J Supercomput (2018) 74:4281–4294
https://doi.org/10.1007/s11227-016-1861-1

A secure ECC-based RFID mutual authentication protocol for internet of things

Amjad Ali Alamr[1] · **Firdous Kausar**[1] · **Jongsung Kim**[2,3] · **Changho Seo**[4]

Published online: 8 September 2016

Abstract Progression of the internet technologies has led to the emergence of internet of things (IoT). One of the familiar deployment of IoT is through radio-frequency identification (RFID) technology. In recent times, RFID based systems are one of the most widely spread applications for tagging and keep tracking purposes in IoT deployment. This is due to their powerful features compared to their counterparts of similar techniques such as barcodes. In contrast, radio-frequency identification systems suffer from various attacks and security threats. The wireless channel used for communication is responsible for the majority of these vulnerabilities. In this paper, we propose a new radio-frequency identification authentication protocol based on elliptic curve cryptography (ECC) to eliminate these vulnerabilities. In addition, we use elliptic curve Diffie–Hellman (ECDH) key agreement protocol to generate a temporary shared key used to encrypt the later transmitted messages. Our protocol achieves a set of security properties likes mutual authentication, anonymity, confidentiality, forward

✉ Jongsung Kim
jskim@kookmin.ac.kr; jongsung.k@gmail.com

Amjad Ali Alamr
amjad.imamu@ccis.imamu.edu.sa

Firdous Kausar
firdous.kausar@ccis.imamu.edu.sa

Changho Seo
chseo@kongju.ac.kr

[1] Department of Computer Science, College of Computer and Information Science, Imam Mohammad bin Saud Islamic University, Riyadh, Saudi Arabia

[2] Department of Mathematics, Kookmin University, Seoul, Korea

[3] Department of Financial Information Security, Kookmin University, Seoul, Korea

[4] Department of Applied Mathematics, Kongju National University, Kongju, Korea

Table 1 Comparison of LF, HF, and UHF

	Frequency	Read range
LF	125 kHz	Greater than 30 cm
HF	13.56 MHz	Greater than 1 m
UHF	866 MHz 960 MHz	Greater than 7 m

security, location privacy, resistance of man-in-the-middle attack, resistance of replay attack and resistance of impersonation attack. We implement our proposed protocol in real RFID system using Omnikey smartcard reader (Omnikey 5421) and NXP Java smartcards (J3A040). Implementation results shows that our proposed protocol outperform in term of time complexity as compared to other similar protocols and requires less number of operations.

Keywords RFID · ECC · ECDH · Authentication protocol · Internet of things

1 Introduction

Radio-frequency identification (RFID) is an automatic identification technology that transmits data through the use of wireless communication using radio waves. The first use of RFID system was in the World War II for the friend or foe? identification system. Recently, there has been many applications that take advantage from RFID technology such as: point of sale (POS), automated vehicle identification (AVI) systems, asset tracking, pet ownership identification, product security, library books check-in/check-out, and e-passports. The principal advantage of this technology is that it automatically identifies objects using electromagnetic waves without requiring contact and line of sight. Which at the same time raise various vulnerabilities. In general, RFID system composed of three main parts: tags, readers, and server or backend database. Tags can be passive or active according to the power source. The active tag contains its built in power supply so it gets the power from itself. While the passive tag need to be charged by the electromagnetic field produced by the reader by the transmitted signal so it gets the power from external resources (reader). Also the tags can be low frequency (LF), high frequency (HF), or ultra high frequency (UHF) based on the used frequencies as shown in Table 1.

Recently, internet of things (IoT) is becoming as one of the most dominant communication model in the modern world. IoT allowed all the physical object in our daily life connect to internet and create an environment where these object can identify and communicate with each other through different communication methods including RFID, WIFI, QR codes or other sensor technologies [1]. Some of the prevalent applications of IoT include, but not limited to: home automation, smart city, wearable, industrial internet, connected car, smart grid, smart retail, and telehealth. Security is a big issue in case when with IoT, the physical world is becoming one big information system by connecting billions of devices together to make sure that their information stays secure. RFID systems use wireless channel for communication between reader and tags which is vulnerable to various security and privacy threats such as eavesdropping, cloning, traceability, and skimming. Therefore, the need to include security approaches to protect transmitted data is becoming urgent. RFID authentication protocols are one of these approaches used to authorize each party (reader and tags) in the communication process which connected to the IoT infrastructure.

