_{s2} = x_S KR_2$ and $Auth_s = x_t \oplus TK_{s1} \oplus TK_{s2}$, then sends the message $Auth_s$ to the tag.

- Upon receiving $Auth_s$, the tag calculates $TK_{t1} = r_2 KR_1$ and $TK_{t2} = r_2 KP_s$. The tag authenticates the server by verifying if $x_t = Auth_s \oplus TK_{t1} \oplus TK_{t2}$, using its secret and public keys. Next, the tag computes $Auth_t = x_t \oplus 2TK_{t1} \oplus 2TK_{t2}$ and sends it to the server.
- Upon the reception of $Auth_t$, the server authenticates the tag by checking if $Auth_t = x_t \oplus 2TK_{s1} \oplus 2TK_{s2}$.

Finally, after the mutual authentication, the two entities update their secret keys and the pseudonym of the tag as given in the updating phase (subsection 4.3 of [3]).

Moreover, the authors defined a new operation on elliptic curves which is the XOR operation between two EC points expressed as follows: given two EC points (P_1, P_2) represented by their abscissa and ordinate (x_1, y_1) and (x_2, y_2), respectively, so that to obtain the new point $P_3 = (x_3, y_3)$ by the XOR operation between the two points $(P_3 = P_1 \oplus P_2)$, the first and second components of the two points have to be *XORed* as follows; $(x_3 = x_1 \oplus x_2, y_3 = y_1 \oplus y_2)$. Here, we believe that this new operation could break the elliptic curve point addition algebraic properties; as a result it might give a point outside the defined curve.

5.1 Invalid Curve Attack Description

In this subsection, we will show what will occur if an attacker forces an entity, in a given authentication scheme, to compute its scheduled protocol steps using a point outside of the defined curve. We will demonstrate that this disturbance could have serious concerns on the considered authentication scheme. In fact, the dilemma is that the injected point could belong to another elliptic curve with a limited number of points, where the cryptanalysis becomes easy to implement. Consequently, we validate this idea through an efficient cryptanalysis of the most recently proposed ECC protocol designed by Dinarvand and Barati [3]. The different attack steps are given as follows and summarized in the Fig. 1.

1. First, the attacker selects a point P' outside of the used curve that generates a subgroup with small order. (Let n be the order of this point (in our simulations $n = 5$)).
2. The attacker eavesdrops (the man-in-middle attack) on the Dinarvand and Barati scheme and captures the tag response $\langle R_2, ID_S \rangle$ and replaces the point R_2 by the point P' i.e. $\langle P', ID_S \rangle$ and forwards it to the server.
3. Upon receiving the message $\langle P', ID_S \rangle$ first, the server generates a random number r_1 and using its current key K and private key x_S, it computes: $TK_{s1} = r_1 KP'$, $TK_{s2} = x_S KP'$ and $Auth_s = TK_{s1} \oplus TK_{s2} \oplus x_t$. Then, the server sends the message $Auth_s$ to the tag (the attacker in the middle).
4. The attacker intercepts the message $Auth_s$ and he/she withdrawals. Then, he/she proceeds off-line as follows:

   a) As the calculated points TK_{s1}, TK_{s2} will be automatically in the defined small subgroup (because they are calculated using the fake point P') (in our simulations a subgroup of 5 points as given in Table 7).

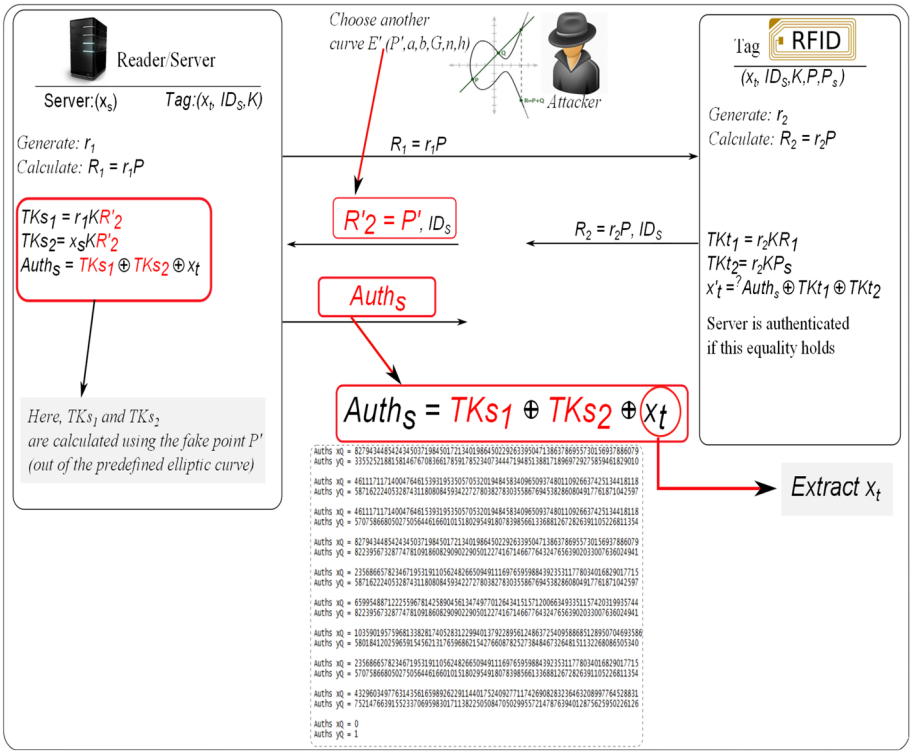

Fig. 1 Invalid curve attack on Dinarvand and Barati protocol

b) The attacker discloses the tag secret x_t by resolving the two equations defined by the XOR operation between the abscissa and the ordinate of TK_{s1}, TK_{s2} and x_t. Here, the attacker has only very limited number of possibilities for the abscissa and the ordinate values.

c) $\left.\begin{array}{l} \{TK_{s1}\}_x \oplus \{TK_{s2}\}_x \oplus \{x_t\}_x = \{Auth_s\}_x \\ \{TK_{s1}\}_y \oplus \{TK_{s2}\}_y \oplus \{x_t\}_y = \{Auth_s\}_y \end{array}\right\}$ where $\{P\}_x$ and $\{P\}_y$ denote the abscissa and the ordinate of the point P, respectively.

