Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 32 (2014) 198 – 206

Procedia Computer Science

5th International Conference on Ambient Systems, Networks and Technologies (ANT-2014)

An Elliptic Curve-based Mutual Authentication Scheme for RFID Implant Systems

Sanaz Rahimi Moosavi*, Ethiopia Nigussie, Seppo Virtanen, Jouni Isoaho

Department of Information Technology, University of Turku, 20014 Turku, Finland

Abstract

In this paper, a secure mutual authentication scheme for an RFID implant system is developed. An insecure communication channel between a tag and a reader makes the RFID implant system vulnerable to attacks and endangers the user's safety and privacy. The proposed scheme relies on elliptic curve cryptography and the D-Quark lightweight hash design. Compared to the available public-key cryptosystems, elliptic curve-based cryptosystems are the best choice due to their small key sizes as well as their efficiency in computations. The D-Quark lightweight hash design is tailored for resource constrained pervasive devices, cost, and performance. The security analysis of the proposed authentication scheme revealed that it is secure against the relevant threat models and provides a higher security level than related work found in the literature. The computational performance comparison shows that our work has 48% less communication overhead compared to existing similar schemes. It also requires 24% less total memory than the other approaches. The required computational time of our scheme is generally similar to other existing schemes. Hence, the presented scheme is a well-suited choice for providing security for the resource-constrained RFID implant systems.

Selection and Peer-review under responsibility of the Program Chairs.
Keywords: RFID implant system; IoT; security; healthcare; authentication and identification; elliptic curve cryptography

1. Introduction

Internet of Things (IoT) is emerging as an attractive future networking paradigm. The new generation of Internet is an IPv6 network interconnecting traditional computers and a large number of smart objects or networks. IoT consists of smart objects and low-power networks such as Wireless Sensor Networks (WSNs) [1], Radio Frequency Identification (RFID) networks [2], Body Area Networks (BANs) [3], and actuators. IoT provides an integration approach for all physical objects that contain embedded technology to be coherently connected and enables them to communicate, sense and interact with the physical world. Thus, information of any object or service will be accessible in a systematic way. This results in the generation of enormous amounts of data which have to be stored, communicated, processed and presented in a seamless, secure, and easily interoperable manner. IoT has many potential applications in our everyday life: a smart home where no energy is wasted, productive businesses where offices turn into smart and interactive environments and factories transmit production-related information in real-time, and a proactive healthcare system

* Corresponding author. Tel.: +3-582-333-8647.
E-mail address: saramo@utu.fi

Selection and Peer-review under responsibility of the Program Chairs.
doi:10.1016/j.procs.2014.05.415

that reduces costs without compromising the quality of health services. In the near future, most of the population will benefit from the BANs. The combination of "things" such as sensors, wireless radio, RFIDs and 6LoWPAN[1] will enhance monitoring methods and measurements of vital functions such as temperature, blood pressure, heart rate, cholesterol levels and blood glucose. IoT services and applications will also have a great impact on independent living of elderly population by detecting their chronic diseases and activities of daily living using wearable and ambient sensors.

An RFID implant system is one of the components of IoT-based healthcare solutions. It can be introduced into the human's body in order to store health and medical records that can save a patient's life in emergency situations. In such a system, the identification process can be done completely automatically and there is no need to type, confirm or remember passwords. People who suffer from cancer, diabetes, coronary heart disease, cognitive impairments, seizure disorders and Alzheimer's are the best choice to benefit from the RFID implant system. It was approved by the U.S. Food and Drug Administration (FDA) in 2004 for clinical use[4]. VeriMed, the commercial application of VeriChip RFID implants, has been designed to be used for patient identification in healthcare. An RFID Implant system, consists of three components: *Implantable RFID Tags*, *RFID Reader(s)*, and *Back-end Database Server*. Implantable RFID Tags are medical devices embedded into a human body through a surgical procedure. The commercial implantable tags used for patients are passive tags, they do not need any built-in battery and their operation relies on energy that is emitted by an external RFID reader. As these tags do not have any moving parts, once implanted they can remain activated for more than 10 years[4]. An RFID Reader communicates with the implantable RFID tags and the back-end database server. In an RFID implant system, the reader runs queries to the tags. The essential information associated to the owner of the tag is kept in a back-end database server for the subsequent utilization.