Various authentication protocols have been proposed to achieve certain security and privacy goals. The limited resources of RFID system in term of storage capacity, computational capacity, and power restrict the use of strong and complex authentication protocols. Based on the RFID system resources, RFID authentication protocols can be classified into full-fledge, simple, lightweight, ultra-lightweight authentication protocols [2]. In the full-fledge class, the protocol requests the support of conventional cryptographic functions such as public key cryptographic (PKC) or one-way cryptographic function. In fact, PKC assures highest level of security and privacy protection, but it is not fully supported by RFID system because of its high capacity requirement in term of key size and computational cost. One of the most attractive PKC solution is elliptic curve cryptography (ECC) as it provides the same level of security with the smaller key sizes, faster computations, lower power consumptions as well as memory and bandwidth savings in contrast to the other PKC such as RSA. An elliptic curve is defined as a set of points (x,y) that satisfy an elliptic curve equation: $E : y^2 = x^3 + ax + b$, where x, y, a and b are within a field. For cryptographic purpose those over the finite field of Fp and F2m are most suitable. The strength of our proposed protocol is based on two elliptic curve computational problem which are: elliptic curve discrete logarithm problem (ECDLP) and elliptic curve factorization problem (ECFP). ECDLP is to find $k \in [1, n - 1]$ such that $Q = k.P$ where Q and P are two points over E. And the ECFP is to find the points $s.P$ and $t.P$ from $Q = s.P + t.P$ where $P, Q \in E$ and $s, t \in [1, n - 1]$.

2 Related work

Recently, RFID technology deployed in various applications, especially as an identity management system, such as supply chain management, e-passports, and credit card. [3]. These applications request different level of security based on their requirements and capacity which can be achieved by authentication protocols. RFID authentication protocols can be classified into three major classes based on used mechanisms, available resources, and cryptographic technique. Each of these classes can be classified into more subclasses [2]. Currently, series of full-fledge RFID authentication protocols have been proposed. In 2012, Benssalah et al. [4] proposed an efficient challenge-response protocol based on elliptic curve ElGamal encryption schemes. They minimize the computation amount on the tag side by a pseudorandom number generation (PRNG), an elliptic curve point addition, and two scalar multiplications. They mentioned that their protocol resist from the following security attacks: passive attacks, man-in-the-middle attacks, replay attacks. While Chou et al. [5] proposed a new RFID mutual authentication protocol based on ECC. This protocol possesses the properties of location privacy, forward secrecy, and mutual authentication. In addition, it can resist replay attack, man-in-the-middle attack, impersonation attack and physical attack. It can achieve a good performance in term of number of multiplication points and hash function. In 2013, Chou [6] adopt ECC to design an efficient RFID mutual authentication protocol operating under the constraint of a tags limited computational ability. His protocol possesses the following security properties: location privacy and mutual authentication. Also it can resist replay attacks, man-in-the-middle attacks, impersonation attacks. Farash in 2014 [7], analyze Chou protocol and found that it suffers from lack of tag privacy, lack of forward privacy, lack of mutual authentication weaknesses. Also, it is defenseless to impersonation attacks, tag cloning attacks and location tracking attacks. Then he proposes a more secure and efficient scheme to cover all the security flaws and weaknesses of Chous protocol. Moreover, by combining a secure ID-verifier transfer protocol and challenge-response protocol, Liao and Hsiao [8] introduce a new ECC-based RFID authentication scheme using hybrid protocols. Their scheme can satisfy the security requirements of RFID, such as mutual authentication, ID-verifier confidentiality, anonymity, availability, forward security and scalability. Also, it resists some attacks like replay attack, tag masquerade attack, server spoofing attack, DoS attack, location tracking attack and cloning attack. However, all of these protocols were not sufficient enough and thy still suffer from different issues. In our paper, we introduce a new ECC-based RFID authentication protocol to overcome these issues and improve their efficiency. We use ECDH as a key agreement protocol so establish a secure communication between tag and reader.

Hannes et al. [9] presents an IPSec conform mutual authentication protocol with added attribute of privacy awareness for IoT infrastructure based on the Diffie–Hellman Integrated Encryption (DHIES) scheme [10]. It has been shown that the tag does not reveal the sensitive information unless it has assured that communication is initiated by the genuine backend reader which achieve privacy preservation concern of RFID carriers.

Debiao et al. [11] presents an in-depth survey of ECC-based RFID authentication schemes and shows their suitability for the IoT based healthcare environment in term of security and performance requirements. The analysis shows that none of these currently available schemes is provably secure against different types of malicious attacks.

3 Essential RFID security requirement

Several security requirements for RFID systems were defined [7,8]. To enhance the security of our proposed protocol, we need to define the security requirements that must be considered in designing an RFID authentication protocol. The major requirements are mutual authentication, confidentiality, anonymity, availability, scalability, forward security, and location privacy. Also, we should specify potential attacks, such as man-in-the-middle attack (MIMA), replay attack, impersonation attack, brute force attack, denial-of-service (DoS) attack, and tracking attack [6–8]. As the wireless communication between tag and reader is the most vulnerable part of the RFID system, we consternate on the most related requirements and attacks, such as:

Mutual authentication Where each party in RFID system authenticate the other (tag authenticate the reader and vice versa).

Confidentiality Where all the secret information is securely exchanged during all communications. This required the encryption of information in a way that can be recognized only by authorized party.

Anonymity It is the most important security requirement for privacy. Where the attacker can learn the tags identifier that is used in the authentication process.

