Department of Computer Science and Engineering (CSE)
Islamic University of Technology (IUT)

ECC Based RFID Authentication in WSN/IoT

Authors
Salman Sayeed Khan - 124402
and
Sheik Benazir Ahmed - 124410

Supervisor
Prof. Dr. Muhammad Mahbub Alam
Professor
Department of CSE
Islamic University of Technology

A thesis submitted to the Department of CSE
in partial fulfillment of the requiremnts for the degree of B.Sc.
Engineering in CSE
Academic Year: 2015-16

November - 2016

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Salman Sayeed Khan and Sheik Benazir Ahmed under the supervision of Prof. Dr. Muhammad Mahbub Alam, Professor, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Salman Sayeed Khan
Student ID - 124402

Sheik Benazir Ahmed
Student ID - 124410

Supervisor:

Prof. Dr. Muhammad Mahbub Alam
Professor
Department of CSE
Islamic University of Technology

Acknowledgement

We would like to express our grateful appreciation for **Prof. Dr.Muhammad Mahbub Alam**, Professor, Department of Computer Science & Engineering, IUT for being our advisor and mentor. His motivation, suggestions and insights for this thesis have been invaluable. Without his support and proper guidance this research would not have been possible. His valuable opinion, time and input provided throughout the thesis work, from first phase of thesis topics introduction, subject selection, proposing algorithm, modification till the project implementation and finalization which helped us to do our thesis work in proper way. We are really grateful to him.

Abstract

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. The tags contain electronically stored information. Unlike a barcode, the tag need not be within the line of sight of the reader, so it may be embedded in the tracked object. RFID is one method for Automatic Identification and Data Capture (AIDC). RFID tags are used in many industries, for example, an RFID tag attached to an automobile during production can be used to track its progress through the assembly line; RFID-tagged pharmaceuticals can be tracked through warehouses; and implanting RFID microchips in livestock and pets allows positive identification of animals. In this we paper we review some of the RFID authentication protocols and compare their strengths/weaknesses and propose an authentication protocol that we have thought of provide a comparative analysis of our protocol.

Contents

0.1 Overview

Radio Frequency Identification (RFID) is a wireless AIDC technology that uses radio signals to identify a product, animal or person. The three main components of an RFID system are RFID tags, RFID readers and a back-end server. A tag is an identification device attached to an item, which uses radio frequency (RF) to communicate. The reader is a device that can recognize the presence of RFID tags and read the information supplied by them. The reader queries tags by broadcasting an RF signal, and the tag responds to the reader with a number or other identifying information. The reader forwards the tag response to a back-end server. The server has a database of tags and can retrieve detailed information regarding the tag (or the item attached to the tag) from the tag response. The main benefits of RFID systems are that they can provide automated and multiple identification capture and system analysis,can read several tags in the field at the same time automatically, and can help to track valuable objects. However, they can threaten the privacy of the owner carrying the tag as a result of automatic identification [3]; more specifically, tag information could be disclosed to unauthorized readers, and multiple readers could cooperate to track the movements of a tag. In addition, many possible security threats arise from the use of wireless communications. Moreover, it is infeasible to use computationally intensive cryptographic algorithms for privacy and security, because memory and processing power in a low cost tag are limited . Therefore, authentication protocols for RFID systems should not only be designed to address these privacy and security threats, but should also take into account the limited capabilities of RFID tags.

Authentication is a process in which the credentials provided are compared to those on file in a database of authorized users' information on a local operating system or within an authentication server. If the credentials match, the process is completed and the user is granted authorization for access. The permissions and folders returned define both the environment the user sees and the way he can interact with it, including hours of access and other rights such as the amount of allocated storage space. User authentication occurs within most human-to-computer interactions other than guest accounts, automatically logged-in accounts and kiosk computer systems. Generally, a user has to enter or choose an ID and provide their password to begin using a system. User authentication authorizes human-to-machine interactions in operating systems and applications as well as both wired and wireless networks to enable access to networked and Internet-connected systems, applications and resources. Machines need to authorize their automated actions within a network too. Online backup services, patching and updating systems and remote monitoring systems such as those used in telemedicine and smart grid technologies all need to securely authenticate to verify that it is the authorized system involved in any interaction and not a hacker.

Machine authentication can be carried out with machine credentials much like a users' ID and password only submitted by the device in question. They can also use digital certificates issued and verified by a Certificate Authority (CA) as part of a public key infrastructure to prove identification while exchanging information over the Internet, like a type of digital password.

The RFID technology has been one of the hottest issues in the wireless communication area. One of the reasons many developers are researching this topic is that the RFID

is supposed to replace the bar code systems. However, the application area is not restricted to product supply chains but covers livestock tracking, airline baggage, road toll management, hotel room access and so on. In order to be popular in commercial markets, the RFID system should overcome the restriction of cheap RFID tags. The limited price means limited functionalities and resources in tags. Because of the limitation, using asymmetric or symmetric key encryption algorithm or making memory secure in tags is improper. To solve security problems related with low-cost RFID systems, many authentication protocols were proposed. However, those protocols could not satisfy the RFID security requirements and operational requirements.

In this paper, we provide an authentication protocol based on Elliptic Curve Cryptography. We implemented the protocol in telosb using TinyOS NesC. We also implemented another existing protocol and performed a comparison between our and the existing protocol which show that our protocol provides faster communication.

0.2 Problem Statement

RFID tags are lightweight and resource constraint. These components have limited amount of battery power, memory. Security issue comes in case of communication with RFID tags. Different authentication protocols have been proposed to secure the data transmission. But due to privacy security issues, all standard encryption decryption algorithms authentication protocols cannot be used. We are proposing an authentication protocol in this paper.

0.3 Motivation & Scopes

If we classify cryptographic algorithms, we will find 2 major classifications.

- Symmetric Crypto System
- Asymmetric Crypto System

A secret key algorithm (sometimes called a symmetric algorithm) is a cryptographic algorithm that uses the same key to encrypt and decrypt data. The keys represent a shared secret between two or more parties that can be used to maintain a private information link. The main drawback of Symmetric Crypto System is that both parties have access to the secret key. Symmetric-key encryption can use either stream ciphers or block ciphers. Some important block ciphers are AES, Blowfish DES (Internal Mechanics, Triple DES) etc. Widely used stream ciphers are RC4, Block ciphers in stream mode, ChaCha etc.

Public key cryptography, or asymmetric cryptography, is any cryptographic system that uses pairs of keys: public keys which may be disseminated widely, and private keys which are known only to the owner. This accomplishes two functions: authentication, which is when the public key is used to verify that a holder of the paired private key sent the message, and encryption, whereby only the holder of the paired private key can decrypt the message encrypted with the public key. Some well regarded Asymmetric Crypto Systems are Diffie–Hellman key exchange protocol, DSS (Digital Signature Standard), Elliptic Curve Crypto System, RSA encryption algorithm etc.

Recently ECC seems to be a rather accepted approach towards authentication/privacy schemes. Elliptic curve cryptography (ECC) is an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. ECC requires smaller keys compared to non-ECC cryptography (based on plain Galois fields) to provide equivalent security. Elliptic curves are applicable for encryption, digital signatures, pseudo-random generators and other tasks.

Some of the major motivation and scopes are enlisted below:

- Resource constraint components have limited amount of battery power, memory
- Secure data transmission between lightweight devices

0.4 Thesis Outline

In Chapter 0 we have discussed our study in a precise and concise manner. Chapter 1 provides a brief introduction to Elliptic Curves. Chapter 3 deals with the necessary literature review for our study and their development so far. In Chapter 3 we introduce our proposal. We conclude our discussion in Chapter 4. The last page of our report/book contains all the references and credits used.

1 Elliptic Curves: Brief Intro.

The elliptic curve over Z_p, is the set of all pairs $(x, y) \in Z_p$ which fulfill

$$\mathrm{y}^2 \equiv x^3 + ax + b \quad mod \quad p$$

Together with an imaginary point of infinity, where
a,b $\in Z_p$

And the condition

$$4\mathrm{a}^3 + 27b^2 \neq 0 \quad mod \quad p$$

1.1 Point Addition and Point Doubling in EC

If we have two points $P(x_1, y_1)$ and $Q(x_2, y_2)$ then point addition/doubling will produce a new point $T(x_3, y_3)$ where

$$\mathrm{x}_3 = s^2 - x_1 - x_2 \quad mod \quad p$$
$$y_3 = s(x_1 - x_3) - y_1 \quad mod \quad p$$

and

$$s = \frac{y_2 - y_1}{x_2 - x_1} \quad mod \quad p; \quad if\ P \neq Q\ (point\ addition)$$

$$s = \frac{3{x_1}^2 + a}{2y_1} \quad mod \quad p; \quad if\ P = Q\ (point\ doubling)$$

1.2 Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given is an elliptic curve E. We consider a primitive element P and another element T. The DL problem is finding the integer d, where $1 \leq d \leq \sharp E$ where $\sharp E$ is the order, such that:

P+P+...+P (d times)=dP=T

Here P = Generator, d = Private Key, T = Public Key. To calculate d it takes $\sqrt{2^p}$ steps where p is the prime number (very large).

1.3 Elliptic Curve Diffie–Hellman Key Exchange (ECDH))

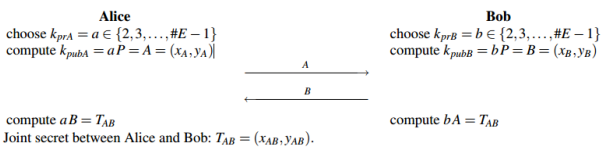

Figure 1: Elliptic Curve Diffie-Hellman

Alice computes $aB = a(bP)$ while Bob computes $bA = b(aP)$.