The communication channel between the tag and the reader is insecure and our goal is to make this channel secure. Security is a major concern wherever networks are deployed at large scale. Due to direct involvement of humans in IoT healthcare, providing robust and secure data communication among healthcare sensors, caregivers and patients carrying RFID tags are crucial. Whether the data gathered from patients or individuals are obtained with the consent of the person or without it due to the need by the system, misuse or privacy concerns may restrict people from taking advantage of the full benefits from the system. An RFID implant system in healthcare is a resource-constrained system and it requires efficient and optimized security solutions where the data concerning the patients is secured with Confidentiality, Integrity, and Authentication (CIA). Without strong security foundations, attacks and malfunctions in the RFID implant system will outweigh any of its benefits. Conventional security and protection mechanisms including existing cryptographic solutions and privacy assurance methods that have been proposed to the RFID systems in general, cannot be re-used. This is because of resource constraints, different security level requirements, and the system architecture of an RFID implant system. Thus, an RFID implant system requires a robust, optimized, and lightweight security framework to fulfill the security level requirement and hardware footprint constraints efficiently.

In this paper, we propose a secure elliptic curve-based mutual authentication scheme for RFID implant systems that can be used in healthcare applications. Compared to related work proposed for RFID systems in general, our proposed scheme is more efficient in terms of communication overhead and memory requirement while offering higher level of security. In previous work[4], we have discussed that the hardware footprint, power consumption limitations, and security level requirements of RFID implant systems are different from mainstream applications of RFID due to the delicate use cases and safety-critical specifications. Thus, security solutions being proposed in this regard must be optimized based on characteristic restrictions and requirements of RFID implant systems.

The remainder of this paper is organized as follows: Section 2 provides an overview of related work. Section 3 discusses the security requirements and threat models of RFID implant systems. Section 4 presents our proposed ECC-based mutual authentication scheme to the RFID implant systems. Section 5 provides a comprehensive security and computational performance analysis of our scheme. In this section, the comparison of this work with similar existing approaches is also presented. Finally, Section 6 concludes the paper.

2. Related Work

Several communication security schemes, either ECC-based or non ECC-based, have been proposed in literature to solve security and privacy issues in RFID systems. In this section, we examine some of the existing ECC-based security schemes for RFID systems since our proposed authentication scheme also relies on ECC.

In 2006, Tuyls *et al.*[5] proposed an ECC-based RFID identification scheme using the Schnorr identification protocol. They claimed that their scheme can resist against tag counterfeiting. However, in 2008 Lee *et al.*[6] presented that this scheme suffers from the location tracking attack as well as forward security. In such a scheme when an adversary can compute the public key $X(= -t.P)$ of a tag, it can benefit from X in order to get access to other information related to the tag. Lack of scalability is another problem of the Tuyls *et al.*'s scheme. This is because at each time a tag needs to be identified, the reader should fetch the tag's public key from the database server to verify it. This means that the reader requires to perform linear search to identify each tag. By doing so, considerable computational cost will be imposed to the whole system.

In 2007, Batina *et al.*[7] proposed an ECC-based RFID identification scheme based on Okamoto's authentication algorithm. Although they claimed that their scheme can resist against active attacks, in 2008, Lee *et al.*[8] asserted that this scheme suffers from tracking as well as a forward secrecy problem. Lee *et al.* in 2010[6], proposed an ECC-based RFID authentication scheme in order to address the existing tracking problems in[5] and[7]. Nevertheless, in the mentioned schemes, the authors merely consider tag to reader identification, excluding reader to tag authentication[9]. This causes tags to reply to any malicious query being sent by an adversary. The major reason is that tags are not capable of confirming to whom they are talking to. In 2011, Zhang *et al.*[10] proposed an ECC-based randomized key scheme in order to improve Tuyls *et al.*'s and Lee *et al.*'s schemes. Although their scheme is secure against relevant attacks concerning the RFID systems, it still not capable of performing mutual authentication. In 2013, Liao *et al.*[9] proposed a secure ECC-based authentication scheme integrated with ID-verifier transfer protocol. Similar to Zhang *et al.*'s work, Lial *et al.*'s scheme achieves the required security level of RFID systems. However, their tag identification scheme lacks performance efficiency in terms of the tag's computation time and its memory requirement.

Based on the above-mentioned weaknesses and vulnerabilities, we believe that there still is lack of secure and efficient authentication scheme for RFID implant systems. In addition, hardware footprint and power consumption limitations and security level requirements of RFID implant systems differ from mainstream applications of RFID due to the safety-critical specifications and delicate use cases.