Forward security Where the previously transmitted data cannot be traced by the current tag transmission. That means, the attacker however exposes a tag and obtain its data, cannot trace the tag through previous conversations.

Location privacy Where the attacker cannot track or monitor the tag by keeping the user location private as well as tags identifier.

Man-in-the-middle attack (MIMA) Where the attacker interrupts the communication between tag and reader and redirects or may modify the exchange messages without knowledge of them.

Replay attack It is the ability of the attacker to eavesdrop and capture the conversation between the tag and the reader and replay the same message previously sent to pass the verification of the system.

Impersonation attack It is the ability of the attacker to successfully impersonate a tag (reader) to authenticate himself to the reader (tag) while he does know the tags (readers) secret key.

4 The proposed ECC based authentication protocol

This paper proposes a new ECC-based mutual authentication protocol that fulfill the RFID security requirements. Also, it uses elliptic curve Diffie–Hellman (ECDH) key agreement protocol to establish a secure communication between tag and reader. It allows each parties having its elliptic curve public-private key pair then use it to authenticate each other and derive a new changeable key which can be used to encrypt communication. The proposed protocol achieves most of the RFID security requirements and resists various attacks. The notations being used in rest of paper are described in Table 2.

Table 2 Notations

Notation	Description
GF(p)	Galois field
n	Elliptic curve order
P	Elliptic curve base point
a, b	Co-factors of elliptic curve equation “part of the ECC common parameters”
Pr_R	Reader private key
Pu_R	Reader public key
Pr_T	Tag private key
Pu_T	Tag public key

Table 3 System parameters

System parameters	Pu_R, P (EC base point), n (EC order)
Reader storage	Pr_R, Pu_R and common parameters (P, Pu_T, n)
Tag storage	Pr_T, Pu_T and common parameters (P, n, Pu_R)

Our protocol is based on ECC and derived its strength from ECDLP and ECFP. It consists of two phases: initialization phase, and authentication phase described below.

4.1 Initialization phase

In this phase, the server generates system parameters. It chooses a random number $Pr_R \in F_p$ as a reader private key and sets $Pu_R = Pr_R.P$ as its public key. Also choses $Pr_T \in F_p$ as the tag private key and sets $Pu_T = Pr_T.P$ as the tags public key. Then each tag and reader store its key pair with the system parameters in the memory. Table 3 summarize the system parameters and the storage of each party.

4.2 Authentication phase

The authentication phase of our protocol is illustrated in Fig. 1. Here, we describe the interaction between tag and reader as follow:
Step 1: The reader generates a random number $r_1 \in F_p$ and computes

$$R_1 = r_1.P \tag{1}$$

Then the reader sends R_1 to the tag.
Step 2: After the tag receives the R_1, it generates random number $t_1 \in F_p$ and computes

$$T_1 = t_1.P \tag{2}$$

Then the tag calculates two secret keys

$$SK1_T = Pr_T.R_1 \tag{3}$$

$$SK2_T = t_1.R_1 \tag{4}$$

Finally, the tag computes

$$C_1 = SK1_T + SK2_T \tag{5}$$

to encrypt the tag secret keys and sends T_1 and C_1 to the reader.