2 Recent RFID Studies and Authentication Protocols

This section reviews recent full-fledged RFID systems, and especially focuses on ECC-based solutions. As to the ECC-based RFID systems, they are typically constructed on an additive algebraic group G over an elliptic curve, which has generator P and order n. Their security basis is the ECDLP (elliptic curve discrete logarithm problem). That is, finding an integer c Zn satisfying Y = cX is considered computationally infeasible when n is sufficiently large, where X and Y are elements of G. In the following, we talk about several recent ECC-based RFID authentication proposals.

2.1 Tuyls et al.'s scheme

Tuyls et al.[**?**] in 2006 first proposed an ECC-based RFID identification protocol using the Schnorr identification scheme, as shown in Fig **??**.

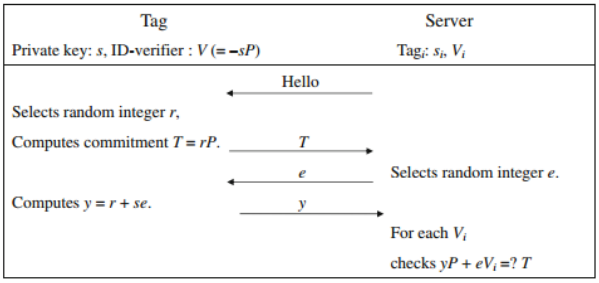

Figure 2: Tuyls et al.'s scheme

In general, a Schnorr identification process is completed through a three-round protocol, in which a prover first makes a commitment, a verifier then issues a random challenge, and finally the prover answers a corresponding response. In Tuyls et al.'s proposed system, the tag is a prover which keeps two secrets, a private key s (is an integer) and an ID-verifier V (is an elliptic curve point, V = sP), and the server is a verifier which stores the two secrets of all tags in the server database. When identifying tags, the server broadcasts a hello message, and starts Schnorr identification processes with each tag. On the server side, it should make each tag's response match a tag record in its database.

Lee et al. [**?**] pointed Tuyls et al.'s protocol suffers a privacy problem. When an adversary eavesdrops tag and server's communications and obtains a transcript, $\{T, e, y\}$, he could use e^{-1} to obtain the ID-verifier,$V(= -sP)$, by computing $(T - yP)e^{-1}$. Then, the adversary can use the computed V to track the tag.

There is another way to recognize a tag of Tuyls et al.'s scheme. An adversary also eavesdrops the communication of a specific tag and obtains three values, $T_1(= r_1P)$, e, and $y_1(= se + r_1)$. He then interrogates an unknown tag and receives the tag's commitment, $T_2(= r_2P)$. The adversary then replays challenge $e'(= e)$ to the unknown tag and obtains $y_2 = se' + r_2$. As a result, he could identify the unknown tag as the specific tag if $(y_2 - y_1)P$ equals to $T_2 - T_1$.

Thirdly, Tuyls et al.'s protocol lacks forward privacy. This is because when an adversary performs above-mentioned steps and obtains ID-verifier $V(= -sP)$ of a specific tag, he

can use this V to determine whether a past conversation, T^*, e^*, y^*, belongs to the specific tag by evaluating the equation $(T^* - y^*P)e^{-1} =?V$.

Finally, a scalability problem exists in Tuyls et al.'s scheme, because the server must fetch a candidate ID-verifier, V_i, from each record in the server database to compare if the value of $yP + eV_i$ equals to the received T . This also implies the server requires a brute search to identify a tag. The search time will be longer when the number of tags gets larger.

2.2 Batina et al.'s scheme

Batina et al. [**?**] proposed a similar ECC-based solution by applying Okamoto's identification. It still has privacy leakage problem. Fig **??** shows Batina et al.'s scheme. Again, when eavesdropping $\{T, e, y1, y2\}$ and computing $(T - y_1P_1 - y_2P_2)e^{-1}$ to obtain V , an adversary can use the computed V to track the tag [**?**]. In addition, the forward privacy and scalability problems in Batina et al.'s scheme are also similar as the situations in Tuyls et al.'s scheme.

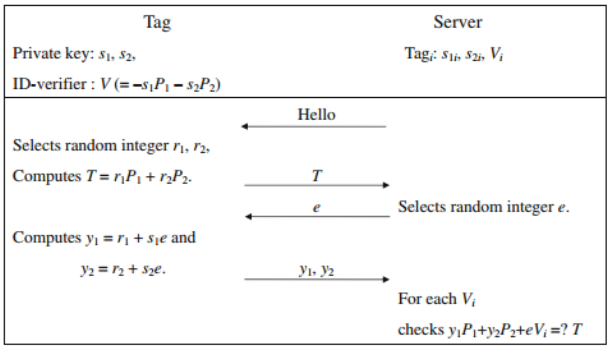

Figure 3: Batina et al.'s scheme

2.3 Lee et al.'s scheme

Lee et al.'s scheme, shown in Fig **??**, is proposed to address the ID-verifier disclosure problems in Tuyls et al.'s and Batina et al.'s schemes. Lee et al.'s. let a tag make the response using $e' = x(eP)$ rather than directly using challenge e, where e' indicates the x-coordinate of eP. This design plays a key role to resist against the possibility of linear operations on the eavesdropped data, and thus avoids privacy leakage. In addition, Lee et al.'s scheme makes the server have a private key y (an integer) and publish the corresponding public key Y (an elliptic curve point which equals to yP). In an identification process, a tag should use server's public key to make a response and the server then should apply its private key to verify the correctness of the response. The usage of this pair of private and public keys can strengthen the system security in forward privacy. However, Lee et al.'s scheme still has no scalability.

2.4 O'Neill and Robshaw's scheme

O'Neill and Robshaw proposed another solution against the linear operations on the eavesdropped transcripts. The key point is letting the tag's commitment be an integer

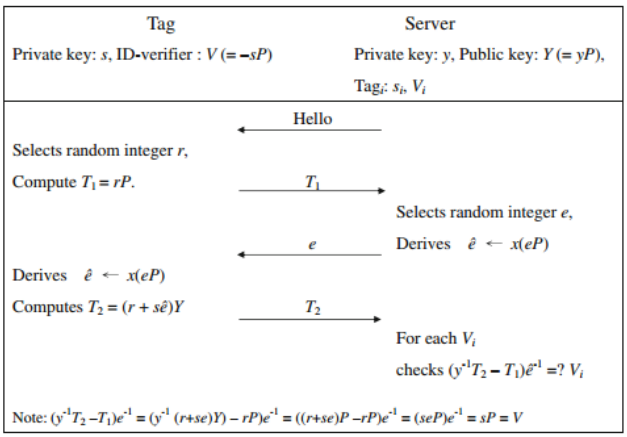

Figure 4: Lee et al.'s scheme

(a hash result) rather than an elliptic curve point. The detailed scheme is illustrated in Fig **??**. Unfortunately, O'Neill and Robshaw's scheme still lacks the scalability.

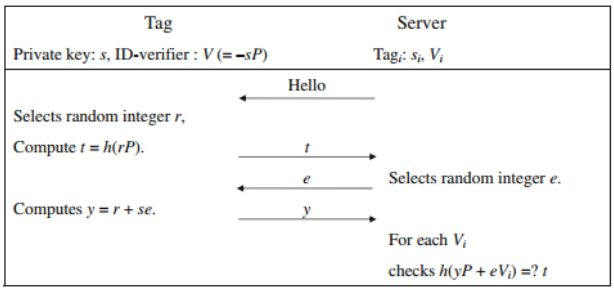

Figure 5: O'neil and Robshaw's scheme

2.5 Chou's Scheme

Chou's [?] system consists of two phases: setup phase and authentication phase. Before describing these two phases, a list of the used notations are given below:

G	A group of order q on an elliptic curve,
P	A primitive element of G,
X_i	Tag$_i$'s identifier is a random chosen point in G,
y	Server's private key,
$Y(=yP)$	Server's public key,
r, k	Two random numbers in Z_q,
h	A one-way hash function.

2.5.1 Setup phase

In this phase, the server chooses a random number $y \in Z_q$ as its private key and sets $Y(= yP)$ as its public key. It also chooses a random point $X_i \in G$ as Tag_i's identifier ID_i and then stores each Tag_i's identifier and related information in its database, where the information includes the name of the tag and production number, and so on. Finally, the server stores $[Xi, Y, P]$ in each Tag_i's memory, $i = 1$ to N, N is the number of tags.

2.5.2 Authentication phase

When interrogating a set of tags, the server broadcasts a random point. Each tag in the range of the interrogation signal performs the proposed authentication protocol shown in Fig **??** with the server.

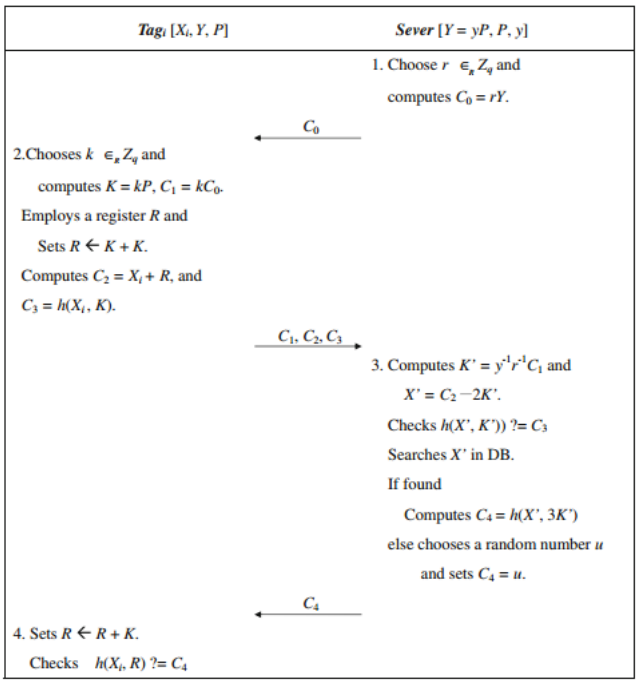

Figure 6: Chou's scheme

Step 1 The server chooses a random integer $r \in Z_q$ and computes $C_0 = rY$. It then broadcasts interrogation message C_0 to the Tag_i.