3. Security Requirements and Threat Models of RFID Implant Systems

Security requirements and threat models of RFID implant systems in healthcare will be discussed in this section. First, we present the security requirements of RFID implant systems and then we introduce the most relevant threat and attack models.

3.1. Security Requirements

When designing an authentication scheme, the security requirements of an RFID implant system need to be well defined. The security requirements can be defined in terms of mutual authentication, confidentiality, integrity, availability, and forward security.

Mutual Authentication: mutual authentication is a scheme where both sides, a tag and a reader, authenticate each other. Unlike the most common authentication schemes, where just a party authenticates another party, mutual authentication is critical if each of the parties is involved in a communication. Without having mutual authentication in an RFID system, either of the parties can falsify their identities.

Confidentiality: all of the secret information concerning the RFID implant system are securely transmitted during all communications. To ensure the confidentiality, one of the two parties, either the tag or the reader, transmit the encrypted information and just the other one can decrypt it.

Data Integrity: the data collected and stored by a device must be protected from tampering by unauthorized parties.

Availability: the device should be resilient to Denial of Service (DoS) attacks, and a malicious entity should not be able to affect the operational capabilities of the device in any way.

Forward Security: The property of forward security ensures that the revelation of the tag's secret information will not threaten the security of previously transmitted information.

3.2. Threat Models

In the following, we sketch some of the most relevant attack models concerning the RFID implant systems.

Unauthorized Location Tracking: such an attack is directed against the privacy of tagged people in order to track their activities. For example, the activity of a person who is implanted with an RFID tag can be tracked by any unauthorized person. This will happen if an adversary pretends to be a trusted component of an RFID implant system. By doing so, the adversary will be able to track an implanted person and access his/her confidential information, or implement a counterfeiting attack to probing the information that he captured from the tag.

Eavesdropping Attack: in an RFID implant system, with an eavesdropping attack the adversary can capture the communications conveyed between the tag and the reader. In this type of attack the adversary does not need to communicate with the RFID tag. He/she only captures the transmitted signals using Radio Frequency (RF) equipment. The information gained by the adversary can be utilized later against the privacy of the implanted users.

Impersonation Attack: to impersonate either a tag or a reader in an RFID implant system. In this system, when there is no authentication scheme to prove that the tag/reader is authentic, it is possible that the adversary implements the impersonation attack against the whole system and utilizes the gained information (e.g. medical history of a patient) in malicious ways. As a result, such a system requires a robust and secure authentication scheme to verify that the tag/reader is valid.

Replay Attack: all messages transmitted between a tag and a reader can be captured and saved by an adversary. Then, he/she can transmit the intercepted information in an attempt to deceive an authorized device and pass the authentication phase. For example, an illegal reader may listen and capture the information transmitted between a tag and an unauthorized reader, and then replay the communication in order to gain the same result that a legal reader and tag can benefit from.

4. The Proposed Authentication Scheme

This section presents an ECC-based mutual authentication scheme that satisfies the security requirements in an RFID implant system. A mutual authentication scheme enables the communicating parties, a tag and a reader, to respectively verify and ensure each other's identity. Later, it will be shown that the proposed communication scheme is secure against several relevant attacks and compared to related work has less communication overhead and requires less memory to perform the authentication.

The proposed scheme consists of three phases: 1. the reader authentication and verification phase, 2. the tag identification phase, and 3. the tag verification phase. In the proposed scheme, we suppose that the communication between the reader and the back-end database server is done through a secure channel, while communication between the RFID implant tag and the reader is not secure. Our proposed ECC-based mutual authentication scheme will provide a secure channel between the tag and the reader in such a way that they can communicate with each other securely and efficiently. Before describing the three mentioned phases, in Definition 1, we first introduce parameters and notations used in our proposed scheme.

4.1. Reader Authentication and Verification (Phase 1)

The reader authentication and verification phase of our proposed scheme relies on Elliptic Curve Discrete Logarithm Problem (ECDLP)[11]. In this phase, the reader chooses a random number $r_1 \in Z_n$ and computes $R_1 = r_1.P$ as its public key. Next, it initializes its counter value i_1 to one and sends both R_1 and i_1 to the tag. It then increments the value i_1 by r_1. Upon receiving the message, the tag checks whether i_2 (which is initialized to zero) is greater than i_1. If the condition holds, it replaces i_2 by i_1 and selects a random number $r_2 \in Z_n$. Then, the tag computes $r_3 = X(r_2.P) * Y(R_1)$ where * is a non-algebraic operation over the abscissa of $(r_2.P)$ and the ordinate of R_1 (This operation can be either modular addition if the field is binary or a bitwise *xor* if the field is prime) and it sends the value r_3 to the reader. After receiving r_3, the reader computes $R_2 = r_1.ID_t + r_3.s_3$ and sends the value R_2 to the tag.