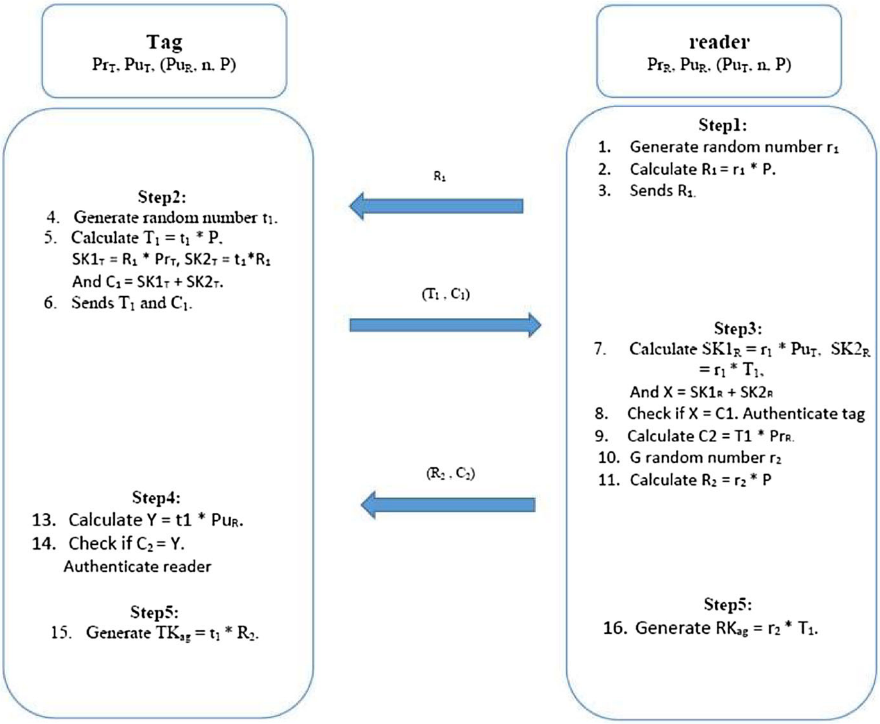

Tag
PrT, PuT, (PuR, n, P)
reader
PrR, PuR, (PuT, n, P)
Step1:
1. Generate random number r1
2. Calculate R1 = r1 * P.
3. Sends R1.
R1
Step2:
4. Generate random number t1.
5. Calculate T1 = t1 * P.
SK1T = R1 * PrT, SK2T = t1*R1
And C1 = SK1T + SK2T.
6. Sends T1 and C1.
(T1 , C1)
Step3:
7. Calculate SK1R = r1 * PuT, SK2R = r1 * T1,
And X = SK1R + SK2R
8. Check if X = C1. Authenticate tag
9. Calculate C2 = T1 * PrR.
10. G random number r2
11. Calculate R2 = r2 * P
(R2 , C2)
Step4:
13. Calculate Y = t1 * PuR.
14. Check if C2 = Y.
Authenticate reader
Step5:
15. Generate TKag = t1 * R2.
Step5:
16. Generate RKag = r2 * T1.

Fig. 1 The proposed authentication protocol

Step 3: After receiving (T_1, C_1), the reader calculate two temporary secret keys

$$SK1_R = r_1.Pu_T \tag{6}$$

$$SK2_R = r_1.T_1 \tag{7}$$

to recover the tag encrypted secret keys. Then calculates

$$X = SK1_R + SK2_R \tag{8}$$

and compare X to C_1 if $X = C_1$ the reader authenticates the tag to be genuine. Then it calculates

$$C_2 = T_1.Pr_R \tag{9}$$

Moreover, generates new random number $r_2 \in F_p$ and computes

$$R_2 = r_2.P \tag{10}$$

to be use it for key agreement. Finally, the reader sends C_2 and R_2 to the tag.
Step 4: The tag compute

$$Y = t_1.Pu_R \tag{11}$$

then compare it to C_2 if $Y = C_2$ the tag authenticates the reader as a genuine.
Step 5: Both parties set the key agreements between them. The tag key agreement

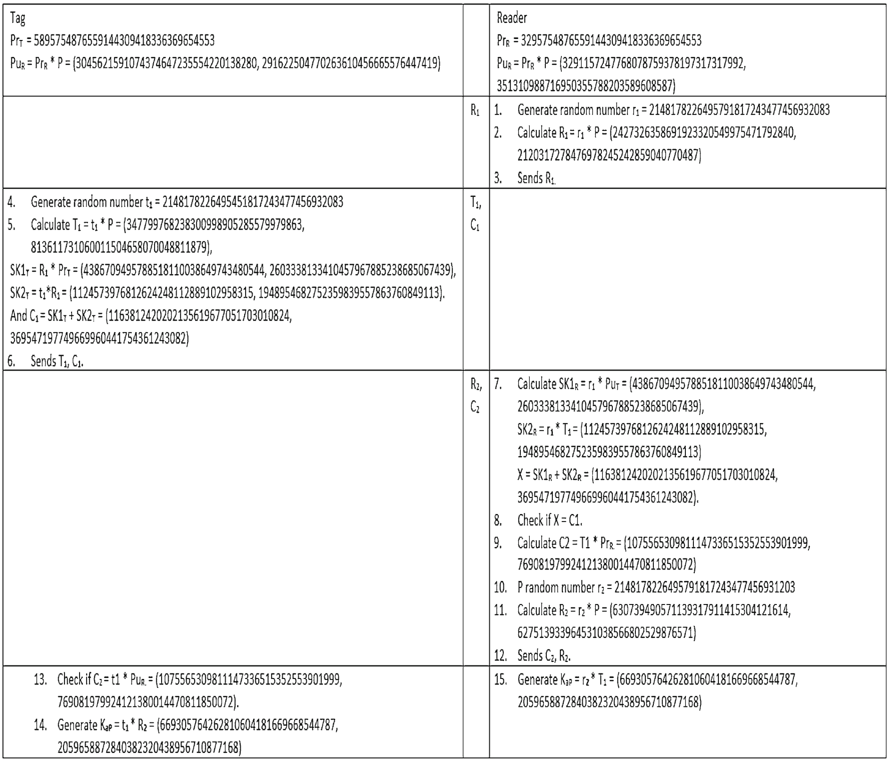
$$TK_{\text{ag}} = t_1.R_2 \tag{12}$$

and the reader key agreement

$$RK_{\text{ag}} = r_2.T_1 \tag{13}$$

4.3 Protocol exemplify

For more clarification of our proposed protocol, we take an example to prove the correctness of our protocol as shown in Fig. 2. We use SECP112R1 as a curves domain parameters. The parameters of this curve is as bellow:



Tag Pr_T = 5895754876559144309418336369654553 $Pu_R = Pr_R$ * P = (3045621591074374647235554220138280, 2916225047702636104566655764447419)		Reader Pr_R = 3295754876559144309418336369654553 $Pu_R = Pr_R$ * P = (3291157247768078759378197317317992, 3513109887169503557882035896085 87)
	R_1	1. Generate random number r_1 = 2148178226495791817243477456932083 2. Calculate $R_1 = r_1$ * P = (2427326358691923320549975471792840, 2120317278476978245242859040770487) 3. Sends R_1.
4. Generate random number t_1 = 2148178226495451817243477456932083 5. Calculate $T_1 = t_1$ * P = (3477997682383009989052855799798 63, 8136117310600115046580700488118 79), $SK1_T = R_1$ * Pr_T = (4386709495788518110038649743480544, 2603338133410457967885238685067439), $SK2_T = t_1$*R_1 = (1124573976812624248112889102958315, 1948954682752359839557863760849113). And $C_1 = SK1_T + SK2_T$ = (1163812420202135619677051703010824, 3695471977496699604417543612430 82) 6. Sends T_1, C_1.	T_1, C_1	
	R_2, C_2	7. Calculate $SK1_R = r_1$ * Pu_T = (4386709495788518110038649743480544, 2603338133410457967885238685067439), $SK2_R = r_1$ * T_1 = (1124573976812624248112889102958315, 1948954682752359839557863760849113) X = $SK1_R + SK2_R$ = (1163812420202135619677051703010824, 3695471977496699604417543612430 82). 8. Check if X = C1. 9. Calculate C2 = T1 * Pr_R. = (1075565309811147336515352553901999, 7690819799241213800144708118500 72) 10. P random number r_2 = 2148178226495791817243477456931203 11. Calculate $R_2 = r_2$ * P = (6307394905711393179114153041216 14, 6275139339645310385668025298765 71) 12. Sends C_2, R_2.
13. Check if C_2 = t1 * Pu_R. = (1075565309811147336515352553901999, 7690819799241213800144708118500 72). 14. Generate $K_{aP} = t_1$ * R_2 = (6693057642628106041816696685447 87, 2059658872840382320438956710877168)		15. Generate $K_{aP} = r_2$ * T_1 = (6693057642628106041816696685447 87, 2059658872840382320438956710877168)

Fig. 2 Protocol example

Field type: prime-field

Prime 4451685225093714772084598273548427.
A 4451685225093714772084598273548424
B 2061118396808653202902996166388514
Order 4451685225093714776491891542548933.
Seed 5464641678502306533941025049572469019726331825.
Cofactor 1

To calculate the operation in our protocol (point addition and scalar multiplication) we use the built in elliptic curve calculator tool [12].

5 Security analysis

In this section, we analyze the proposed protocol and prove its correctness and strength in terms of five major RFID security requirements (mutual authentication, confidentiality, anonymity, forward security and location privacy). Also, it resists from three main attacks (MIMA, replay attack, and impersonation attack). First of all, we make some reasonable assumption to support the security analysis.

A1: all the random numbers are fresh in every session.
A2: the tag private key is unknown to anyone except the tag itself.
A3: the reader private key is unknown to anyone except the reader itself. Also, we set some inferences to guide us in the analysis:
I1: the tags private key is embedded in C1 and securely transmitted to the reader. In step2, the tag sends C1 to the reader if the attacker can get C1 he cannot extract the private key of the tag from it based on the ECFP. Also, the generated temporary secrets key cannot be predicted because they base on ECDLP.
I2: the readers private key is embedded in C2 and securely transmitted to the tag. In step3, C2 = PrR . T1 the attacker cannot extract the readers private key based on the ECDLP.
I3: According to A1, all the generated random numbers are variant in every session so the freshness of the exchange messages are assured

Therefore, the attacker cannot reuse the previous messages to impersonate the tag or the reader or to track the tag. We analyze our protocol for the following properties

1. *Mutual authentication* In our protocol, the reader can authenticate the tag by the ability to calculate the correct value of X which must be equal to C_1. According to I1 and A2, only the genuine reader can calculate the correct value without knowledge of the tag private key. In other hand, the tag can authenticate the reader by the ability of calculating the same value of $C_2(Y)$. From I2 and A3, only the genuine tag can calculate the correct value of Y without knowledge of reader private key. Hence, we prove that both parties authenticate each other.
2. *Confidentiality* According to I1 and I2, the attacker cannot extract private key of any party from the exchange messages (C_1, C_2).
3. *Anonymity* From the confidentiality property the rags identifier (private key) cannot be extracted. Moreover, because of the freshness of random numbers the exchange

Table 4 Security comparison

	Benssalah et al. [4]	Liao et al. [8]	Farash [7]	Proposed protocol
Mutual authentication	No	Yes	Yes	Yes
Confidentiality	–	Yes	–	Yes
Anonymity	–	Yes	–	Yes
Forwards security	–	Yes	Yes	Yes
Location privacy	–	Yes	Yes	Yes
Resistance of MIMA	Yes	–	Yes	Yes
Resistance of replay attack	Yes	Yes	Yes	Yes
Resistance of impersonation attack	–	Yes	Yes	Yes

messages will by varies for each session which prevent attacker from predicting tag identifier.