Step 2 On receiving the interrogation, Tag_i picks a random integer $k \in Z_q$, computes $K = kP$ and $C_1 = kC_0$. Tag_i continues to set a register R as $K + K$ and computes $C_2 = X_i + R$ and $C_3 = h(X_i, K)$. Then Tag_i sends $\{C1, C2, C3\}$ to the server.

Step 3 On receiving the message $\{C1, C2, C3\}$, the server utilizes its private key y to extract $K' = y^{-1}r^{-1}C_1$ (supposed to be equal to $y^{-1}r^{-1}kC_0 = y^{-1}r^{-1}kryY = kP$ and computes candidate tag identifier $X' = C_2 - 2K'$ (supposed to be equal to $X_i + 2K - 2K' = X_i$). The server continues computing a hash value, $h(X', K')$, and compares the hash result with the received C_3. If they are equal, the server directly fetches X' from its database. If succeeds, the server authenticate the Tag_i's identity, and it will authenticate itself to the Tag_i by making a hash value $C_4 = h(X_i, 3K')$. If the candidate X' is not found in the server's database, the server sets C_4 as a random integer u to prevent possible location privacy leakage. Finally, the server returns C_4 to the Tag_i.

Step 4 On receiving C_4, the Tag_i uses this value to authenticate the validity of the server. Tag_i increments the register R by K (now the value in register R is $3K$) and computes a hash value $h(X_i, R)$. Then Tag_i compares the hash result with the received C_4. If they are equal, Tag_i believes that the counterpart is the true server.

2.6 Security analysis of Chou's Scheme provided by Chou

This section analyzes Chou's RFID authentication protocol in terms of the six important security issues (location privacy, forward privacy after physical attacks, mutual authentication, replay attacks, man-in-the-middle attacks, and impersonation attacks).

2.6.1 Privacy

Assume that A (A is an adversary) can deduce kP from $2kP$, then the ECDLP is broken. Without loss of generality, we let $k = 2v$, $v \in N$. Then, $2kP = 2^{v+1}P$. According to the hypothesis that since $2kP$ one can deduce kP, the adversary therefore can iterate on deducing $(k/2)P$ and then $(k/4)P$. This process can continue until reaching the generator point P. The number of iterations can be easily seen to be $log2(2^{v+1}) = v+1$. In other words, it is feasible for A to know that the scalar part of the point multiplication equals 2^{v+1}. Therefore, ECDLP is broken.

2.6.2 Physical Attack

Physical attacks usually cannot be prevented if lowcost tags are not equipped with tamper-resistant device. Therefore, a practical RFID system should at least ensure that (1) other tags' secrets or servers' secrets cannot be further compromised, and (2) past conversations of the corrupted tag cannot be distinguished. Item (2) is referred as forward privacy, which Chou claims that his proposal possesses. We now examine item (1). Assume A (A is an adversary) uses physical means to obtain the secret X_1 of tag_1. Could he then extract tag_2's secret X_2 or server's private key y? For the break of the value X_2 in tag_2, it is impossible, since X_2 is randomly chosen and independent to X_1. Considering the server's private key, knowing that X_1, $C_0(= ryP)$, $C_1(= kryP)$, $C2(= X_1 + 2kP)$, and $C_3(= h(X_1, kP))$, A can only extract $2kP$ but not kP, r, k, or y. The infeasibility of deducing kP has been shown in the above paragraph, whereas the break of value r, k, and y are related to ECDLP.

2.6.3 Mutual Authentication

In Chou's scheme, the server computes a candidate identifier X' and then compares $h(X', K')$ with the received $C_3(= h(X_i, K))$. This mechanism makes the server verify whether tag_i is valid or not. On the other hand, tag_i also computes a hash value $h(X_i, 3K)$ and checks whether the hash result is equal to the received C_4 (sent from the server). This mechanism also makes the tag confirm the server's validity. We know that only a valid tag_i makes a valid X_i embedded in the response message to let the server find it in the database. In addition, only the legal server, having the right X_i, can return the correct C_4 to pass the tag's examination. Thus, Chou claims to have mutual authentication in his protocol.

2.6.4 Replay Attack

When A has eavesdropped on a conversation between a tag and a server and has obtained $\{C0, C1, C2, C3, C4\}$, can A successfully authenticate himself to the server by just replaying $\{C1, C2, C3\}$ in a new session? Chou claims that such an attack is unlikely to succeed, because the server will issue a fresh interrogation, say ${C_0}^{(new)}$, and thus make the replayed response $\{C1, C2, C3\}$ inapplicable for the new session. On the other hand, can A attain any advantage by replaying $C_0(= rY)$ to the tag? Chou believes this is also unlikely. For example, the tag, on receiving an old C_0, will generate a fresh random integer $k^{(new)}$ and answer a new response $\{{C_1}^{(new)}, {C_2}^{(new)}, {C_3}^{(new)}\}$ based on both the old C_0 and $k^{(new)}$. Thus, A still cannot extract any secrets, including the tag's secret X, the server's secret y, or the one-time random integers r, k, and $k^{(new)}$, owing to ECDLP.

2.6.5 Impersonation Attack

If E wants to impersonate a tag to a server, he will fail because E must use Tag_i's secret X_i to compute a valid C_2 and C_3 to pass the server's examination. On the other hand, if E impersonates a server to a tag, he has to use the correct X_i to compute C_4. However, without the serve's computed value $C_4 = h(X_i, 3kP)$, it would be impossible for E to pass Tag_i's authentication.

2.7 Weaknesses in Chou's Scheme Pointed out by Farash

Farash [**?**] pointed out three vulnerabilities in Chou's Scheme:

2.7.1 Lack of Tag Privacy

Tag privacy relies on the inability of the adversary to learn the tag's identifier X_i. However, the tag's identifier can easily be obtained from the tag in Chou's scheme, without physical attacks. To do so, the adversary A performs the following steps with Tag_i as shown in Fig **??**:

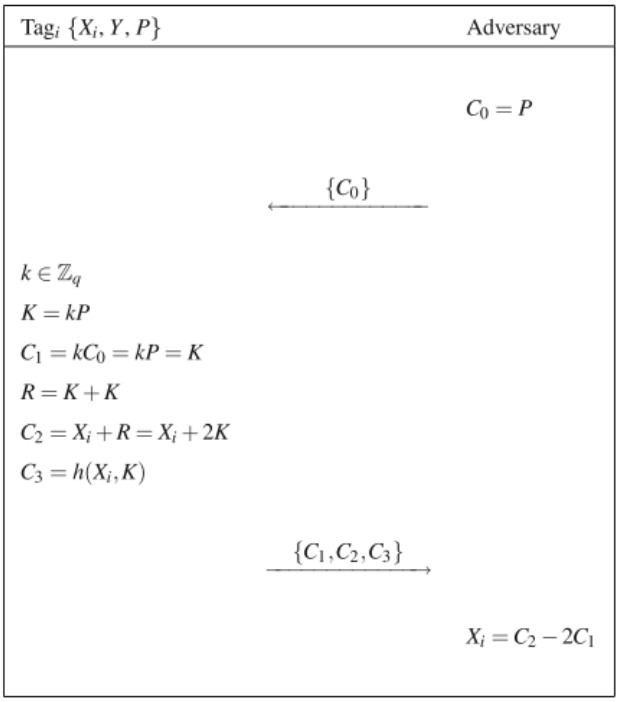

Figure 7: Breaking the privacy of Chou's scheme

Step 1: The adversary A generates and sends the message $C_0 = P$ to the Tag_i.

Step 2: On receiving the interrogation, Tag_i picks a random integer $k \in Z_q$ and computes $K = kP$ and $C_1 = kC_0 = kP = K$. Tag_i then sets a register R as $K + K$ and computes $C_2 = X_i + R = X_i + 2K$ and $C_3 = h(X_i, K)$. Then Tag_i sends $\{C_1, C_2, C_3\}$ to the server.

Step 3: The adversary A intercepts the message $\{C_1, C2, C3\}$. Since $C_1 = K$ and $C_2 = X_i + 2K$, the adversary A can obtain the Tag_i's identifier X_i as follows: $C_2 - 2C_1 = (X_i + 2K) - 2K = X_i$

2.7.2 Lack of Forward Privacy

Forward privacy relies on the inability of the adversary to track Tag_i by knowing the identifier X_i. Chou's scheme lacks forward privacy. This is because when an adversary performs above-mentioned steps and obtains the identifier X_i of a specific tag Tag_i, he/she can use this X_i to determine whether a past conversation, $\{C_0{}^*, C_1{}^*, C_2{}^*, C_3{}^*\}$,

belongs to the specific tag by computing $K^* = 2^{-1}(C_2{}^* - X_i)$, and evaluating the equation $h(X_i, K^*)? = C_2{}^*$. Therefore, Chou's scheme is vulnerable to location tracking attacks.

2.7.3 Lack of Mutual Authentication

After obtaining the Tag_i's identifier X_i, the adversary A can impersonate not only Tag_i but also the server. To impersonate Tag_i, the adversary A performs same as the actual tag because he/she knows the secret identifier X_i. To impersonate the server, the adversary A can continue the attack described in the subsection G(1) by sending $C_4 = h(X_i, 3C_1) = h(X_i, 3K)$ to Tag_i's. On receiving C_4, Tag_i's compares it with $h(X_i, 3K)$, and accepts it because they are equal. Therefore, the adversary A have succeeded to masquerade as the legal server. Therefore, Chou's protocol does not achieve tag authentication, server authentication, and mutual authentication.