Definition 1 Parameters and Notations Used in This Work

G : a group of order q on an elliptic curve having the order n,

P : a primitive element or the base point of G,

s_1 , s_2: each tag keeps two secret points s_1 , $s_2 \in E(F_g)$, which will change over time. These secret points will be varied each time the tag is successfully identified,

ID_t : the tag's identification number or ID,

s_3 : each reader keeps a secret point $s_3 \in Z_n$, which will change over time. This secret point will be varied each time the reader is successfully authenticated,

$ID_r = s_3.P$: the reader's public key,

r_s, i_1, i_2 : random numbers in Z_n,

h : a lightweight hash function,

$(d\,, c)$: a signature generated by the tag during its identification phase,

Algorithm 1 Pseudo-code of Reader Authentication and Verification

Inputs: (r_1, R_1): The private key and the public key of the reader. i_1: The reader's counter value.

Output: Determine whether the reader authentic or not?

Body:

```
1: i_1 ← 1;
2: for i = 1 to n − 1 do
3:     r_1 ← i;
4:     R_1 ← r_1.P;
5:     i_1 ← i_1 + r_1;
6: end for
7: send R_1 to the tag;
8: for j = 1 to n − 1 do
9:     if i_1 ≥ i_2 then
10:        i_2 ← i_1;
11:        r_3 ← X(r_2.P) * Y(R_1);
12:    end if
13: end for
14: Tag send r_3 to the reader;
15: Reader computes R_2 ← r_1.ID_t + r_3.s_3 and sends the value
    R_2 to the tag;
16: if (R_2 − r_1.ID_t)r_3^{-1}.P = ID_r  then
17:     return Success;
18: end if
```

Finally, the tag checks whether $(R_2 - r_1.ID_t)r_3^{-1}.P = ID_r$ holds. Then, the tag verifies that the reader is authentic. Algorithm 1 shows how the authentication and verification of the reader is done in this scheme.

4.2. Tag Identification (Phase 2)

Both the tag identification and the tag verification phases of our proposed scheme rely on Elliptic Curve Digital Signature Algorithm (ECDSA)[11] using Quark lightweight hash design. Quark is one of the most recent lightweight hash designs and it was first proposed by Aumasson *et al.* in 2013[12]. The design of Quark lightweight hash relies on non-linear Boolean functions and bit shift registers. Therefore, not only its implementation becomes feasible, but also, the circuit area requirements of this hash design are well suited for implantable medical devices. In addition, a digital signature offers identification along with integrity and non-repudiation. In our previous work, we stated that due to the resource limitations and the delicate use cases of the RFID implant systems, the need for lightweight cryptographic hash designs has to be carefully considered. That is the reason why in our proposed ECC-based tag identification algorithm, we utilized the D-Quark (one of the flavors of Quark) lightweight hash design rather than the general purpose hash designs (e.g. SHA-1[13] and SHA-3[14])[15].

In the tag identification phase of our proposed scheme, the tag's initial secret point is $s_1 \in E(F_g)$ from which the next secret point s_2 and ID_t will be computed. To generate the second secret point, the tag computes $s_2 = f(X(s_1)).P$. Obtaining the first secret point from the second is difficult, as it requires the computation of an elliptic discrete logarithm. Since the second key is generated from the second key, our scheme provides forward security.

For the sake of efficiency, the function f should be selected in a manner that avoids large Hamming weights for s_2, assuring that the computation of $s_2.P$ will be fast without compromising security[16]. Once the generation of the second secret point s_2 is done, the tag selects a random integer $k \in Z_g$ and computes a curve point $(x, y) = k.G$. In order to send its digital signed message (d, c) to the reader, the tag computes $d = x \bmod n$. If $d = 0$, the tag starts to select another random number $k \in Z_g$ and computes the next curve point. The tag computes its $ID_t = Mb(X(s_1)) * Mb(X(s_2)).P$ where Mb will output some middle bits of the input values. The operand * is a non-algebraic operation $\in F_g$ done over the abscissa of the first and the second secret points (This operation is modular addition as the field is binary). Then, the tag computes $c = k(hash(ID_t) + X(s_1).d)$. Here again, if the computed $c = 0$, the tag will start the algorithm by selecting another random integer k. Finally, the tag sends the computed values (d, c) and (ID_t) to the reader. Algorithm 2 shows the pseudo-code of the tag identification phase of the proposed scheme.