5. Once the tag secret x_t is disclosed, the attacker could launch other attacks such as the tag impersonation attack. In other words, the attacker is able to construct a valid message $(Auth_t)$ to deceive the server and passes the tag authentication process.

Finally, via the following example scenario, we will show that even though the protocol uses a standardized elliptic curve (NIST-256 (secp256r1)) which has a huge number of points, a potential attacker is able to force the server to calculate with a given invalid curve with a point outside of the NIST curve. In fact, for this example, the attacker gets only five possible points whatever the values of the secret and the private keys K and x_S used by the protocol (4 points and a point at infinity). In other words, the resulting group of the using invalid curve has small order equal to 5. Besides, knowing all the

points of the invalid elliptic curve E^* and the tag secret x_t, the attacker could launch other attacks such as the tag impersonation attack.

Finally, to fix the identified vulnerabilities, the following recommendations have to be carefully considered:

1. Each entity involved in the protocol has to check for group membership of the computed and exchanged points before processing.
2. Avoid using other operations other of those defined in the ECC theory. For this point, the introduced XOR operation between points has to be replaced by a point addition to keep the algebraic properties valid. Because, the XOR operation between two points could give another point not on the defined secure curve, but on some other, weaker EC.

■ *Simulation of proposed invalid curve attack:* In this subsection, we give an example scenario using NIST-256 ($secp256r_1$) elliptic curve of the form $y^2 = x^3 + ax + b$, where a and b are given in the Table 5 [38]. As we can see, this EC has a huge number of points. Under this curve the ECDLP is known as difficult to resolve. But, we consider the attack described above, where the attacker chooses a small subgroup generated with the fake point P' (given in Table 6) outside of the NIST-256-$secp256r_1$ elliptic curve. In the proposed attack, the attacker forces the server to compute with the selected invalid curve that generates only 5 points summarized in the Table 7. In other words, the values of the computed TK_{s_1} and TK_{s_2} will be certainly from the set of the five possible points given in the Table 7. An example

Table 5 NIST-256 (secp256r1) EC curve parameters (in Hex)

a	FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFC
b	5AC635D8AA3A93E7B3EBBD55769886BC651D06B0CC53B0F63BCE3C3E27D2604B
q	FFFFFFFF00000001000000000000000000000000FFFFFFFFFFFFFFFFFFFFFFFF
n	FFFFFFFF00000000FFFFFFFFFFFFFFFFBCE6FAADA7179E84F3B9CAC2FC632551

Table 6 Invalid point generator's coordinates (in Dec)

$\{P'\}_x$	827943448542434503719845017213401986450229263395047138637869557301569378860 79
$\{P'\}_y$	335525218815814676708366178591785234073444719485138817189697292758594618290 10

Table 7 Generated points using the invalid point generator (in Dec): (TK_{s_1} and TK_{s_2})

P_{1_x}	827943448542434503719845017213401986450229263395047138637869557301569378860 79
P_{1_y}	335525218815814676708366178591785234073444719485138817189697292758594618290 10
P_{2_x}	461117117140047646153931953505705320194845834096509374801109266374251344181 18
P_{2_y}	587162224053287431180808459342272780382783035586769453828608049177618710425 97
P_{3_x}	461117117140047646153931953505705320194845834096509374801109266374251344181 18
P_{3_y}	570758668050275056446166010151802954918078398566133688126728263911052268113 54
P_{4_x}	827943448542434503719845017213401986450229263395047138637869557301569378860 79
P_{4_y}	822395673287747810918608290902290501227416714667764324765390203300763602494 1
P_{5_x}	0
P_{5_y}	1

scenario of the server computation response using the injected invalid point P' is given in the Table 8.

$$\left\{ \begin{array}{l} \{x_t\}_x = \{TK_{s1}\}_x \oplus \{TK_{s2}\}_x \oplus \{Auth_s\}_x \\ \{x_t\}_y = \{TK_{s1}\}_y \oplus \{TK_{s2}\}_y \oplus \{Auth_s\}_y \end{array} \right\}$$

Finally, knowing all possible elements of the couple of equations, the attacker can derive the tag identifier x_t.

5.2 De-synchronization Attack

To guarantee the tag anonymity feature, Dinarvand and Barati are implemented the tag pseudonym technique which consists to update the shared secrets after each successful authentication. Moreover, to avoid the de-synchronization attack, the server should keep the old and the new ID_S in each successful authentication. However, since it is the server that will update its secret parameters lastly, we find that this protocol is vulnerable to de-synchronization attack which can be mounted just by blocking the last message sent by the tag. In this situation, the tag will updates its parameters K and ID_S to new values (as indicated below) and the server will not be able to updates its parameters. Consequently, this attack will prevent the two entities to authenticate each other's in their subsequent authentication sessions.

- Session i: tag $\langle K_i^*, ID_{S_i} \rangle$, server: $\langle K_i^{old}, K_i^{new}, ID_{S_i}^{old}, ID_{S_i}^{new} \rangle$,
- Session $i+1$: tag $\langle K_{i+1}^*, ID_{S_{i+1}} \rangle$, server keeps the same state: $\langle K_i^{old}, K_i^{new}, ID_{S_i}^{old}, ID_{S_i}^{new} \rangle$.

6 Improved Protocol

Numerous authentication schemes proposed in the literature are prone to security traps of every category. Among these pitfalls, the twist-security attacks are one of most underrated attacks in terms of reported rate but with drastic consequences if they come true. However, the attack described in this paper could have been obstructed if the well-known engineering practices and sanity cryptographic recommendations would have been judiciously followed. Although the ECC is adopted in a wide variety of cryptographic protocols, schemes and standards such as EC-Integrated encryption scheme

Table 8 Example scenario; x_t, $TKS_1 = P_1$ and $TKS_2 = P_3$ (in Dec)

x_{t_x}	10236986424965305732272535072374146159990518000490589729877997143782738172526 6
x_{t_y}	10174449111163519051232566840343258974038453050676414884011213722073228318125 4
TKS_{1_x}	82794344854243450371984501721340198645022926339504713863786955730156937886079
TKS_{1_y}	33552521881581467670836617859178523407344471948513881718969729275859461829010
TKS_{2_x}	46111711714004764615393195350570532019484583409650937480110926637425134418118
TKS_{2_y}	57075866805027505644616601015180295491807839856613368812672826391105226811354
$Auth_{s_x}$	22013090476255624509153625380875801387314805669072032545817945659335459867115
$Auth_{s_y}$	96315368300254706249774342444475638816637997615121607459909609515989467762318