2.8 Weaknesses in Chou's Scheme Pointed out by Zhang

2.8.1 Tag information privacy and impersonation

In RFID-based system, all messages are transmitted through radio wave, which is a wireless communication technology. Then the adversary could intercept or modify message transmitted between the read and the tag. Although Chou demonstrated that their protocol is secure against various attacks. As shown in Fig **??**, Zhang [**?**] shows that the adversary could get the ith tag's identifier X_i through the following steps.

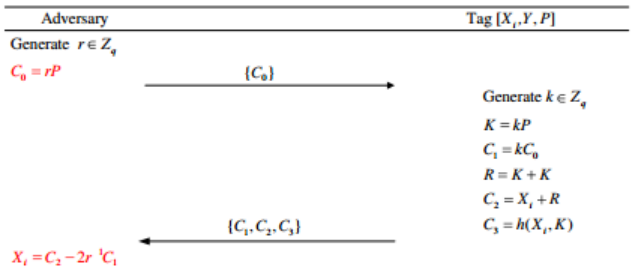

Figure 8: Attack on Chou's protocol

i). The adversary generates a random number $r \in Z_q$, sets $C_0 = rP$ and sends the message $\{C0\}$ to the ith tag.

ii). On receiving $\{C0\}$, the ith tag generates a random number $k \in Z_q$, computes $K = kP$, $C_1 = kC_0$, $R = K + K$, $C_2 = X_i + R$ and $C_3 = h(X_i, K)$. Then the ith tag sends the message $\{C_1, C_2, C_3\}$ to the adversary.

iii). The adversary computes $X_i = C_2 - 2r^{-1}C_1$.

Since $C_0 = rP$, $K = kP$, $C_1 = kC_0$, $R = K + K$, $C_2 = X_i + R$, then we could get

$$\begin{aligned} & C_2 - 2r^{-1}C_1 = X_i + R - 2r^{-1}kC_0 \\ &= X_i + 2K - 2r^{-1}krP \\ &= X_i + 2kP - 2kP \\ &= X_i \end{aligned}$$

According to the above description, we know that the adversary could get the ith tag's identifier X_i successfully. Using the value X_i, the adversary could generate a legal message $\{C_1, C_2, C_3\}$ upon receiving the message $\{C_0\}$ sent by the server. Therefore, the adversary could impersonate the ith tag to the server.

2.8.2 Backward traceability and forward traceability problem

Suppose the adversary could get the tag's identifier X_i stored in the tag. Then he could trace the tag by confirming whether the message is transmitted by the tag.

i). The adversary gets tag's identifier X_i.

ii). The adversary collects the message $\{C_0\}$, $\{C_1, C_2, C_3\}$ and $\{C_4\}$ transmitted between the server and the tag, where $C_0 = rY$, $K = kP$, $C_1 = kC_0$, $R = K + K$, $C_2 = X_i + R$, $C_3 = h(X_i, K)$ and $C_4 = h(X_i, 3K)$.

iii). The adversary computes $R' = C_2 X_i$, $K' = 2^{-1}R'$ and checks whether C_3 and $h(X_i, K)$ are equal. If they are equal, the adversary could confirm the message is transmitted by the tag.

According to the above attack, we know that the adversary could confirm whether the message is sent by the tag. Then he could trace the tag through the above attack. Therefore, Chou's protocol suffers from the backward traceability and forward traceability problem.

2.9 Farash's Improved Scheme

To solve the security problems of RFID authentication protocols, Farash proposed an improved ECC-based protocol. When interrogating a set of tags, the server broadcasts a random point. Each tag in the range of the interrogation signal performs the authentication protocol shown in Fig **??** as follows:

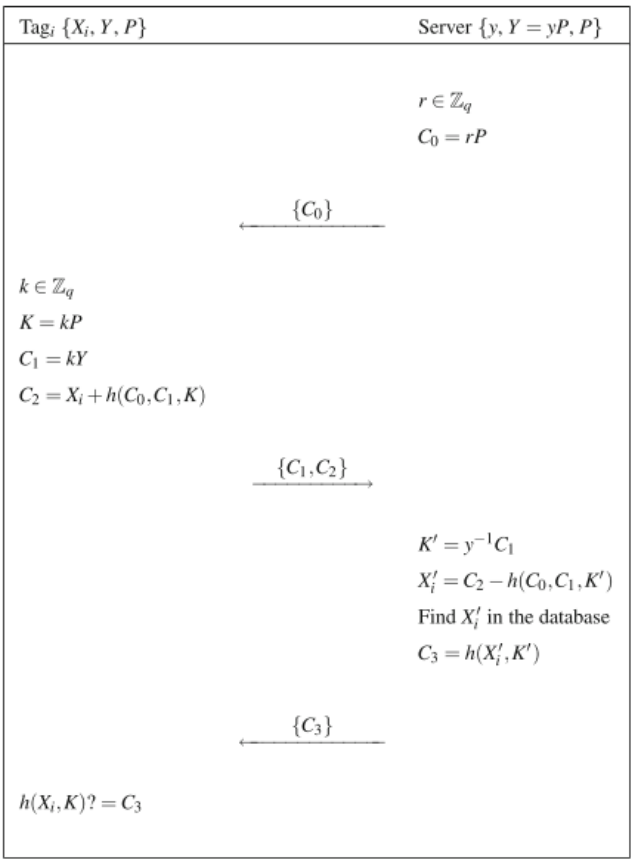

Figure 9: Farash's proposed protocol

Step 1: The server chooses a random integer r Zq, computes
$C_0 = rP$

and broadcasts interrogation message C_0 to the Tag_i.

Step 2: On receiving the interrogation message C_0, Tag_i picks a random integer $k \in Z_q$ and computes
K=kP
$C_1 = kY$
$C_2 = X_i + h(C_0, C_1, K)$

Tag_i then sends $\{C_1, C_2\}$ to the server.

Step 3: On receiving the message $\{C_1, C_2\}$, the server extracts
K'=$y^{-1}C_1$

and computes candidate tag identifier

X'$_i = C_2 - h(C_0, C_1, K')$

The server then directly fetches X_i' from its database. If succeeds, the server makes a hash value

$C_3 = h(X_i', K')$

Finally, the server returns C_3 to the Tag_i

Step 4: On receiving C_3, the Tag_i checks if

$h(X_i, K)? = C_3$

If it holds, Tag_i believes that the counterpart is the true server.

Farash claims that his protocol is secure against Replay Attack, Man-in the-middle attack and Impersonation attack. he again claims that his protocol provides Mutual authentication, Location privacy and forward privacy.

2.10 Zhang's improved scheme based on Chou's scheme

Zhang proposed an improved RFID authentication protocol based on Chou's protocol. There are two phases in the proposed protocol, i.e., the setup phase and the authentication phase.

2.10.1 Setup phase

The server generates his private key and public key in this phase. He also generates the identifier of each tag.

i). The server generates a random number $y \in Z_q$ as his private key and computes his public key $Y = yP$.

ii). The server chooses a random point X_i in G as the ith tag's identifier. Then the server stores the ith tag's identifier and related information in its database. The server also stores $[X_i, Y, P]$ into the ith tag's memory.

2.10.2 Authentication phase

When interrogating a tag, the server starts the phase to authenticate each other. As show in Fig **??**, the details of the phase are presented as follows.

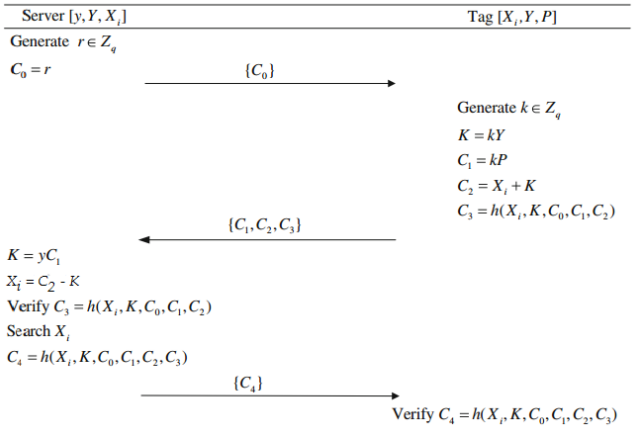

Figure 10: Zhang's proposed protocol

i). The server generates a random number $r \in Z_q$, sets $C_0 = r$ and sends the message $\{C_0\}$ to the ith tag.

ii). On receiving $\{C_0\}$, the ith tag generates a random number $k \in Z_q$, computes $K = kY$, $C_1 = kP$, $C_2 = X_i + K$ and $C_3 = h(X_i, K, C_0, C_1, C_2)$, $C_3 = h(X_i, K)$. Then the ith tag sends the message $\{C_1, C_2, C_3\}$ to the server.

iii). Upon receiving $\{C_1, C_2, C_3\}$, the server computes $K = yC_1$ and $X_i = C_2R$. Then the server checks whether C_3 and $h(X_i, K, C_0, C_1, C_2)$ are equal. If they are not equal, the server rejects the session; otherwise, the server searches its database for X_i. If succeeds, the server is confirm that the tag is a legal one; then, the server computes $C_4 = h(X_i, K, C_0, C_1, C_2, C_3)$ and the message $\{C_4\}$ to the ith tag.

iv). Upon receiving $\{C_4\}$, the ith tag checks whether C_4 and $h(X_i, K, C_0, C_1, C_2, C_3)$ are equal. If they are not equal, the ith tag rejects the session; otherwise, the ith tag confirms that the server is a legal one.

2.11 Security analysis on Zhang's protocol performed by Zhang

2.11.1 Tag information privacy

Suppose that the adversary generates a random number $r \in Z_q$, sets $C_0 = r$ and sends the message $\{C_0\}$ to the ith tag. Upon receiving $\{C_0\}$, the ith tag generates a random number $k \in Z_q$, computes $K = kY$, $C_1 = kP$, $C_2 = X_i + K$ and $C_3 = h(X_i, K, C_0, C_1, C_2)$, $C_3 = h(X_i, K)$. Then the ith tag sends the message $\{C_1, C_2, C_3\}$ to the adversary. If the adversary wants to get X_i from $C_2 = X_i + K$, he has to compute $K = kY$ from $C_1 = kP$ and $Y = yP$. Then he will face the computational Diffie-Hellman problem. Thus, the proposed RFID authentication protocol could overcome weaknesses in Chou's protocol and provide tag information privacy.