4. *Forward security* By assuming that an attacker knows the tag key pairs (Pu_T, Pr_T) by physical attacks he still cannot know the fresh random number temporary generated and used by its party. So the attacker cannot predict the previous exchanged messages and use it later.
5. *Location privacy* According to confidentiality property and I3, the exchange messages between the tag and reader is well protected and provided on unpredictable variation in every session. This making it difficult for the attacker to track the tag.
6. *Resistance of MIMA* From the I1 and I2, the value of exchange messages $(C_1 or C_2)$ cannot be calculated correctly unless by the genuine parties. So if an attacker intercepts the communication channel between tag and reader he cannot extract any secret or useful data that initiate an attack. For example, if an attack intercepts the exchange message (C_1) from I1 he cannot extract the private key so he cannot reuse it to send it to the reader. And if he used uncorrected private key the reader cannot calculate the correct value of X.
7. *Resistance of replay attack* If the attacker tries to intercept the previous communication and replay the same message to pass the verification process. According to I3, because of the freshness of the transmitted messages this attacker will be fail to reuse the previous exchange messages $(C_1 or C_2)$ to masquerade as the reader or tag.
8. *Resistance of impersonation attack* From I1 and A2, if an attacker tries to impersonate a tag to a reader he will fail because he must use the tags private key to compute C_1. On the other hand, from I2 and A3, the attacker fails to impersonates a reader to a tag because he need to use readers private key to calculate C_2.

Table 4 conclude the security comparisons of the related ECC-based RFID authentication protocols with our proposed protocol.

6 Performance evaluation

To evaluate the performance and functionality of our protocol in term of time and memory space, we choose to implement it in real RFID system. For hardware, we

Table 5 Performance comparison

	TC					SC		
	T_{GKey} (ms)	T_{RSet} (ms)	T_{AutC} (ms)	T_{AutR} (ms)	T_{KAgr}	SC_{PRS} (Byte)	SC_{TRN} (Byte)	SC_{TRNRST} (Byte)
Liao et al. [8]	845	532	1003	2156	No	294	165	189
Our protocol	845	533	891	297	256 ms	294	165	189

use a laptop (ASUS 46 bit windows with Operating system windows 8.1), Omnikey smartcard reader (Omnikey 5421) [13], and smartcards ($J3A040$) [14]. Our smart cards are Jcop J3A040 version 2.4.1 with dual interface, $T = 1$, 40 KB EEPROM. These are an NXP [15] implementation cards with support for PKC (both ECC and RSA).

For software, we use eclipse IDE for Java Developer (Mars.2 Release (4.5.2)), Java Runtime Environment (jre7), Java Development Kit (jdk 1.7.0_79), and Java Card Kit (java_card_kit_2_2_2) for building smart card applets. And GPShell (GPShell-1.4.4) for writing script that communicate with the reader. See Appendix for more description of software installation and our applets. In our ECC implementation, we use $secp192r1$ [16]. It is specified by the six tuple $T = (p, a, b, G, n, h)$ where:

$P = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF$
$FFFFFFFF$
$a = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFFFF$
$FFFFFFFC$
$b = 64210519E59C80E70FA7E9AB72243049FEB8DEECC146B9B1$
$G = 04188DA80EB03090F67CBF20EB43A18800F4FF0AFD82FF101207$
$192B95FFC8DA78631011ED6B24CDD573F977A11E794811$
$n = FFFFFFFFFFFFFFFFFFFFFFFF99DEF836146BC9B1B4D22$
831
$h = 01$

We compare the performance of our proposed protocol with one of the similar ECC-based RFID authentication protocol [8] in term of time and memory space requirement.

As it is known, the tag's computing capability and memory are restricted which make the computation cost and storage requirements as most important characteristics for practical applications. Therefore, we constraint in our comparison in tag side only. We measure the time and storage cost for the tag only. The storage cost is denoted as (SC) and time cost as (TC). For more clarification, we use $T_{\text{GKey}}, T_{\text{RSet}}, T_{\text{AutC}}, T_{\text{AutR}}, T_{\text{End}}$ for key pair generation, random points setting, authenticate card step, authenticate reader step and end operation, respectively. Also, we use $SC_{\text{PRS}}, SC_{\text{TRN}}, SC_{\text{TRNRST}}$ for memory type persistent, transient, and transient with reset, respectively.