(ECIES), EC-Digital signature algorithm (ECDSA), EC-Diffie–Hellman (ECDH), American National Standards Institute (ANSI), NIST Federal Information Processing Standards (FIPS), etc., many notable number of potential flaws continue to be discussed in the literature [7, 39]. Consequently, prudent engineering practices and rigorous security proof analysis together with typical vulnerabilities to avoid, must be conducted when designing new authentication schemes ECC-based. Hereafter, some countermeasures to follow that could overcoming this kind of attack:

1. Carefully check the group membership of different exchanged points before performing any processing.
2. Carefully choose the used curves and validate its various parameters.
3. Implement the Montgomery ladder for the scalar point multiplication computation to avoid side-channel attacks [8, 40].
4. Carefully choose the elliptic curve order (large) to avoid some attacks like, naive attack, Baby Step, Giant Step attack and Pollard's Rho attack [10].
5. Consider the formal security analysis via formal model such BAN (Abadi and Needham) logic, AVISPA model, etc. This kind of security models gives a set guidelines and principles for designing robust cryptographic schemes [39].
6. Consider the formal security analysis via informal model such as the random oracle models.

The improved version of Dinarvand and Barati's protocol, which takes into account these countermeasures to resist to common passive and active attacks, is given in the Table 9. This protocol is composed of two phases, the authentication and updating phases.

Table 9 The improved protocol version

Reader/server (DB) $\{Server(x_S), Tag(x_t, ID_S)\}$	Insecure channel	Tag $\{x_t, ID_S, P, P_S\}$
Generates: $r_1 \in Z_n^*$	$\xrightarrow{\texttt{Query}, r_1}$	Generates $r_2 \in Z_n^*$ Computes $R_2 = r_2 P_S$ $R_3 = r_2 P$ $R_4 = x_t + h(\{R_2\}_x \| \{R_3\}_x \| r_1)$
Calculates: $R_2^* = x_S R_3$ $x_t = R_4 - h(\{R_2^*\}_x \| \{R_3\}_x \| r_1)$ $\langle ID_S, x_t \rangle$ `authenticates the tag`	$\xleftarrow{R_3, R_4, ID_S}$	
Computes: $R_5 = h(x_t \| \{R_2\}_x \| r_1 \| R_4)$	$\xrightarrow{R_5}$	$R_5^* = h(x_t \| \{R_2\}_x \| r_1 \| R_4)$ The server is authenticated if the equality holds
Updating phase: ID_S		Updating phase: ID_S

6.1 Authentication Phase

(1) The server generates a random number r_1 and broadcasts it to the tag.
(2) Upon receiving r_1, first, the tag generates a random number r_2 then, it computes: $R_2 = r_2 P_s$, $R_3 = r_2 P$, $R_4 = x_t + h(\{R_2\}_x || \{R_3\}_x || r_1)$. Then, it forwards $\{R_3, R_4, ID_S\}$ back to the server. Where $\{\cdot\}_x$ denotes the x-coordinate of the given point.
(3) After the reception of the message, the server uses its secret key x_S to calculate: $R_2^* = x_S R_3$ and $x_t = R_4 - h(\{R_2^*\}_x || \{R_3\}_x || r_1)$. Using the received tag pseudonym ID_S, the server fetches x_t from its database. If they are not equal, the server terminates the session; otherwise, the tag is authenticated. The server computes: $R_5 = h(x_t || \{R_2\}_x || r_1 || R_4)$. Then, it transmits $\{R_5\}$ to the tag.
(4) The tag calculates $R_5^* = h(x_t || \{R_2\}_x || r_1 || R_4)$ and verifies if R_5^* is equal to received message R_5. If they are different then, it rejects the server otherwise, it authenticates it and updates the pseudonym ID_S.

6.2 Updating Phase

After each successful mutual authentication session, the server and the tag update the pseudonym of the tag ID_S as follows:

1. The tag:
   $ID_S^* = h(\{R_2\}_x || ID_S || r_1 || R_4)$
   $ID_S \leftarrow ID_S^*$
2. The server:
   If ID_S^{old} is received: $ID_S^{new} = h(\{R_2\}_x || ID_S^{old} || r_1 || R_4)$
   Else, if ID_S^{new} is received: $ID_S^{old} = ID_S^{new}$, $ID_S^{new} = h(\{R_2\}_x || ID_S^{new} || r_1 || R_4)$

As for this new version of the protocol, the main improvements are summarized in the following:

In order to avoid the de-synchronization attack, the server keeps the old and the new version of the tag's pseudonym. In addition, we have ensured that the tag updates its pseudonym lastly. The improved version takes advantages of asymmetric features and excludes the need to use a shared secret contrary to Dinarvand and Barati protocol. Moreover, our improved version incorporates a hash function which allows strengthen the integrity feature and replaying attacks. Eventually, in order to definitely exclude the security concerns related to invalid point attack, the computed points along the protocol are protected using the hash function.

7 Security Analysis

Security analysis is an important step to detect possible security imperfections in authentication schemes. In this section, first, we give the formal security analysis using random oracle against an adversary who attempt to disclose the tag's secret identifier and the server's secret key. Then, via informal security analysis, we show that our improved protocol is secure against several known-attacks and achieves many security requirements.

7.1 Formal Security Analysis

In this section, we carry out the formal security analysis of the improved protocol using a random oracle model as specified in [41]. Thus, we will show that the improved protocol is secure against disclosing the tag's secret identifier and server's secret key. We suppose the following random oracles for the adversary ($\mathcal{A}$):

- **Reveal 1**: *Reveal 1* random oracle will completely output the string x from the corresponding hash value y, knowing that ($y = h(x)$).
- **Reveal 2**: This random oracle will completely output the integer k from a given two points P and $Q = kP$ in $E(GF(q))$.
- **Adversarial model**: We consider the following threat model where $\mathcal{A}$ may have an entire control of the insecure channel between the server and the tag:
  - The adversary $\mathcal{A}$ can eavesdrop on all the transmitted messages between the server and tag.
  - $\mathcal{A}$ can inject his own counterfeit messages.
  - $\mathcal{A}$ can block and modify any exchanged message between the server and the tag.
  - $\mathcal{A}$ could obtain, using different traffic analysis tools, the crucial information to control a specific tag using the captured information from the public channel.