2.11.2 Mutual authentication

Without the tag's identifier X_i, the adversary cannot generate a message $\{C_1, C_2, C_3\}$, where $K = kY$, $C_1 = kP$, $C_2 = X_i + K$ and $C_3 = h(X_i, K, C_0, C_1, C_2)$. Then the server could authenticate the tag by checking the correctness of C_3. Without the tag's identifier X_i and the server's private key y, the adversary cannot generate a message $\{C_4\}$, where $C_4 = h(X_i, K, C_0, C_1, C_2, C_3)$. Then the tag could authenticate the server by checking the correctness of C_4. Thus, the proposed RFID authentication protocol could provide mutual authentication.

2.11.3 Tag anonymity

Suppose that the adversary could intercept the message $\{C_0\}$, $\{C_1, C_2, C_3\}$ and $\{C_4\}$ transmitted between the server and the tag, where $C_0 = r$, $K = kY$, $C_1 = kP$, $C_2 = X_i + K$, $C_3 = h(X_i, K, C_0, C_1, C_2)$ and $C_4 = h(X_i, K, C_0, C_1, C_2, C_3)$. If the adversary wants to get the tag's identifier X_i, he has to compute $K = kY$ from $C_1 = kP$ and $Y = yP$. Then he will face the computational Diffie-Hellman problem. Thus, the proposed RFID authentication protocol could overcome weaknesses in Chou's protocol and provide anonymity.

2.11.4 Backward traceability and forward traceability

Suppose that the adversary could get the identifier X_i of the ith tag. Suppose he could also intercept a message $\{C_0\}$, $\{C_1, C_2, C_3\}$ and $\{C_4\}$ transmitted between the server and the tag, where $C_0 = r$, $K = kY$, $C_1 = kP$, $C_2 = X_i + K$, $C_3 = h(X_i, K, C_0, C_1, C_2)$ and $C_4 = h(X_i, K, C_0, C_1, C_2, C_3)$. If he wants to verify whether those messages $\{C_0\}$, $\{C_1, C_2, C_3\}$ and $\{C_4\}$ are transmitted between the ith tag and the server, he has to compute $K = kY$ from $C_1 = kP$ and $Y = yP$. Then he will face the computational Diffie-Hellman problem. Thus, the proposed RFID authentication protocol could provide backward traceability and forward traceability.

2.11.5 Tag impersonation attack

Suppose that the adversary wants to impersonate the ith tag to the server when he intercepts the message $\{C_0\}$ sent by the server. He has to generate a legal message $\{C_1, C_2, C_3\}$, where $K = kY$, $C_1 = kP$, $C_2 = X_i + K$ and $C_3 = h(X_i, K, C_0, C_1, C_2)$. However, he cannot generate C_3 without the ith tag's identifier X_i. Thus, the proposed RFID authentication protocol could withstand the tag impersonation attack.

2.11.6 Server spoofing attack

Suppose that the adversary wants to impersonate the server to the ith tag. He could generates a random number $r \in Z_q$, sets $C_0 = r$ and sends the message $\{C_0\}$ to the ith tag. However, he cannot generate the message $\{C_4\}$ without the ith tag's identifier X_i and the server's private key y, where $C_0 = r$, $K = kY$, $C_1 = kP$, $C_2 = X_i + K$, $C_3 = h(X_i, K, C_0, C_1, C_2)$ and $C_4 = h(X_i, K, C_0, C_1, C_2, C_3)$. Thus, the proposed RFID authentication protocol could withstand the server spoofing attack.

2.11.7 Replay attack

Suppose the adversary intercepts the message $\{C_0\}$ and replays it to the ith tag. However, when he receives the message $\{C_1, C_2, C_3\}$, the adversary cannot generate the message $\{C_4\}$ without the ith tag's identifier X_i and the server's private key y, where $C_0 = r$, $K = kY$, $C_1 = kP$, $C_2 = X_i + K$, $C_3 = h(X_i, K, C_0, C_1, C_2)$ and $C_4 = h(X_i, K, C_0, C_1, C_2, C_3)$. Then the ith tag could find the replay attack by checking the correctness of C4.

Suppose the adversary intercepts the message $\{C_1, C_2, C_3\}$ and replays it to the ith tag when he receives the message $\{C_0\}$, where $C_0 = r$, $K = kY$, $C_1 = kP$, $C_2 = X_i + K$, $C_3 = h(X_i, K, C_0, C_1, C_2)$. The server could find the attack by checking the correctness of C_3 since he generates a new random number r for each session.

Thus, the proposed RFID authentication protocol could withstand the replay attack.

2.11.8 DoS attack

According to the description of the proposed protocol, we know that the ith tag and the server do not need to update the ith tag's identifier X_i. Thus, the proposed RFID authentication protocol could withstand the DoS attack.

2.11.9 Modification attack

Suppose that the adversary intercepts the message $\{C_0\}$ or $\{C_4\}$ and send it to the ith tag after modification, where $C_0 = r$, $K = kY$, $C_1 = kP$, $C_2 = X_i + K$, $C_3 = h(X_i, K, C_0, C_1, C_2)$ and $C_4 = h(X_i, K, C_0, C_1, C_2, C_3)$. The ith tag could find the attack by checking the correctness of C_4. Suppose the adversary intercepts the message $\{C_1, C_2, C_3\}$ and send it to the server after modification. The server could also find the attack by checking the correctness of C_3. Thus, the proposed protocol could withstand the modification attack.

2.11.10 De-synchronization attack

According to the description of the proposed protocol, we know that the ith tag and the server do not need to update the ith tag's identifier X_i. Thus, the proposed RFID authentication protocol could withstand the de-synchronization attack.

2.11.11 Man-in-the-middle attack

The proposed RFID authentication protocol could provide mutual authentication between the tag and the server. Thus, the proposed RFID authentication protocol could withstand the man-in-the-middle attack.

2.12 Liao and Hsiao's protocol

Liao and Hsiao's [?] protocol consists of two phases, i.e., the setup phase and the authentication phase. For convenience, notations used in the paper are presented as follows.

- q,n: two large prime numbers.
- $F(q)$: a finite filed.
- E: an elliptic curve defined by the equation $y^2=x^3+ax+b$, where $a,b \in F(q)$.
- P: a generator point with order n.
- (x_S,P_S): the private/public key pair of the server, where $P_S=x_SP$.
- (x_T,Z_T): the private keys of the tag, where $Z_T=x_TP$.
- Computational Diffie-Hellman problem: For given aP,bP, the task of the computational Diffie-Hellman problem is to compute abP.

2.12.1 Setup phase

In this phase, systems parameters, private keys and public keys will be generated for the server and the tag.

i). The server chooses the elliptic curve domain parameters $\{q, a, b, P, n\}$.

ii). The server generates a random number $x_S \in Z_n$ as its private key and computes the public key $P_S = x_SP$.

iii). For each tag, the server generates a random number $x_T \in Z_n$ as the tag's private key and computes the tag's public key or ID-verifier $Z_T = x_TP$. The server stores (Z_T, x_T) in its database. The server also stores $\{q, a, b, P, n\}$, (Z_T, x_T) and P_S into the tag's memory.

2.12.2 Authentication phase

In this phase, the server and the tag could authenticate each other. As shown in Fig **??**, the details are described as follows.

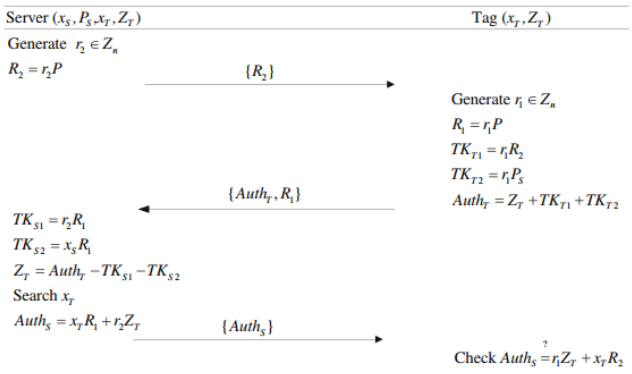

Figure 11: Liao's proposed protocol

i). The server generates a random number $r_2 \in Z_n$ and computes $R_2 = r_2P$. Then the server sends the message $\{R_2\}$ to the tag.

ii). Upon receiving $\{R_2\}$, the tag generates a random number $r_1 \in Z_n$ and computes $R_1 = r_1P$. The tag also computes $TK_{T1} = r_1R_2$, $TK_{T2} = r_1P_S$ and $Auth_T = Z_T + TK_{T1} + TK_{T2}$. Then the tag sends the message $\{Auth_T, R_1\}$ to the server.

iii). Upon receiving $\{Auth_T, R_1\}$, the server computes $TK_{S1} = r_2R_1$, $TK_{S2} = x_SR_1$ and $Z_T = Auth_T - TK_{S1} - TK_{S2}$. Then, the server searches Z_T in its database. If it is not found, the server stops the session; otherwise, the sever obtains the corresponding private key x_T and computes $Auth_S = x_TR_1 + r_2Z_T$. Then the server sends the message $\{Auth_S\}$ to the tag.

iv). Upon receiving $\{Auth_S\}$, the tag checks whether $Auth_S$ and $r_1Z_T + x_TR_2$ are equal. If they are not equal, the tag stops the session; otherwise, the server is authenticated.