4.3. Tag Verification (Phase 3)

In this phase, in order to verify the tag is authentic the reader selects a random integer $r_s \in Z_n$ and it computes its public key $p_r = r_s.P$. for $j \in [1, n-1]$, the reader checks whether $d, c \in Z_n$. If the result is valid, the reader calculates

Algorithm 2 Pseudo-code of Tag Identification

Inputs: $r_s \in Z_n$: a random integer (sent from the reader's side) and a hello request. s_1: tag's first secret point.

Output: ID_t: Tag's *ID* and (d, c): the tag's digital signature.

Body:

1: The tag checks:
2: **if** $r_s \neq 0$ **then**
3: $s_2 = f(X(s_1)).P$;
4: **for** $i = 1$ to $n - 1$ **do**
5: The tag selects a random integer k and computes the curve point $(x, y) = k.G$;
6: The tag computes $d = x \bmod n$;
7: **if** $d = 0$ **then**
goto 3;
8: **end if**
9: The tag computes the value of its *ID* as: $ID_t = (Mb(X(s_1)) * Mb(X(s_2))).P$;
10: Then, the tag computes: $c = k.(Hash(ID_t) + X(s_1) * d) \bmod n$;
11: **if** $c = 0$ **then**
goto 3;
12: **end if**
13: send $ID_t, (d, c)$ to the reader;
14: **end for**
15: **end if**

Algorithm 3 Pseudo-code of Tag Verification

Inputs: ID_t: The tag's *ID* and (d, c): the tag's digital signature.

Output: Determine whether the tag is authentic or not?

Body:

1: **for** $j = 1$ to $n - 1$ **do**
2: **if** $d, c \in [1, n - 1]$ **then**
3: $h = Hash(ID_t)$;
4: z= left most bit of h;
5: $w = c^{-1} \bmod n$;
6: $u_1 = zw \bmod n$;
7: $u_2 = dw \bmod n$;
8: curve point $(x, y) = u_1.P + p_r$;
9: **end if**
10: **end for**
11: **if** $r = x \bmod n$ **then**
12: **return** *Success*;
13: **end if**

$h = Hash(ID_t)$, where Hash is the same Quark lightweight hash function that is used in the previous phase to generate the tag's signature. Once the hash value of (ID_t) is computed, the reader selects the leftmost bit of h and denotes it as z. Then, the reader calculates the values w, u_1, u_2 exactly as shown in Algorithm 3. Based on the calculated values, the reader computes the curve point $(x, y) = u_1.P + p_r$. Finally, the reader will accept the tag's signature as a valid one if the equation $r = x \bmod n$ holds.

5. Security and Computational Performance Analysis of The Proposed Authentication Scheme

In this section, we will analyze the security and performance of the proposed scheme in order to verify whether the essential requirements have been satisfied.

5.1. Security Analysis

In the following, we analyze our proposed scheme against some of the most relevant attacks. As it is mentioned in section 4, we assume that the communication between the reader and the back-end database server is done through a secure channel, while communication between the implantable tag and the reader is not secure.

Mutual Authentication: in the reader authentication phase of our proposed scheme, to verify that the reader is legal, the tag computes whether $(R_2 - r_1.ID_t)r_3^{-1}.P = ID_r$. Conversely, to verify whether the tag is authentic (based on its transmitted (ID_t) and the digital signed message), the reader checks if $r = x \bmod n$ holds. This is how mutual authentication is achieved in our proposed scheme.

Availability: in our scheme, since the tag and the reader change their secret points s_1, s_2, and s_3 once they are successfully authenticated, it is not possible that an adversary performs a denial of service attack.

Forward Security: in our scheme, if an adversary tries to decrypt some of the information that he has eavesdropped, for example the tag's second secret key s_2, he/she will not benefit from the gained information. Obtaining the first secret key from the second one will require a solution to the ECDSA, which is not easily possible.

Unauthorized Tracking of The Tag: In our proposed scheme, the only public information concerning the tag is its *ID*. In the tag identification phase, it was shown that the value of the tag's *ID* results from the product of a non-algebraic operation done over some middle bits of the abscissa of the first and second secret keys of the tag. As a result, it is impossible to compute and obtain the tag's secret keys from its current *ID*. The main reason is that obtaining the secret points implies the computation of the elliptic curve discrete logarithm problem. Since solving the discrete logarithm problem is as hard as the integer factorization problem, this problem cannot be solved effortlessly. Thus far, there has not been any polynomial time algorithm proposed to solve discrete logarithm problems.