Proposition 1 *Under the ECDLP problem and the one-way hash function $h(\cdot)$ assumptions which closely act as random oracles, our improved protocol is secure against an attacker $\hat{\mathcal{A}}$ disclosing the tag's secret identifier x_t.*

Algorithm 1 $\text{Exp1}^{ECDLP,Hash}_{\hat{\mathcal{A}},I-protocol}$

1-Eavesdrop on the insecure communication channel (r_1, R_3, R_4, R_5, ID^*_S), where $R_3 = r_2P$, $R_4 = x_t + h(\{R_2\}_x||\{R_3\}_x||r_1)$ and $ID^*_S = h(\{R_2\}_x||ID_S||r_1||R_4)$.
2-Call Reveal oracle 2 on input R_3, P. Let $(r_2^{'}) \leftarrow$ `Reveal 2` (R_3)
3-Compute $h(\{r_2P_S\}_x||\{R_3\}_x||r_1)$.
4-Compute $x_t^{'} = R_4 - h(\{r_2P_S\}_x||\{R_3\}_x||r_1)$.
5-Call Reveal oracle 1 on input R_5. Let $(x^{"}t) \leftarrow$ `Reveal 1` (R_5).
if $(x_t^{'} = x_t^{"})$ **then**
  Accept $x_t^{'}$ as the identifier x_t of the tag.
  Return 1 (Success)
else
  Return 0 (Failure)
end if

Proof We aim to build an attacker $\mathcal{A}$ who will have the ability to disclose the tag's secret identifier x_t and the server secret key x_S. $\mathcal{A}$ will use the Reveal Oracle 1 and 2 in the experiment $Exp1^{ECDLP,Hash}_{\hat{\mathcal{A}},I-protocol}$ given in Algorithm 1 for our improved protocol, say I-protocol. We outline the success probability for the experiment $Exp1^{ECDLP,Hash}_{\hat{\mathcal{A}},\text{I-protocol}}$ in Algorithm 1 as

$succ1^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}} = |Pr[Exp1^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}} = 1] - 1|$ and the advantage function for this experiment is expressed by $Adv1^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}}(t, q_1, q_2) = \max_{\hat{\mathcal{A}}}\{Exp1^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}}\}$, where the maximum is obtained over all $\hat{\mathcal{A}}$ during a time t and the number of requests q_1 and q_2 launched to reveal the random oracles Reveal 1 and Reveal 2. Our improved protocol is secure against $\hat{\mathcal{A}}$ for disclosing the tag's secret identifier x_t, if the $Adv1^{ECDLP,Hash}_{\hat{\mathcal{A}},I-protocol}(t, q_1, q_2) \leq \varepsilon$, for any sufficiently insignificant value of $\varepsilon > 0$. Consider the experiment $Exp1^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}}$ given in Algorithm 1 for $\hat{\mathcal{A}}$. Referring to this latter, if $\hat{\mathcal{A}}$ has the ability to solve the ECDLP and invert the one-way hashing function, expressed in Definitions 1 and 2, he/she can correctly reveal the tag's secret identifier x_t, and then win the game. Nevertheless, referring to Definitions 1 and 2, it is a computationally difficult to discover a discrete logarithm r_2, from a given point R_3 and invert the input from a given hashing value, i.e., $Adv^{ECDLP}_{\hat{\mathcal{A}}}(t) \leq \varepsilon$ and $Adv^{Hash}_{\hat{\mathcal{A}}}(t) \leq \varepsilon$, for any sufficiently insignificant $\varepsilon > 0$. Hence, $Adv1^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}}(t, q_1, q_2) \leq \varepsilon$, since it is dependent on $Adv^{ECDLP}_{\hat{\mathcal{A}}}(t)$ and $Adv^{Hash}_{\hat{\mathcal{A}}}(t)$. Therefore, our improved protocol is secure against disclosing the tag's secret identifier x_t by any adversary. □

Proposition 2 *Under the one-way hash function $h(\cdot)$ and ECDLP assumptions which act as random oracles, our improved protocol is secure against an attacker $\hat{\mathcal{A}}$ deriving the server secret key x_s.*

Algorithm 2 $\mathrm{Exp2}^{ECDLP,Hash}_{\hat{\mathcal{A}},I-protocol}$

1-Eavesdrop the authentication message $(P_S,\ R_5)$, where $P_S = x_S P$ and $R_5 = h(x_t||\{R_2\}_x||r_1||\{R_4\}_x)$.
2-Call Reveal oracle 2 on input P_S, P. Let $(x'_s) \leftarrow Reveal2(P_S)$.
3-$R'_2 = x'_S R_3$. Let $(\{r_2 P\}'_x) = \{R_2\}'_x)$.
4-Call Reveal oracle 1 on input R_5. Let $(x'_t, \{R_2\}"_x, r'_1, \{R_4\}'_x) \leftarrow Reveal1(R_5)$
if $\{r_2 P\}'_x = \{R_2\}"_x$ **then**
  Accept x'_S as the secret key x_S of the server.
  Return 1 (Success)
else
  Return 0 (Failure)
end if

Proof We proceed similarly as in the Proposition 1. We build an attacker that can extract the server secret key x_S of the RFID system. $\hat{\mathcal{A}}$ will use the Reveal Oracle 1 and 2 in the experiment $Exp2^{ECDLP,Hash}_{\hat{\mathcal{A}},I-protocol}$ given in Algorithm 2 for our improved protocol. We express the success probability for the experiment $Exp2^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}}$ in Algorithm 2 as $succ2^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}} = |Pr[Exp2^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}} = 1] - 1|$ and the experiment advantage function is specified by $Adv2^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}}(t', q_3, q_4) = \max_{\hat{\mathcal{A}}}\{Exp2^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}}\}$ where the maximum is taken over all $\hat{\mathcal{A}}$ with processing time t' and the number of queries q_3 and q_4 taken to reveal the two random oracles Reveal 1 and Reveal 2. Our improved protocol is safe against the $\hat{\mathcal{A}}$ for extracting the server secret key x_s, if the $Adv2^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}}(t, q_3, q_4) \leq \varepsilon$, for any suffi-