2.13 Security analysis of Liao and Hsiao's protocol

Liao and Hsiao claimed that their protocol could withstand various attacks. However, Zhao [**?**] showed that Liao's protocol suffers from the key compromise problem, i.e., an adversary could get the secret key Z_T. Since the channel between the reader and the tag is not secure. We could assume that the adversary could control the channel totally, i.e., he could send, modify and replay a message at his will. The attack is presented as follows.

i). The adversary generates a random number $r_2 \in Z_n$ and computes $R_2 = r_2PP_S$. Then the adversary sends the message $\{R_2\}$ to the tag.

ii). Upon receiving $\{R_2\}$, the tag generates a random number $r_1 \in Z_n$ and computes $R_1 = r_1P$. The tag also computes $TK_{T1} = r_1R_2$, $TK_{T2} = r_1P_S$ and $Auth_T = Z_T + TK_{T1} + TK_{T2}$. Then the tag sends the message $\{Auth_T, R_1\}$ to the adversary.

iii). Upon receiving $\{Auth_T, R_1\}$, the adversary compute $Z_T = Auth_T - r_2R_1$.

Since $R_2 = r_2P - P_S$, $R_1 = r_1P$, $TK_{T1} = r_1R_2$, $TK_{T2} = r_1P_S$ and $Auth_T = Z_T + TK_{T1} + TK_{T2}$, then we have

$\text{Auth}_T - r_2R_1 = Z_T + TK_{T1} + TK_{T2} - r_2R_1$
$= Z_T + r_1R_2 + r_1P_S - r_2r_1P$
$= Z_T + r_1(r_2P - P_S) + r_1P_S - r_1r2P$
$= Z_T + r_1r_2P - r_1P_S + r_1P_S - r_1r_2P$
$= Z_T$

Then, the adversary could get the tag's secret key Z_T. Using the secret key, the adversary could generate a legal message $\{Auth_T, R_1\}$ upon receiving the message $\{R_2\}$ sent by the server. Therefore, Liao and Hsiao's protocol is not secure at all.

2.14 Zhenguo Zhao's Protocol

To solve the security problem in Liao and Hsiao's protocol, Zhao proposed a new RFID authentication protocol using ECC. The proposed protocol also consists of two phases, i.e., the setup phase and the authentication phase.

2.14.1 Setup phase

In this phase, systems parameters, private keys and public keys will be generated for the server and the tag.

i). The server chooses the elliptic curve domain parameters $\{q, a, b, P, n\}$.

ii). The server generates a random number $x_S \in Z_n$ as its private key and computes the public key $P_S = x_S P$.

iii). For each tag, the server generates a random number $x_T \in Z_n$ as the tag's private key and computes the tag's public key $Z_T = x_T P$. The server stores (Z_T, x_T) in its database. The server also stores $\{q, a, b, P, n\}$, (Z_T, x_T) and P_S into the tag's memory.

2.14.2 Authentication phase

In this phase, the server and the tag could authenticate each other. As shown in Fig **??**, the details are described as follows.

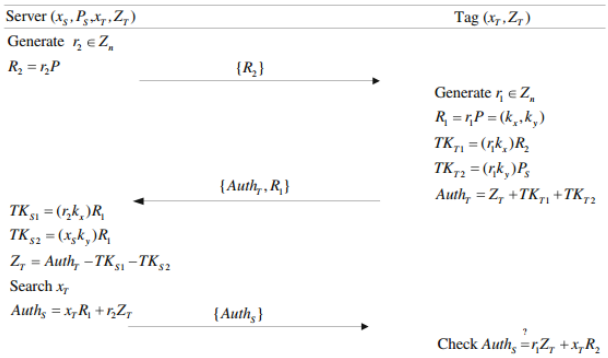

Figure 12: Zhao's proposed protocol

i). The server generates a random number $r_2 \in Z_n$ and computes $R_2 = r_2 P$. Then the server sends the message $\{R_2\}$ to the tag.

ii). Upon receiving $\{R_2\}$, the tag generates a random number $r_1 \in Z_n$ and computes $R_1 = r_1 P = (k_x, k_y)$. The tag also computes $TK_{T1} = (r_1 k_x) R_2$, $TK_{T2} = (r_1 k_y) P_S$ and $Auth_T = Z_T + TK_{T1} + TK_{T2}$. Then the tag sends the message $\{Auth_T, R_1\}$ to the server

iii). Upon receiving $\{Auth_T, R_1\}$, the server computes $TK_{S1} = (r_2 k_x) R_1$, $TK_{S2} = (x_S k_y) R_1$ and $Z_T = Auth_T - TK_{S1} - TK_{S2}$. Then, the server searches Z_T in its database. If it is not found, the server stops the session; otherwise, the sever obtains the corresponding private key x_T and computes $Auth_S = x_T R_1 + r_2 Z_T$. Then the server sends the message $\{AuthS\}$ to the tag.

iv). Upon receiving $\{Auth_S\}$, the tag checks whether $Auth_S$ and $r_1 Z_T + x_T R_2$ are equal. If they are not equal, the tag stops the session; otherwise, the server is authenticated.

2.15 Security analysis on Zhao's protocol provided by Zhao

Theorem 1

The proposed protocol could overcome the key compromise problem in Liao and Hsiao's protocol.

Proof

Suppose the adversary generates a random number $r_2 \in Z_n$, computes $R_2 = r_2PP_S$ and sends the message $\{R_2\}$ to the tag. Upon receiving $\{R_2\}$, the tag generates a random number $r_1 \in Z_n$, computes $R_1 = r_1P$, $TK_{T1} = (r_1k_x)R_2$, $TK_{T2} = (r_1k_y)P_S$ and $Auth_T = Z_T + TK_{T1} + TK_{T2}$. Then the tag sends the message $\{Auth_T, R_1\}$ to the adversary. Since $R_2 = r_2P - P_S$, $R_1 = r_1P$, $TK_{T1} = (r_1k_x)R_2$, $TK_{T2} = (r_1k_y) - P_S$ and $Auth_T = Z_T + TK_{T1} + TK_{T2}$, the we have

$\text{Auth}_T = Z_T + TK_{T1} + TK_{T2}$
$= Z_T + (r_1k_x)R_2 + (r_1k_y)P_S$
$= Z_T + (r_1k_x)(r_2P - P_S) + (r_1k_y)P_S$
$= Z_T + (r_2k_x)R_1 + (k_yk_x)r_1P_S$

The adversary could compute $(r_2k_x)R_1$. However, he cannot compute $r_1P_S = (r_1x_S)P$ since he will be faced with the computational Diffie-Hellman problem. Therefore, the adversary cannot get the private key Z_T and the proposed protocol could overcome the key comprise problem in Liao and Hsiao's protocol.

Theorem 2

The proposed protocol could provide mutual authentication between the tag and the server.

Proof

The adversary cannot generate a legal message $\{Auth_T, R_1\}$ without the knowledge Z_T, where $Auth_T = Z_T + TK_{T1} + TK_{T2}$, $R_1 = r_1P = (k_x, k_y), TK_{T1} = (r_1k_x)R_2 and TK_{T2} = (r_1k_y)P_S$. Then the server could authenticate the tag by checking the correctness of $Auth_T$.

The adversary cannot generate a legal message $\{Auth_S\}$ without the knowledge of x_T and Z_T, where $Auth_S = x_TR_1 + r_2Z_T$. Then the tag could authenticate the server through checking the correctness of $Auth_S$.

Therefore, the proposed protocol could withstand mutual authentication between the tag and the server.

Theorem 3

The proposed protocol could provide anonymity.

Proof

In the proposed protocol, the tag's identity Z_T is included in the message $Auth_T = Z_T + TK_{T1} + TK_{T2}$ and $Auth_S = x_TR_1 + r_2Z_T$, where $R_2 = r_2P$, $R_1 = r_1P = (k_x, k_y)$, $TK_{T1} = (r_1k_x)R_2$ and $TK_{T2} = (r_1k_y)P_S$. The adversary cannot compute x_TR_1 since he does not know x_T. Then he cannot get Z_T from $Auth_S$ either. Therefore, the adversary cannot get the tag's identity Z_T and the proposed protocol could provide anonymity.

Theorem 4

The proposed protocol could provide availability.

Proof

From the description of the proposed protocol, we know that no synchronously update of the secret key is needed in the execution of the protocol. Therefore, the proposed could be executed between the server and the tag. Therefore, the proposed protocol could provide availability.

Theorem 5

The proposed protocol could provide forward security.

Proof

Suppose that the adversary could get the tag's secret key x_T and Z_T. However, he cannot determine whether the messages $\{R_2\}$, $\{Auth_T, R_1\}$ and $\{Auth_S\}$ are transmitted between the tag and the server since he does not know the random numbers r_1 and r_2. Therefore, the adversary cannot trace the tag and the proposed protocol could provide forward security.

Theorem 6

The proposed protocol could withstand replay attack.

Proof

Suppose that the adversary intercepts the message $\{R_2\}$ and replay it to the tag. However, he cannot generate $Auth_S = x_T R_1 + r_2 Z_T$ upon receiving the message $\{Auth_T, R_1\}$ since the tag generates a new random number r_1 and he does not know the tag's secret keys x_T and Z_T, where $R_2 = r_2 P$, $R_1 = r_1 P = (k_x, k_y)$, $TK_{T1} = (r_1 k_x) R_2$ and $TK_{T2} = (r_1 k_y) P_S$. Then the tag could find the attack by checking the correctness of $Auth_S$. From the same method, we could show the server could find the replay attack by checking the correctness of $Auth_T$. Therefore, the proposed protocol could withstand replay attack.

Theorem 7

The proposed protocol could withstand impersonation attack.

Proof

To impersonate the tag to the server, the adversary has to generate a legal message $\{Auth_T, R_1\}$ after receiving the message $\{R_2\}$ sent by the server, where $Auth_T = Z_T + TK_{T1} + TK_{T2}$, where $R_2 = r_2 P$, $R_1 = r_1 P = (k_x, k_y)$, $TK_{T1} = (r_1 k_x) R_2$ and $TK_{T2} = (r_1 k_y) P_S$. However, the adversary cannot generate $Auth_T$ since he does not know the value of Z_T. Therefore, the proposed protocol could withstand the impersonation attack.