Table 1. Security properties comparison with the available ECC-based designs.

	Batina et al.[7]	Zhang et al.[10]	Liao et al.[9]	Lee et al.[6]	This work
Tracking of the tag	No	Yes	Yes	Yes	Yes
Eavesdropping attack	Yes	Yes	Yes	Yes	Yes
Impersonation attack	No	Yes	Yes	Yes	Yes
Replay attack	Yes	Yes	Yes	Yes	Yes
Forward security	No	Yes	Yes	Yes	Yes
Anonymity	No	Yes	Yes	No	Yes
Mutual authentication	No	No	Yes	No	Yes
Availability	Yes	Yes	Yes	Yes	Yes

Eavesdropping Attack: In our scheme, from one hand, in the tag identification phase, if an adversary tries to guess the tag's secrets s_1 and s_2, the only public information concerning it is ID. As it was discussed earlier, the bits of the tag's ID result from a non-algebraic operation done over some middle bits of the abscissa of two different secret points s_1 and s_2. Thus, it is computationally unfeasible to obtain the secret from its ID. On the other hand, in the digital signature generation section, if an adversary could guess the value d, it cannot obtain the value c effortlessly. This value is also generated from a non-algebraic operation done over the abscissa of the secret point s_1 and the value d. The gained result will be added to the hash value of ID_t and multiplied by a random number k. Such an operation cannot be easily computed by an adversary as it requires to compute the discrete logarithm problem that is not computationally feasible. For the same reason, in the reader authentication phase, even if an adversary could guess one of the values R_1 or R_2 or r_3, he/she still cannot easily obtain other secure information related to the reader. Based on the discussion above, the adversary also cannot implement any *Replay Attack*.

Impersonation Attack: concerning this type of attack, we consider two different scenarios:

- *Impersonation of the reader*: here, if an adversary tries to impersonate the reader, he/she will fail. This is because if the attacker tries to impersonate as a fake reader to the tag, he/she must compute R_1 and at the same time try to guess the value r_2 (which is not easily feasible). Nevertheless, without the reader's computed value $R_2 = r_1.ID_t + r_3.s_1$, the adversary (fake reader) cannot compute $(R_2 - r_1.ID_t)r_3^{-1}.P = ID_r$ to verify whether the reader is authentic.
- *Impersonation of the tag*: in order to impersonate the tag of our proposed scheme, an adversary needs to have an access to the tag's secrets s_1 and s_2 and as it was presented earlier in this section, the values of the secret keys cannot be acquired from the public information of the system ID_t.

Based on the discussion above, our proposed scheme is secure and robust against relevant attacks related to RFID systems. The security properties comparison of our proposed scheme and other ECC-based related works is presented in Table 1. In the table, the term "Yes" states that the available ECC-based designs are secure against the above-mentioned attacks. "No" indicates that those ECC-based designs are not robust and secure against the specified attacks and the threats models. From the security point of view, as the table shows, Lee *et al.*'s and Zhang *et al.*'s schemes have almost the same properties against different types of attacks. Nevertheless, their major disadvantage is that they do not have any security solution for mutual authentication. Although the security properties of our scheme are similar to Liao *et al.*'s scheme, in the next section we will show that our scheme provides better efficiency in terms of computational cost, total memory required, and communication overhead.

5.2. Computational Performance Analysis

As it was presented earlier, implantable tags are resource-constrained pervasive devices. They are tiny in terms of size and computational capacity. Hence, it is important to analyze the performance of the authentication scheme to ensure that the overhead is minimal. Such an analysis can be done based on various criteria including computational cost, memory requirements, and communication overhead. In this work, we mainly focus on the performance analysis of implantable tags since RFID readers are known to be robust devices[9].

As a common cryptographic primitive, we utilize standardized 163-bit elliptic curve domain parameters recommended by National Institute of Standard and Technology (NIST). The parameters are defined over the binary finite field F(2^{163}). We utilize ECDSA algorithm having the coordinate (x, y). As a reminder, the elliptic curve domain parameters over F(2^m) are specified by the tuple $T = (m, f(x), a, b, G, n, h)$ where $m = 163$ and the representation of F(2^{163}) is defined by, $f(x) = x^{163} + x^7 + x^6 + x^3 + 1$ [11]. In their work, Godor *et al.* [17] measured the **computational time** required for scalar multiplication of 163-bit point elliptic curve, the SHA-1 hash function [13], and the Advanced Encryption Standard (AES) [18] algorithm. As an environment to measure the computational time for the mentioned cryptography algorithms, they used an Intel Core2 CPU T5500 1.66 GHz having 1GB RAM. Based on the results deduced from their work, at the frequency of 5 MHz, the computational time required to compute the scalar multiplication of 163-bit point elliptic curve is 64 ms [19].