Table 5 summarizes the performance comparison of our proposed protocols with [8] by computing time and storage cost of each of the above mentioned measures. From the storage cost point of view, we found that they are the same and no difference which in fact means that our protocol is better because it has more extra step for

Table 6 Number of operations

	Benssalah et al. [4]		Liao et al. [8]		Farash [7]		Proposed protocol	
	Tag	Reader	Tag	Reader	Tag	Reader	Tag	Reader
Random number generator	1	1	1	1	1	1	1	2
Scalar multiplication	4	1	5	5	2	2	4	5
Point addition	1	1	2	2	0	0	1	1

key agreement based on ECDH. In contrast, our protocol outperforms the Liao et al. [8] protocol in term of time cost. The total time cost of our protocol equal 2822 ms, whereas the total time cost of Liao et al. protocol [8] is 4536 ms. It shows that our proposed protocol reduces around half of the time cost for Liao et al. protocol [8].

Table 6 shows the comparison of our proposed protocol with other related protocol [4,7,8] in term of number of operations required in each protocol. It has been found that our protocol performed better to Liao et al. [8] because it has one less point addition operation in both parties and one less scalar multiplication operation in the tag side.

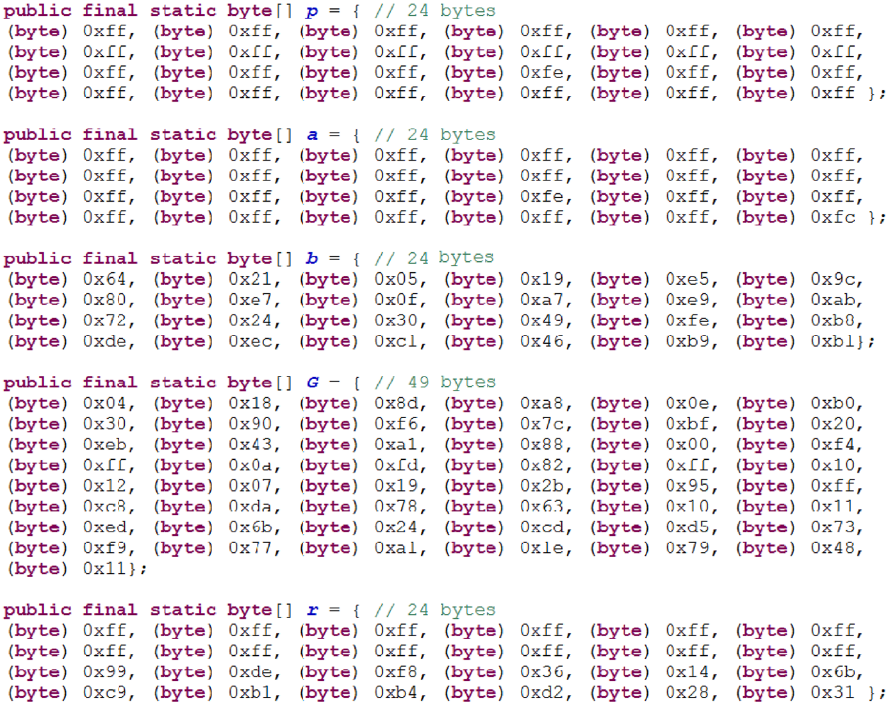
Further, our protocol is proposed for applications that does not depend on database. It stores the sensitive data information on its' corresponding tag memory and need to authenticate the reader before allowing access to these sensitive data. In addition, the reader also need to authenticate the tag to avoid cloned tag. These authentication is done without referring to the backend database. After each party authenticate each other, the ECDH key agreement protocol is added to encrypt the data transmitted later as the data required is stored in the tag memory.

7 Conclusion

Limited resources of RFID systems making the introducing of a strong and efficient security system very challenging process. In our paper, we propose a secure ECC-based authentication protocol to eliminate the current RFID vulnerabilities raised be insecure communication channel between tag and reader. The strength of our protocol is based on the two main ECC computational problem: ECDLP and ECFP. We used ECDH as a key agreement protocol to agree on a shared used to encrypt the later exchanged messages to protect the tag data. Our security analysis show that the proposed protocol will fulfill the requirements of mutual authentication, confidentiality, anonymity, forward security and location privacy. Also, our protocol resist from the following attacks MIMA, replay attack and impersonation attack. Performance evaluation shows that our proposed protocol is more efficient and requires much less time as compared to others [8].

Acknowledgements This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government (MEST) (No. 2016$R1A4A$1011761).