Theorem 8

The proposed protocol could withstand server spoofing attack.

Proof

To impersonate the server to the tag, the adversary could generate a random number $r_2 \in Z_n$, compute $R_2 = r_2P$ and send $\{R_2\}$ to the tag. However, he cannot generate the message $\{Auth_S\}$ upon receiving the message $\{Auth_T, R_1\}$ sent by the tag since he does not know the tag's secret key x_T, where $Auth_S = x_TR_1 + r_2Z_T$. Therefore, the adversary cannot impersonate the server to the tag and the proposed protocol could withstand server spoofing attack.

Theorem 9

The proposed protocol could withstand DoS attack.

Proof

According to the description of the proposed protocol, there is no synchronous update of the tag's secret keys. Therefore, the proposed protocol could withstand DoS attack.

Theorem 10

The proposed protocol could withstand tracking attack.

Proof

According to Theorem 5, the adversary cannot trace the tag even if he could get the tag's secret key x_T and Z_T. Therefore, the proposed protocol could withstand tracking attack

Theorem 11

The proposed protocol could withstand cloning attack.

Proof

According to the description of the proposed protocol, we know that each tag has its own secret keys x_T and Z_T, where $Z_T = x_TP$. Suppose the adversary could get secret keys of several tags. However, he cannot get secret key of another tag since he there is no relation among tags' secret keys. Therefore, the proposed protocol could withstand cloning attack.

2.16 Debiao He's proposed protocol

Debiao He [**?**] also proposed a system based on Liao and Hsiao's protocol. The proposed ECC-based RFID authentication scheme consists of two phases, i.e., the setup phase and the authentication phase. The notations used in the scheme are defined as follows.

- n,q: two large prime numbers.
- $F(q)$: a finite field, where q represents the size of the finite field.
- (a, b): two parameters of an elliptic curve E, which is defined by the equation $y^2 = x^3 + ax + b$ over the finite field $F(q)$.
- P: a generator point with order n of the elliptic curve E.
- x_S: the private key of the server.
- P_S: the public key of the server, where $P_S = x_S P$.
- X_T: the ID-verifier of the tag.

2.16.1 Setup phase

In the setup phase, both the server and the tag will be equipped with private keys and the elliptic curve domain parameters params=$\{q, a, b, P, n\}$. The detail of the phase is presented as follows.

i) The server chooses a random number $x_S \in Z_n^*$ and computes $P_S = x_S P$. The server also chooses a random point X_T on the elliptic curve E for each tag.

ii) The server stores the ID-verifier X_T and params into the tag's memory. The server also keeps x_S as his private, and stores X_T into its database.

2.16.2 Authentication phase

In the authentication phase as shown in Fig **??**, the server and the tag will authenticate each other through the following steps.

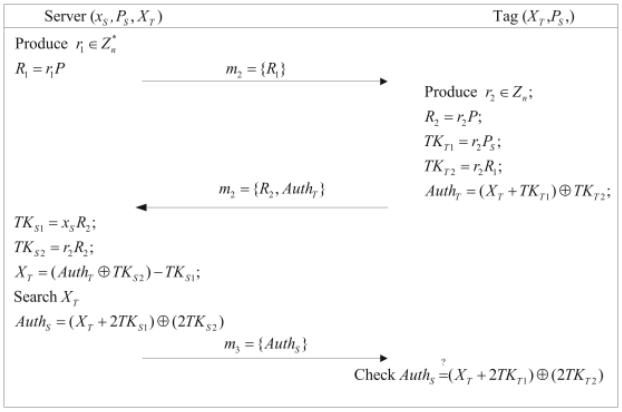

Figure 13: Debiao He's proposed protocol

i) The server produces a new random number $r_1 \in Z_n^*$ and calculates $R_1 = r_1 P$. Then the server sends the message $m1 = \{R1\}$.

ii) The tag produces a new random number $r_2 \in Z_n^*$ and calculates $R_2 = r_2 P$, $TK_{T1} = r_2 P_S$, $TK_{T2} = r_2 R_1$ and $Auth_T = (X_T + TK_{T1}) \oplus TK_{T2}$. The tag sends the message $m2 = \{R_2, Auth_T\}$ to the server.

iii) The server calculates $TK_{S1} = x_S R_2$, $TK_{S2} = r_1 R_2$ and $X_T = (Auth_T \oplus TK_{S2}) - TK_{S1}$. The server search his database for X_T. If it is not found, the server stops the session; otherwise, the server calculates $Auth_S = (X_T + 2TK_{S1}) \oplus (2TK_{S2})$ and sends the message $m3 = \{Auth_S\}$.

iv) The tag checks whether $(XT + 2TK_{T1}) \oplus (2TK_{T2})$ and $Auth_S$ are equal. If they are not equal, the tag stops the session; otherwise, the server is authenticated.

2.17 Security analysis of Debian He's protocol provided by Debian He

Mutual authentication between the tag and the server

In the Step iii of the proposed scheme, the server receives the message $m2 = \{R_2, Auth_T\}$, where $R_2 = r_2 P$, $TK_{T1} = r_2 P_S$, $TK_{T2} = r_2 R_1$ and $Auth_T = (X_T + TK_{T1}) \oplus TK_{T2}$. Without the tag's ID-verifier X_T, the adversary cannot generate the correct $Auth_T$. Then the server could compute $X_T = (Auth_T \oplus TK_{S2}) - TK_{S1}$ and authenticate the tag be checking whether X_T is stored in his database.

In the Step iv of the proposed scheme, the tag receives the message $m3 = \{Auth_S\}$, where $TK_{S1} = x_S R_2$, $TK_{S2} = r_1 R_2$ and $Auth_S = (X_T + 2TK_{S1}) \oplus (2TK_{S2})$. Without the server's secret key x_S, the adversary cannot get neither of X_T nor TK_{S1}. Then the tag could authenticate the server by verifying whether $Auth_S$ and $(X_T + 2TK_{T1}) \oplus (2TK_{T2})$ are equal. Thus, the proposed ECC-based RFID authentication scheme could provide mutual authentication between the tag and the server.

ID-verifier confidentiality

The tag's ID-verifier X_T is included in the message $m2 = \{R_2, Auth_T\}$ and $m3 = \{Auth_S\}$, where $R_2 = r_2 P$, $TK_{T1} = r_2 P_S$, $TK_{T2} = r_2 R_1$, $Auth_T = (X_T + TK_{T1}) \oplus TK_{T2}$, $TK_{S1} = x_S R_2$, $TK_{S2} = r_1 R_2$ and $Auth_S = (X_T + 2TK_{S1}) \oplus (2TK_{S2})$. Without the server's secret key x_S, the adversary cannot get X_T by decrypting $Auth_T$ or $Auth_S$. Thus, the proposed ECC based RFID authentication scheme could provide ID-verifier confidentiality.

Anonymity

The tag's ID-verifier X_T is included in the message $m2 = \{R_2, Auth_T\}$ and $m3 = \{Auth_S\}$, where $R_2 = r_2 P$, $TK_{T1} = r_2 P_S$, $TK_{T2} = r_2 R_1$, $Auth_T = (X_T + TK_{T1}) \oplus TK_{T2}$, $TK_{S1} = x_S R_2$, $TK_{S2} = r_1 R_2$ and $Auth_S = (X_T + 2TK_{S1}) \oplus (2TK_{S2})$. Without the server's secret key x_S, the adversary cannot get X_T by decrypting $Auth_T$ or $Auth_S$. Besides, the server and the tag generate new random numbers r_1 and r_2 separately in each session. Then, the adversary cannot trace the location of the tag by collecting message. Thus, the proposed ECC-based RFID authentication scheme could provide anonymity.

Availability

According to the above discussion, the tag's ID-verifier X_T is protected well when the proposed scheme is executed. Then, there is no need to update the secret ID-verifier after the execution. Thus, the proposed ECC-based RFID authentication scheme could provide availability.

Perfect forward security

perfect forward security means that the adversary cannot trace the tag even he could get the tag's ID-verifier. We suppose that the adversary could extract the tag's ID-verifier X_T and intercept messages $m1 = \{R_1\}$, $m2 = \{R_2, Auth_T\}$ and $m3 = \{Auth_S\}$ transmitted between the server and the tag, where $R_1 = r_1P$, $R_2 = r_2P$, $TK_{T1} = r_2P_S$, $TK_{T2} = r_2R_1$, $Auth_T = (X_T + TK_{T1}) \oplus TK_{T2}$, $TK_{S1} = x_SR_2$, $TK_{S2} = r_1R_2$ and $Auth_S = (X_T + 2TK_{S1}) \oplus (2TK_{S2})$. Without the server's secret key x_S and two random numbers r_1 and r_2, the adversary cannot confirm whether those messages are transmitted between the tag and the server. Thus, the proposed ECC-based RFID authentication scheme could provide perfect forward security.

Scalability

According to the Step iii of the proposed scheme, the server gets the tag's ID-verifier X_T by computing $X_T = (Auth_T \oplus TK_{S2}) - TK_{S1}$ and checking whether X_T is in the database. Then the server does not need to search the identity linearly. Thus, the proposed ECC-based RFID authentication scheme could provide scalability.

Replay attack resisting

We suppose that the adversary intercepts the messages $m1 = \{R_1\}$, $m3 = \{Auth_S\}$ and replays them to the tag, where $R_1 = r_1P$, $TK_{S1} = x_SR_2$, $TK_{S2} = r_1R_2$, $Auth_S = (X_T + 2TK_{S1}) \oplus (2TK_{S2})$ and $R_2 = r_2P$. The tag could find the attack by verifying the correctness of $Auth_S$ since it generates a new $R_2 = r_2P$ for each session. We also suppose that the adversary intercepts the message $m2 = \{R_2, Auth_T\}$ and replays it to the server, where $R_2 = r_2P$, $TK_{T1} = r_2P_S$, $TK_{T2} = r_2R_1$, $Auth_T = (X_T + TK_{T1}) \oplus TK_{T2}$. The server could find the attack by verifying the correctness of $Auth_T$ since it generates a new $R_1 = r_1P$ for each session. Thus, the proposed ECC based RFID authentication scheme could provide replay attack resisting.