ciently small value $\varepsilon > 0$. Consider the experiment $Exp2^{ECDLP,Hash}_{\hat{\mathcal{A}},\mathtt{I-protocol}}$ specified Algorithm 2 for $\hat{\mathcal{A}}$. According to this experiment, if $\hat{\mathcal{A}}$ has the capability to resolve the ECDLP problem and invert the one-way hash function, described in Definitions 1 and 2, respectively, he/she can correctly disclose the server secret key x_S, and then break the system. However, according to Definitions 1 and 2, it is a computationally difficult to extract the discrete logarithm x_s from a given point P_S and invert the input from a given hashing value, i.e., $Adv^{ECDLP}_{\hat{\mathcal{A}}}(t') \leq \varepsilon$ and $Adv^{Hash}_{\hat{\mathcal{A}}}(t') \leq \varepsilon$ for any sufficiently small $\varepsilon > 0$. Hence, $Adv2^{ECDLP,Hash}_{\hat{\mathcal{A}},I-protocol}(t, q_3, q_4) \leq \varepsilon$, since it is associated to $Adv^{ECDLP}_{\hat{\mathcal{A}}}(t')$ and $Adv^{Hash}_{\hat{\mathcal{A}}}(t')$. Consequently, our improved protocol is safe against disclosing the server secret key x_S by any attacker. □

7.2 Informal Security Analysis

In this section, we show that our improved protocol is resilient against several well-known attacks in the literature and achieves many security requirements under the adversarial model given below.

7.2.1 Provided Functionalities

■ *Mutual Authentication:* A mutual authentication is an important security requirement that allows a bilateral verification between two entities and then avoids the identity usurpation problem. Our proposed improved protocol provides a mutual authentication between the server and the tag. The tag gets authenticated by the verification if the locally computed message $x_t^{'} = R_4 - h(\{R_2^*\}_x||\{R_3\}_x||r_1)$ using the server's private key is identical to the fetched one from its database. Likewise, the server is proved to be genuine by the calculation of the message $R_5^{'} = h(x_t||\{R_2\}_x||r_1||R_4)$ on the tag side, which must be identical to received message R_5.

■ *Scalability:* Scalability property is one of the most desirable features to be integrated in RFID systems. This property describes the capability of the system to properly handle growing workloads. In our improved protocol, the tag identification process is carried out using the received ID_S (step 3), where the server fetches x_t from its database to complete the tag authentication process. Here, the server does not need to search for the corresponding tag content linearly from all existing tags in the database, so that the server takes $O(1)$ to search for x_t. Hence, when the number of tags of the system increases, the improve protocol keeps the same workload. Consequently, the improved authentication protocol provides the scalability property.

■ *Untraceability and anonymity:* Untraceability and anonymity are two important security features that must be incorporated in a given RFID system to guarantee the tag owner privacy, since an RFID tag automatically replies to any received message query. In our improved version, we use the pseudonym technique for the tag identification in DB which is updated every each successful session. Furthermore, the attacker cannot extract the tag unique identifier x_t from the eavesdropped message $\{R_3, R_4, ID_S\}$ since it is never sent openly over the insecure channel. Besides, as all the protocol messages are linked to the generated random numbers r_1 and r_2, this makes the tag response $\{R_3, R_4, ID_S\}$ unpredictable to the attacker, so he/she cannot locate or trace a specific tag

by launching a simple malicious query thanks to all these countermeasures. Therefore, we deduce that our improved version provides untraceability and anonymity.

■ *Availability:* In our improved protocol the tag unique identifier x_t is exchanged in a random message protected by the hash function ($R_4 = x_t + h(\{R_2\}_x||\{R_3\}_x||r_1)$) which means it is not accessible by any attacker. Besides, the identification is carried out using the ID_S identity which is updated after every each successful authentication session. In addition, as the improved protocol avoids the de-synchronization attack, the two entities are continually harmonized. Consequently, the availability is provided in our improved protocol.

7.2.2 Resistance to Different Attacks

■ *Replay attack resisting:* This attack consists to replay some previously intercepted authentication messages to pass the authentication process. In our improved version, the attacker will fail to do that, thanks to the used countermeasures and verification mechanisms. For each session, all the transmitted messages $\{r_1, R_3, R_4, ID_S\}$ and $\{R_5\}$ are constructed and controlled by new random numbers r_1, r_2, $\{R_2\}_x$ and $\{R_3\}_x$. For example, if an attacker ($\mathcal{A}$) intercepts the message $\{R_3, R_4, ID_S\}$ which is transmitted from the tag to the reader. Using this message, $\mathcal{A}$ may try to launch the replay attack. $\mathcal{A}$ replays the $\{R_3, R_4, ID_S\}$ to the server. Upon receiving the message, the server uses its secret key x_S to calculate: $R_2^* = x_S R_3$ and $x_t = R_4 - h(\{R_2^*\}_x||\{R_3\}_x||r_1)$. Here, the server will terminate the session because the ID_S is old and the corresponding x_t in the database is different from the computed value x_t^*. Consequently, the improved protocol is secure against the replay attack.

■ *Forward security resisting:* Our improved protocol guarantees the forward security requirement because even though an attacker gets information of the tag in a given session, he/she cannot get any previous transmitted information since the two random numbers r_1 and r_2 are different for each session.

■ *De-synchronization attack resisting:* Our improved protocol uses the pseudo-identity technique to guarantee the tag owner anonymity, where the tag's identity is updated on both tag and server after each successful authentication session. Nevertheless, an attacker ($\mathcal{A}$) can disrupt the synchronization between two communicating entities by compelling them to update their shared parameters to different values, in such away that they will be unable to recognize each other in their succeeding sessions. As specified by the updating phase, both old and new values of the pseudo-identity ID_S are stored in the database, which avoids the de-synchronization attack. If for example $\mathcal{A}$ obstructs the message R_5 (sent from the server to the tag) to prevent the tag from updating its pseudonym ID_S, the tag authentication in the next session will remain possible thanks to the availability of the old value of ID_S in the database. The only possibility to achieve this task, $\mathcal{A}$ have to impersonate the two entities by providing a correct tag or server responses ($\{R_3, R_4, ID_S\}$) and (R_5), which is an infeasible task because these messages are protected by the ECDLP. Accordingly, our improved protocol is protected against the de-synchronization attack.