Kumar *et al.* [20] presented that in High Frequencies (HF) such as 13.56 MHz, which is normally the frequency used in most RFID applications (e.g. smart cards, access control and libraries), the scalar multiplication of 163-bit point elliptic curve is done in 31.8 ms. Nevertheless, such a frequency and other higher frequencies have not been approved by the U.S. Food and Drug Administration (FDA) neither for Implantable Medical Device (IMD) applications nor human identification purposes [4]. In Low Frequencies (LF) such as 323 KHz, 243 ms computational time is needed for completing the scalar multiplication, which is too long compared to 64 ms. Hence, we evaluate the performance of our proposed ECC-based scheme at 5 MHz frequency. In addition to reducing the computation time, this allow us to make a fair comparison with related work and also to take into account the restriction of the FDA.

In our proposed scheme, we outline the **storage requirement** by considering the tag's memory including its public key and private key. The private key is denoted as the tag's secret keys s_1 and s_2 and the public key is the tag's public key ID_t. In the proposed scheme, the required memory consists of (ID_t,s_1,s_2) where the ID_t needs 163 bits memory and s_1 and s_2 together require 326 bits memory. So the total required memory is: 62 bytes= 163 bits + 326 bits. Table 2 presents the performance comparison of our proposed tag identification scheme with related work.

The computational cost of our proposed tag identification algorithm includes three scalar points and it is computed as: (64 ms * 3 = 192 ms). Thus, our tag identification algorithm requires 192 ms to compute the multiplication of the three scalar points of the scheme. As Table 2 presents, when the number of ECC scalar point multiplication (ECm) increases, it will have a direct impact to the time required to do this multiplication. Hence, in real-time systems, the system will require considerable time until the authentication is performed successfully.

Table 2. Performance comparison with the available ECC-based designs.

	Batina et al. [7]	Zhang et al. [10]	Liao et al. [9]	Lee et al. [6]	This work
Communication-overhead	82	82	82	82	42
(ECm,hash)	(2,0)	(3,0)	(5,0)	(3,0)	(3,1)
Public-key memory	41	41	61	41	21
Private-key memory	41	41	41	41	41
Total memory (byte)	82	82	102	82	62
Computational time (ms)	128	192	320	192	192

To evaluate the **communication overhead** of our algorithm, the information that is transmitted between the tag and the reader during the tag identification phase needs to be considered. Hence, in our scheme we evaluated the value of communication overhead based on the messages $ID_t, (d, c)$ exchanged between the tag and the reader in the mentioned phase. Here, the overhead is 42 bytes and it is evaluated as: (163 * 2 = 326/8 bytes).

The communication overhead of the proposed elliptic curve-based mutual authentication scheme is compared with the other schemes. The proposed scheme achieves 48% reduction in communication overhead compared to the Batina *et al.*'s, the Zhang *et al.*'s, the Liao *et al.*'s and the Lee *et al.*'s schemes, respectively. In case of total memory, it requires 24% less memory than the Batina *et al.*'s, the Zhang *et al.*'s and the Lee *et al.*'s schemes. Compared to Liao *et al.*'s scheme, the proposed scheme requires 39% less storage. Our proposed scheme needs the same amount of computation time as Zhang *et al.*'s and the Lee *et al.*'s to perform the authentication between the tag and the reader. Compared to Liao *et al.*'s scheme, the computational time of the proposed scheme reduces by 60%. However, the computation time increases by 50% compared to Batina *et al.*'s scheme.