8 Appendix



```
public final static byte[] p = { // 24 bytes
(byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff,
(byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff,
(byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xfe, (byte) 0xff, (byte) 0xff,
(byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff };

public final static byte[] a = { // 24 bytes
(byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff,
(byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff,
(byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xfe, (byte) 0xff, (byte) 0xff,
(byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xfc };

public final static byte[] b = { // 24 bytes
(byte) 0x64, (byte) 0x21, (byte) 0x05, (byte) 0x19, (byte) 0xe5, (byte) 0x9c,
(byte) 0x80, (byte) 0xe7, (byte) 0x0f, (byte) 0xa7, (byte) 0xe9, (byte) 0xab,
(byte) 0x72, (byte) 0x24, (byte) 0x30, (byte) 0x49, (byte) 0xfe, (byte) 0xb8,
(byte) 0xde, (byte) 0xec, (byte) 0xc1, (byte) 0x46, (byte) 0xb9, (byte) 0xb1};

public final static byte[] G = { // 49 bytes
(byte) 0x04, (byte) 0x18, (byte) 0x8d, (byte) 0xa8, (byte) 0x0e, (byte) 0xb0,
(byte) 0x30, (byte) 0x90, (byte) 0xf6, (byte) 0x7c, (byte) 0xbf, (byte) 0x20,
(byte) 0xeb, (byte) 0x43, (byte) 0xa1, (byte) 0x88, (byte) 0x00, (byte) 0xf4,
(byte) 0xff, (byte) 0x0a, (byte) 0xfd, (byte) 0x82, (byte) 0xff, (byte) 0x10,
(byte) 0x12, (byte) 0x07, (byte) 0x19, (byte) 0x2b, (byte) 0x95, (byte) 0xff,
(byte) 0xc8, (byte) 0xda, (byte) 0x78, (byte) 0x63, (byte) 0x10, (byte) 0x11,
(byte) 0xed, (byte) 0x6b, (byte) 0x24, (byte) 0xcd, (byte) 0xd5, (byte) 0x73,
(byte) 0xf9, (byte) 0x77, (byte) 0xa1, (byte) 0x1e, (byte) 0x79, (byte) 0x48,
(byte) 0x11};

public final static byte[] r = { // 24 bytes
(byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff,
(byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff, (byte) 0xff,
(byte) 0x99, (byte) 0xde, (byte) 0xf8, (byte) 0x36, (byte) 0x14, (byte) 0x6b,
(byte) 0xc9, (byte) 0xb1, (byte) 0xb4, (byte) 0xd2, (byte) 0x28, (byte) 0x31 };
```

References

1. Atzori Luigi, Iera Antonio, Morabito Giacomo (2010) The internet of things: a survey. Comput Netw 54(15):2787–2805
2. Chien Hung-Yu (2009) Development and implementation of RFID Technology, chapter the study of RFID authentication protocols and security of some popular RFID tags, page 554. i-tech, Vienna
3. Hof C (2006) Rfid and identity management in everyday life: striking the balance between convenience, choice and control. Report IPOL/A/STOA/2006-22, ETAG (European Technology Assessment Group), European Parliament, Strasbourg
4. Benssalah M, Djeddou M, Drouiche K (2012) RFID authentication protocols based on ECC encryption schemes. In: 2012 IEEE international conference on RFID-technologies and applications, RFID-TA 2012, Nice, France, November 5–7, 2012, pages 97–100
5. Chou J-S, Chen Y, Wu C-L, Lin C-F (2011) An efficient rfid mutual authentication scheme based on ecc. Cryptology ePrint Archive, Report 2011/418
6. Chou Jue-Sam (2014) An efficient mutual authentication RFID scheme based on elliptic curve cryptography. J Supercomput 70(1):75–94
7. Farash Mohammad Sabzinejad, Kumari Saru, Bakhtiari Majid (2016) Cryptanalysis and improvement of a robust smart card secured authentication scheme on SIP using elliptic curve cryptography. Multimed Tools Appl 75(8):4485–4504
8. Liao Y-P, Hsiao C-M (2012) A secure ECC-based RFID authentication scheme using hybrid protocols. In: Proceedings of the International Computer Symposium ICS 2012-Advances in Intelligent Systems and Applications, vol 2. Springer, pp 1–13
9. Gross H, Hlbl M, Slamanig D, Spreitzer R (2015) Privacy-aware authentication in the internet of things. Cryptology ePrint Archive, Report 2015/1110

10. Abdalla M, Bellare M, Rogaway P (2001) The oracle diffie-hellman assumptions and an analysis of dhies. In: Naccache D (ed) CT-RSA, volume 2020 of lecture notes in computer science. Springer, Berlin, pp 143–158
11. He Debiao, Zeadally Sherali (2015) An analysis of RFID authentication schemes for internet of things in healthcare environment using elliptic curve cryptography. IEEE Int Things J 2(1):72–83
12. Christelbach. http://www.christelbach.com/eccalculator.aspx. Accessed 10 May 2014
13. OMNIKEY 5421 Reader. https://www.hidglobal.com/products/readers/omnikey/5421. Accessed 15 Sept 2014
14. 40k EEPROM J3A040 NXP JAVA based smart card. http://www.smartcardsource.com/contents/en-ca/p94_J3A040.html. Accessed 4 Apr 2013
15. Smart solutions for smart services : NXP. http://www.nxp.com/documents/line_card/75016728.pdf. Accessed 25 Dec 2013
16. Certicom Research. Standards for efficient cryptography sec 2: recommended elliptic curve domain parameters. http://www.secg.org/SEC2-Ver-1.0.pdf. Accessed 17 Mar 2014