■ *Invalid point attack resisting:* Our improved protocol is protected against the invalid point attack since to definitely exclude the security concerns related to this attack, the computed points along our protocol steps are protected using hash function. For example, if an attacker injects a fake point in tag response ($R_3 = r_2 P$, $R_4 = x_t + h(\{R_2\}_x||\{R_3\}_x||r_1)$, ID_S) i.e., R_3', the server will detect this trick via the hash

function which guarantees the integrity $h(\{R_2\}_x||\{R_3\}_x||r_1)$. Therefore, the improved scheme can resist to the invalid point attack.

■ *Disclosure attack resisting:* If an eventual attacker wants to reveal the secret parameters involved in our improved version, i.e., (x_t, x_S), he /she will face to solve the ECDLP and hash function which are computationally intractable. In fact, all the secret parameters and random variables x_S, x_t, r_2, $\{R_3\}_x$ are well preserved using ECDLP and the one-way hash function. For more detail on disclosure attack, see the formal security analysis in Sect. 7.1.

■ *Impersonation attack resisting:* We show in this section how the improved protocol can resist to the impersonation attack. To impersonate the tag, the attacker needs to produce valid messages (r_1, R_5) and sends them to the tag to pass the authentication process. However, the message R_5 requires knowledge of a number of secret parameters such as x_t and x_S which are protected by the ECDLP hard problem and the hash function. Consequently, the attacker cannot cheat the tag to authenticate him/her as a legitimate reader. Similarly, to impersonate the reader, the attacker needs to generate a valid tag response $(\{R_3, R_4, ID_S\})$ to cheat the legitimate reader to pass the authentication process. However, the tag unique identifier x_t and the point R_2 cannot be revealed to $\mathcal{A}$ unless he/she can resolve the ECDLP hard problem and the hash function. Therefore, our improved protocol is resilient against impersonation attacks.

■ *Man-in-the-middle attack resisting:* In our improved protocol, the man-in-the-middle (MITM) attack is declined by mutual authentication between the server and the tag (as shown in Sect. 7.2.1). In other words, an eventual attacker who eavesdrops on messages sent between legitimate server and tag is unable to insert, delete or arbitrarily modify any message sent from one entity to another thanks to the introduced security mechanisms related to the ECDLP and the integrity via the hash function with random numbers. Even if $\mathcal{A}$ obtains R_3, R_4 and ID_S, he/she cannot obtain R_2 and x_t because $R_2 = r_2 P_s$ and $x_t = R_4 - h(\{R_2^*\}_x||\{R_3\}_x||r_1)$. All these parameters are protected using hash function and ECDLP intractable problems, so $\mathcal{A}$ cannot obtain any secret information. Therefore, our improved version is secure against MITM attacks.

■ *Tracking attack resisting:* Our proposed protocol uses the pseudonym technique (ID_S) for the tag identification in the database which is updated every each successful session. Besides, because the random variables r_1, r_2 are different on each session, R_3 and R_4 are also different, hence $\mathcal{A}$ cannot get any fixed information to track. Consequently, the improved protocol can resist the tracking attack.

8 Performance Analysis and Comparison

In this section, we perform a comparative study on security and functionality properties, storage memory, computation and communication costs during the authentication phase between our improved version and the existing authentication protocol of Jin et al.'s protocol [42], Naeem et al. [35] and Dinarvand and Barati [3].

8.1 Comparison of Security and Functionality Properties

In Table 10, our improved protocol is compared with the some earlier ECC-based protocols of Jin et al.'s protocol [42], Naeem et al. [35] and Dinarvand and Barati [3] based

Table 10 Security performance comparison (×: not satisfied, ✓: satisfied, – not mentioned)

Features ↓	Protocols →			
	Jin et al.'s protocol [42]	Naeem et al. protocol [35]	Dinarvand and Barati [3]	Improved protocol
Mutual authentication	✓	✓	✓	✓
Untraceability and anonymity	✓	✓	✓	✓
Scalability	✓	✓	✓	✓
Availability	✓	✓	✓	✓
Invalid point attack	–	–	×	✓
Tracking attack	✓	✓	✓	✓
Man-in-the-middle attack	✓	✓	✓	✓
Disclosure attack	✓	×	✓	✓
De-synchronization attack	✓	✓	×	✓
Replay attack	✓	✓	✓	✓
Tracking attack	✓	✓	✓	✓
Impersonation attack	✓	×	×	✓
Forward security	×	–	✓	✓
Key compromise problem	×	✓	×	✓

on several security and functionality properties such as mutual authentication, scalability, forward security, untraceability and anonymity, availability, invalid point attack, tracking attack, Man-in-the-middle attack, disclosure attack, de-synchronization attack, replay attack, tracking attack, impersonation attack and invalid point attack. It is worth noticing that the proposed protocol by Dinarvand and Barati [3] fails to achieve invalid point and impersonation attacks. In addition, it is not resilient against de-synchronization attack. Naeem et al. [35] is vulnerable to secret identifier disclosure attack and tag impersonation attack. Jin et al.'s protocol [42] does not ensure data integrity and key compromise problem. In summary, the improved version supports additional functionality features and besides offers better security properties as compared to those for other protocols.

8.2 Comparison of Communication Costs

The communication costs of a given authentication protocol is carried out by computing the length of the different conducted messages. Let us consider that the hash function output is 160 bits, identities and random numbers are 160 bits, the length of the elliptic curve is 160 bits (each point (x, y) on the elliptic curve is 320 bits). In our

Table 11 Comparison of communication costs

Protocols ↓	Components →		
	Server (bits)	Tag (bits)	Total (bits)
Jin et al.'s protocol [42]	640	640	1280
Naeem et al. protocol [35]	480	480	960
Dinarvand and Barati [3]	800	640	1440
Our protocol	320	480	800

improved protocol, the exchanged messages include r_1, R_3, R_4, ID_S and R_5 which need $(160 + 320 + 160 + 160) = 800$ bits as total communication cost. In Table 11, we compare the communication cost of the improved protocol with other protocols. It is worth noticing that our improved version needs less communication cost as compared to other protocols while guaranteeing more security services and functionality features.