6. Conclusion

In this paper, we presented a novel secure elliptic curve-based mutual authentication scheme for RFID implant systems. To the best of our knowledge, previously proposed elliptic curve-based authentication schemes, concerning RFID systems in general, cannot fully fulfill the essential security and performance requirements of RFID implant systems. Most of the earlier proposed solutions were not secure against the most relevant attacks of the RFID systems or they were not capable of performing mutual authentication between a tag and a reader. The proposed mutual authentication scheme relies on elliptic curve cryptography. An elliptic curve cryptosystem is more efficient in terms of key sizes and required computations than conventional public key cryptosystems. In the proposed scheme, reader authentication and verification is performed based on ECDLP. While tag identification and tag verification phases rely on ECDSA using Quark lightweight hash. We proved that our proposed scheme is secure against the relevant attacks and also ensures a higher security level than related work found in the literature. In addition, we carried out computational performance analysis of our proposed scheme and the analysis results show that our elliptic curve-based mutual authentication scheme has 48% less communication overhead than similar available schemes. It also requires 24-39% less total memory than the compared existing schemes. Based on the results presented in this paper, we conclude that the proposed scheme has the appropriate features for use in RFID implant systems. We believe that our scheme is not just limited to RFID implant systems, it can also be applied to any application of IoT that requires secure and efficient authentication.

References

1. G. Pottie. Wireless Sensor Networks. In *Information Theory Workshop*, pages 139–140, 1998.
2. C. Roberts. Radio frequency identification (RFID). *Journal of Computers and Security*, 25:18–26, 2006.
3. L. Huan-Bang, K. Takizawa, Z. Bin, and R. Kohno. Body Area Network and Its Standardization at IEEE 802.15.MBAN. In *Mobile and Wireless Communications Summit*, pages 1–5, 2007.
4. N. Gasson, E. Kosta, and D. Bowman. Technical Challenges of Human ICT Implants. In *Human ICT Implants: Technical, Legal and Ethical Considerations*, pages 55–63, 2012.
5. P. Tuyls and L. Batina. RFID-tags for Anti-Counterfeiting. In *Topics in Cryptology*, pages 115–131. Springer Verlag, 2006.
6. Y. Lee, L. Batina, D. Singele, B. Preneel, and I. Verbauwhede. Anti-counterfeiting, Untraceability and Other Security Challenges for RFID. In *Towards Hardware-Intrinsic Security*, pages 237–257. Springer, 2010.
7. L. Batina, J. Guajardo, T. Kerins, N. Mentens, P. Tuyls, and I. Verbauwhede. Public-Key Cryptography for RFID-Tags. In *Pervasive Computing and Communications Workshops*, pages 217–222, 2007.
8. K. Yong, L. Batina, and I. Verbauwhede. EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID authentication protocol. In *RFID, 2008 IEEE International Conference on*, pages 97–104, 2008.
9. Yi-Pin Liao and Chih-Ming Hsiao. A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol. *Ad Hoc Networks*, 2013.
10. Xinglei Zhang, Jianhua Li, Yue Wu, and Quanhai Zhang. An ECDLP-Based Randomized Key RFID Authentication Protocol. In *Network Computing and Information Security (NCIS), 2011 International Conference on*, volume 2, pages 146–149, 2011.
11. N. Koblitz. Elliptic Curve Cryptosystems. *Journal of American Mathematical Society*, 48:203–209, 1987.
12. J. Aumasson, L. Henzen, W. Meier, J. Miret, and M. Plasencia. Quark: A Lightweight Hash. *Journal of Cryptography*, 26(2):313–339, 2013.
13. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Journal of Association for Computing Machinery*, 21(2):120–126, 1978.
14. E. Kavun and T. Yalcin. A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications. In *Radio Frequency Identification: Security and Privacy Issues*, volume 6370, pages 258–269, 2010.
15. S. Rahimi, A. Hakkala, J. Isoaho, S. Virtanen, and J. Isoaho. Specification Analysis for Secure RFID Implant Systems. *Journal of Computer Theory and Engineering*, 6(2):177–189, 2014.
16. S. Martinez, M. valls, C. Roing, J. Miret, and F. Gine. A Secure Elliptic Curve-Based RFID Protocol. *Journal of Computer Science and Technology*, 24(2):309–318, 2009.
17. G. Godor and S. Imre. Elliptic curve cryptography based authentication protocol for low-cost RFID tags. In *RFID-Technologies and Applications (RFID-TA), 2011 IEEE International Conference on*, pages 386–393, 2011.
18. J. Daemen and V. Rijmen. Specication of Rijndael. In *The Design of Rijndael*, volume 17, pages 31–50, 2002.
19. G. Godor, M. Antal, and S. Imre. Mutual Authentication Protocol for Low Computational Capacity RFID Systems. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pages 1–5, 2008.
20. S. Kumar and C. Paar. Are standards compliant elliptic curve cryptosystems feasible on rfid? In *In Proc. of RFIDSec06*, 2006.