8.3 Comparison of Computation Costs

Let T_{Hash} and T_{ecm} denote the required time for executing a one-way hash function and the scalar point multiplication operations, respectively. According to [3], the running time of the scalar multiplication (T_{ecm}) on 5 MHz tags is 0.064 s. In addition, it is assumed that $T_{Hash} = 0.00032$ seconds [42]. Further, as the scalar multiplication is the most complex operation in the considered authentication protocols, the running time of other operations such as addition and Xoring can be neglected. The computation cost comparisons with some related works are recapitulated in Table 12. During the authentication and updating phases of the improved protocol, a tag needs the computational cost of $2T_{ecm} + 3T_{Hash}$ while a reader/server requires the computational cost of $T_{ecm} + 3T_{Hash}$. Thus, the total computation cost of our improved protocol is $3T_{ecm} + 6T_{Hash}$. Accordingly, it is noticeable that our improved version consumes less computational cost than Dinarvand and Barati's protocol and in addition, it does not need an extra calculation workload to provide additional functionality and security features.

8.4 Comparison of Storage Memory Costs

The storage memory cost signifies the required space area to store the different parameters of tag and server that are used to achieve the authentication process. In the improved version, the server has to store the common ECC system parameters $\{a, b, P, p \text{ and } n\}$, the server's secret key x_S, the tag's unique identifier x_t and the new and old tag's pseudonym ID_S^{old} and ID_S^{new}. For the tag, it should stock also its common ECC system parameters $\{a, b, P, p \text{ and } n\}$, its unique identifier and pseudonym x_t and ID_S, respectively and the public point P_S. So, the required storage memory costs for the tag and server are as follows: Server: $160 + 160 + 320 + 160 + 160 + 160 + 160w + 160w + 160w = 1120 + 480w$ (bits), where w indicates the number of the tags of the system. Tag: $160 + 160 + 320 + 160 + 160 + 160 + 160 = 1280$ (bits). We then compare the storage space of the improved protocol with other protocols in Table 13. It is observed that the

Table 12 Comparison of computation costs

Protocols ↓	Components →	
	Tag computational cost (ms)	Server computational cost (ms)
Jin et al.'s protocol [42]	$4T_{ecm} + 2T_{Hash} = 256.64$	$2T_{ecm} + 2T_{Hash} = 128.64$
Naeem et al. protocol [35]	$4T_{ecm} + 2T_{Hash} = 256.64$	$4T_{ecm} + 2T_{Hash} = 256.64$
Dinarvand and Barati [3]	$3T_{ecm} = 192$	$3T_{ecm} = 192$
Our protocol	$2T_{ecm} + 3T_{Hash} = 128.96$	$T_{ecm} + 3T_{Hash} = 64.96$

Table 13 Comparison of storage space costs

Protocols ↓	Components →		
	Server (bits)	Tag (bits)	Total (bits)
Jin et al.'s protocol [42]	$1220 + 320w$	1600	$2880 + 480w$
Naeem et al. protocol [35]	$1440 + 160w$	1440	$2880 + 160w$
Dinarvand and Barati [3]	$1120 + 800w$	1760	$2880 + 800w$
Our protocol	$1120 + 480w$	1280	$2400 + 480w$

improved protocol has less storage as compared to those for other protocols. Though, the improved protocol is the only protocol which is able to safeguard the system from numerous possible attacks.

9 Conclusion

The outstanding performance of ECC with its high security level, its small key sizes and its reduced complexity has fascinated numerous researchers in designing secure authentication solutions. In this paper, first, we have shown a series of efficient attacks on some of recently proposed authentication solutions using elliptic curve cryptography. The proposed attacks were in light of flaws related to several causes such as the lack of security maturity within protocols designs, lack of rigorous security verification using appropriate security tools, non-compliance with the fundamental cryptographic principles, etc. Therefore, we have learned that the most effective and simplest way to avoid these kinds of attacks is to judiciously fulfill the well-known engineering practices and sanity cryptographic recommendations and carefully use the formal and informal security analysis via the well-known security models. Moreover, even though the idea behind these kinds of attacks is basic, the attacks could have drastic consequences in case these against-measures are not seriously taken into account by the protocol's designers. Furthermore, an efficient improved protocol was proposed to overcome the discovered flaws with low computational complexity and interesting security features. The security proof of the improved protocol was checked using informal and formal security proof models based on a random oracle model. As for future work, we want to discuss the practical limitations of the improved proposed in terms of computational power and extend it to an anonymous multi-server for IoT applications. This could be a further interesting research.

References

1. Bos, J. W., Halderman, J. A., Heninger, N., Moore, J., Naehrig, M., & Wustrow, E. (2014). In elliptic curve cryptography in practice. *International conference on financial cryptography and data security* (pp. 157–175). New York: Springer.
2. Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., & Verbauwhede, I. (2007). In Public-key cryptography for RFID-tags. In: Fifth annual IEEE international conference on pervasive computing and communications workshops (PerComW'07) IEEE, pp. 217–222.

3. Dinarvand, N., & Barati, H. (2019). An efficient and secure RFID authentication protocol using elliptic curve cryptography. *Wireless Networks*, *25*(1), 415.
4. Wu, F., Li, X., Xu, L., Kumari, S., Karuppiah, M., & Shen, J. (2017). A lightweight and privacy-preserving mutual authentication scheme for wearable devices assisted by cloud server. *Computers and Electrical Engineering*, *63*, 168.
5. 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.
6. 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.
7. Antipa, A., Brown, D., Menezes, A., Struik, R., & Vanstone, S. (2003). Validation of elliptic curve public keys. *International workshop on public key cryptography* (pp. 211–223). New York: Springer.
8. Hankerson, D., & Menezes, A. (2011). *Elliptic curve cryptography*. New York: Springer.
9. Hales, T. C. (2013). The NSA back door to NIST. *Notices of the AMS*, *61*(2), 190.
10. Khoirom, M. S., Laiphrakpam, D. S., & Themrichon, T. (2018). Cryptanalysis of multimedia encryption using elliptic curve cryptography. *Optik*, *168*, 370.
11. Lee, Y. K., Sakiyama, K., Batina, L., & Verbauwhede, I. (2008). Elliptic-curve-based security processor for RFID. *IEEE Transactions on Computers*, *57*(11), 1514.
12. Kaya, S. V., Savaş, E., Levi, A., & Erçetin, Ö. (2009). Public key cryptography based privacy preserving multi-context RFID infrastructure. *Ad Hoc Networks*, *7*(1), 136.
13. Tuyls, P., & Batina, L. (2006). In RFID-tags for anti-counterfeiting. *Cryptographers' track at the RSA conference* (pp. 115–131). New York: Springer.
14. Lee, Y. K., Batina, L., & Verbauwhede, I. (2008). In EC-RAC (ECDLP based randomized access control): provably secure RFID authentication protocol. In: Proceedings of the 2008 IEEE international conference on RFID IEEE, pp. 97–104.
15. 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.
16. Zhao, Z. (2014). A secure RFID authentication protocol for healthcare environments using elliptic curve cryptosystem. *Journal of Medical Systems*, *38*(5), 46.
17. Chou, J. (2014). A secure RFID authentication protocol to enhance patient medication safety using elliptic curve cryptography. *Journal of Supercomputer*,. https://doi.org/10.1007/s11227-013-1073-x.
18. 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.
19. 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.
20. Qu, J., & Tan, X. L. (2014). Two-factor user authentication with key agreement scheme based on elliptic curve cryptosystem. *Journal of Electrical and Computer Engineering*, *2014*
21. Huang, B., Khan, M. K., Wu, L., Muhaya, F. T. B., & He, D. (2015). An efficient remote user authentication with key agreement scheme using elliptic curve cryptography. *Wireless Personal Communications*, *85*(1), 225.
22. Chaudhry, S. A., Naqvi, H., Mahmood, K., Ahmad, H. F., & Khan, M. K. (2017). An improved remote user authentication scheme using elliptic curve cryptography. *Wireless Personal Communications*, *96*(4), 5355.
23. Chen, Y., & Chou, J. S. (2015). ECC-based untraceable authentication for large-scale active-tag RFID systems. *Electronic Commerce Research*, *15*(1), 97.
24. Shen, H., Shen, J., Khan, M. K., & Lee, J. H. (2017). Efficient RFID authentication using elliptic curve cryptography for the internet of things. *Wireless Personal Communications*, *96*(4), 5253.
25. Jin, C., Xu, C., 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.
26. Luo, M., Zhang, Y., Khan, M. K., & He, D. (2017). A secure and efficient identity-based mutual authentication scheme with smart card using elliptic curve cryptography. *International Journal of Communication Systems*, *30*(16), e3333.
27. Islam, S. H., & Biswas, G. (2014). Dynamic id-based remote user mutual authentication scheme with smartcard using elliptic curve cryptography. *Journal of Electronics (China)*, *31*(5), 473.
28. Madhusudhan, R., Hegde, M., & Memon, I. (2018). A secure and enhanced elliptic curve cryptography-based dynamic authentication scheme using smart card. *International Journal of Communication Systems*, *31*(11).
29. Truong, T. T., Tran, M. T., & Duong, A. D. (2014). Enhanced dynamic authentication scheme (edas). *Information Systems Frontiers*, *16*(1), 113.

30. 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.
31. Adhikari, S., Ray, S., Biswas, G. P., & Obaidat, M. S. (2019). Efficient and secure business model for content centric network using elliptic curve cryptography. *International Journal of Communication Systems*, *32*(1), e3839.
32. Naresh, V. S., Sivaranjani, R., & Murthy, N. V. E. S. (2018). Provable secure lightweight hyper elliptic curve-based communication system for wireless sensor networks. *International Journal of Communication Systems*, *31*(15), e3763.
33. Qi, M., & Chen, J. (2018). New robust biometrics-based mutual authentication scheme with key agreement using elliptic curve cryptography. *Multimedia Tools and Applications*, *77*, 1.
34. Sahoo, S. S., Mohanty, S., & Majhi, B. (2019). Improved biometric-based mutual authentication and key agreement scheme using ECC. *Wireless Personal Communications*, *111*, 1–27.
35. Naeem, M. Chaudhry, S. A., Mahmood, K., Karuppiah, M. & Kumari, S. (2019). A scalable and secure RFID mutual authentication protocol using ECC for internet of things. *International Journal of Communication Systems*, p. e3906.
36. Jager, T., Schwenk, J., & Somorovsky, J. (2015). In practical invalid curve attacks on TLS-ECDH. *European Symposium on research in computer security* (pp. 407–425). New York: Springer.
37. 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.
38. Marzouqi, H., Al-Qutayri, M., & Salah K. (2013). In an FPGA implementation of NIST 256 prime field ECC processor. In: Proceedings of the 2013 IEEE 20th international conference on electronics, circuits, and systems (ICECS) IEEE, pp. 493–496.
39. Abadi, M., & Needham, R. (1996). Prudent engineering practice for cryptographic protocols. *IEEE Transactions on Software Engineering*, *22*(1), 6.
40. Joye, M., & Quisquater, J. J. (2001). Hessian elliptic curves and side-channel attacks. *International workshop on cryptographic hardware and embedded systems* (pp. 402–410). New York: Springer.
41. Canetti, R., Goldreich, O., & Halevi, S. (2004). The random oracle methodology, revisited. *Journal of the ACM (JACM)*, *51*(4), 557.
42. Jin, C., Xu, C., Zhang, X., & Li, F. (2016). A secure ECC-based RFID mutual authentication protocol to enhance patient medication safety. *Journal of Medical Systems*, *40*(1), 12.

Mustapha Benssalah is with Signal Processing Laboratory, Military Polytechnic School, Algiers. He received the M.Sc. degree in 2008 from the Military Polytechnic School and his Ph.D. in 2014 from Cergy Pontoise University. He is now working as an associate professor at EMP. His research interests include RFID, cryptography, wireless communication and WBAN security. Dr. Benssalah had published over 30 technical papers in international conferences and journals.

Izza Sarah is with Signal Processing Laboratory, Military Polytechnic School, Algiers. She received the M.Sc. degree in 2017 from the Boumerdes University. Currently she prepares his Ph.D. degree at EMP. Her research interests include WBAN, RFID, cryptography, IoT and WBAN security.

Karim Drouiche obtained his engineering diploma in 1988 at USTA, Algiers, and his Ph.D. in 1993 from École Nationale Supérieure des Télécommunications, Paris, France. Currently, he is a researcher at Cergy University, France. His main interests are signal processing, statistics and RFID.