Tag masquerade attack resisting

We suppose that the adversary wants to impersonate the tag to the server. Then, he has to generate a valid message $m2 = \{R_2, Auth_T\}$ when he receives the message $m1 = \{R_1\}$, where $R_1 = r_1P$, $R_2 = r_2P$, $TK_{T1} = r_2P_S$, $TK_{T2} = r_2R_1$, $Auth_T = (X_T + TK_{T1}) \oplus TK_{T2}$. It is easy to say that the adversary cannot generate a valid $Auth_T$ since he does not know the tag's ID-verifier X_T. Thus, the proposed ECC-based RFID authentication scheme could provide tag masquerade attack resisting.

Server spoofing attack resisting

We suppose that the adversary wants to impersonate the server to the tag. He could generate the message $m1 = \{R_1\}$ easily. However, he cannot generate the message $m3 = \{Auth_S\}$ when he receives the message $m2 = \{R_2, Auth_T\}$ since he does not know the tag's ID-verifier X_T and the server's secret key x_S. Thus, the proposed ECC-based RFID authentication scheme could provide server spoofing attack resisting.

DoS attack resisting

According to the above discussion, the tag's ID-verifier X_T is protected well when the proposed scheme is executed. Then, there is no need to update the secret ID-verifier after the execution. Thus, the proposed ECC-based RFID authentication scheme could DoS attack resisting.

Location tracking attack resisting

We suppose that the adversary could extract the tag's ID-verifier X_T and intercept messages $m1 = \{R_1\}$, $m2 = \{R_2, Auth_T\}$ and $m3 = \{Auth_S\}$ transmitted between the server and the tag, where $R_1 = r_1P$, $R_2 = r_2P$, $TK_{T1} = r_2P_S$, $TK_{T2} = r_2R_1$, $Auth_T = (X_T + TK_{T1}) \oplus TK_{T2}$, $TK_{S1} = x_SR_2$, $TK_{S2} = r_1R_2$ and $Auth_S = (X_T + 2TK_{S1}) \oplus (2TK_{S2})$. Without the server's secret key x_S and two random numbers r_1 and r_2, the adversary cannot confirm whether those messages are transmitted between the tag and the server. Thus, the proposed ECC-based RFID authentication scheme could provide location tracking attack resisting.

Cloning attack resisting

According to the setup phase of the proposed scheme, the server generates a random point X_T for each tag. Then, the tag keeps X_T as its ID-verifier. We suppose that the adversary could get ID-verifiers of a group of tags. However, he cannot get other tag's ID-verifier using those known ID-verifiers since they are generated randomly. Thus, the proposed ECC-based RFID authentication scheme could provide cloning tracking attack resisting.

2.18 Chunhua Jin's proposed protocol

Jin [**?**] very recently proposed a new ECC based RFID authentication protocol based on Liao and Hsiao's protocol and Debian He's protocol. And Jin claims that their scheme is more efficient since it requires lower computational cost and communication overhead.

There are two phases, i.e., the setup phase and the authentication phase. For convenience, the notations employed in this scheme are described as follows:

- q, n: Two large prime numbers.
- P : A generator with order n.
- $F(q)$: A finite field.
- E: An elliptic curve defined over a finite field F_q by the equation $y^2 = x^3 + ax + b$, where $a, b \in F(q)$.

- X_T : The identifier of the tag, where X_T is a point on the elliptic curve E.
- (x_S,P_S): The private/public key of the server, where $P_S = x_S P$, $x_S \in Z_n^*$.
- H_1,H_2: Two secure and collision-resistant hash functions.

2.18.1 Setup phase

In this phase, the server gets its public/ private key and the tag obtains its identifier. The elliptic curve system parameters params = $\{q, n, a, b, P\}$ are included both in the server and the tag. We describe the details of this phase as follows.

i). The server selects a random value $x_S \in Z_n^*$ as its private key and calculates its corresponding public key $P_S = x_S P$. Then the server stores (x_S, P_S) in its database.

ii). The server produces a random point X_T on the elliptic curve E as the tag's identifier. Then the server stores the identifier X_T and params into the tag's memory.

2.18.2 Authentication phase

In this phase, as shown in Fig **??**, the server and the tag will mutual authenticate through the following steps.

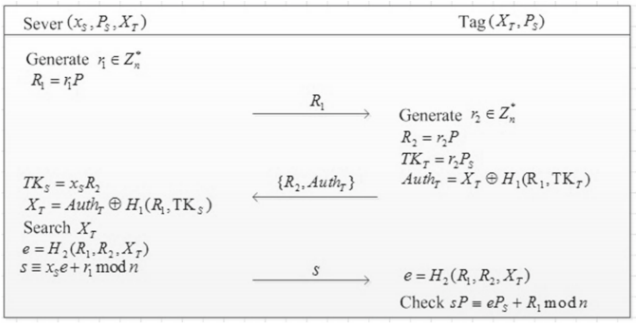

Figure 14: Jin's proposed protocol

i). The server generates a random value $r_1 \in Z_n^*$ and computes $R_1 = r_1 P$. Then the server sends the message R_1 to the tag.

ii). After receiving the message R_1, the tag produces a random value $r_2 \in Z_n^*$ and computes $R_2 = r_2 P$, $TK_T = r_1 P_S$ and $Auth_T = X_T \oplus H_1(R_1, TK_T)$. Then the tag sends the message $\{R_1, Auth_T\}$ to the server.

iii). After receiving the message $\{R_1, Auth_T\}$, the server computes $TK_S = x_S R_2$ and $X_T = Auth_T \oplus H_1(R_1, TK_S)$. Then the server searches its database for X_T . If it is not found, the server stops the session; otherwise, the server computes $e = H_2(R_1, R_2, X_T)$ and $s \equiv x_S e + r_1 \bmod n$, and sends the message s to the tag.

iv). After receiving the message s, the tag computes $e = H_2(R_1, R_2, X_T)$ and checks whether $s_P \equiv eP_S + R_1 \bmod n$. If they are not equal, the tag stops the session; otherwise, the server is authenticated.

2.19 Security analysis of Jin's protocol provided by Jin

Confidentiality

In the proposed scheme, the tag's identifier X_T is used in the message $Auth_T = X_T \oplus H_1(R_1, TK_T)$. Although the adversary can obtain the communication messages $\{R_1\}$, $\{R_2, Auth_T\}$ and the public key P_S of the server, without the random value r_2, it cannot get the tag's identifier X_T. Therefore, the proposed scheme could provide the confidentiality of the tag's identifier.

Mutual authentication

The adversary cannot generate a legitimate message $\{R_2, Auth_T\}$ since it is not able to obtain the tag's identifier X_T and the random value r_2, where $R_2 = r_2P$, $TK_T = r_2P_S$, $Auth_T = X_T \oplus H_1(R_1, TK_T)$. Then the server could compute $X_T = Auth_T \oplus H_1(R_1, TK_T)$ and authenticate the tag by checking whether X_T is saved in its database.

The adversary cannot produce a legitimate signature $\{s\}$ since it is not able to obtain the server's private key x_S, the random value r_1 and the tag's identifier X_T , where $s \equiv x_S e + r_1 \bmod n$, $e = H_2(R_1, R_2, X_T)$, $X_T = Auth_T \oplus H_1(R_1, TK_S)$, $TK_S = x_S R_2$. Then the tag could authenticate the server by verifying whether $sP \equiv eP_S + R_1 \bmod n$. Thus, the proposed scheme could provide mutual authentication.

Tag's anonymity

In the proposed scheme, the tag's identifier is used in the message $AuthT = X_T \oplus H_1(R_1, TK_T)$. Although the adversary can obtain the communication messages $\{R_1\}$, $\{R_2, Auth_T\}$ and the public key P_S of the server, without the random value r_2, it cannot compute TK_T , and hence it cannot get the tag's identifier X_T . In addition, in each new session, the server and the tag produces new random values r_1 and r_2 separately. The adversary cannot trace the tag's location. Thus, the proposed scheme could provide tag's anonymity.

Availability

In the proposed scheme, we know that the tag's identifier X_T is protected well. Any adversary cannot obtain it when the proposed scheme is executed. Therefore, after executing the proposed scheme, there is no need to update the tag's identifier. Therefore, the proposed scheme could provide availability.

Forward security

In the proposed scheme, The adversary cannot trace the tag even he could obtain the tag's identifier X_T . We suppose that the adversary could extract the tag's identifier X_T and intercept the messages $\{R_1\}$, $\{R_2, Auth_T\}$ and $\{s\}$ transmitted between the server and the tag, where $R_1 = r_1P$, $R_2 = r_2P$, $TK_T = r_2P_S$, $Auth_T = X_T \oplus H_1(R_1, TK_T)$, $TK_S = x_S R_2$, $X_T = Auth_T \oplus H_1(R_1, TK_S)$, $e = H_2(R_1, R_2, X_T)$, $s \equiv x_S e + r_1 \bmod n$. The adversary cannot confirm whether those messages are transmitted between the tag and the server since it does not know the server's private key x_S and two random values r_1 and r_2. Therefore, the proposed scheme could provide forward security.

Scalability

In the proposed scheme, the server obtains the tag's identifier X_T by computing $X_T = Auth_T \oplus H_1(R_1, TK_S)$, $TK_S = x_S R_2$ and checking whether X_T is in its database. Then the server does not search the identifier one by one. Thus, the proposed scheme could provide scalability.

Replay attack resistance

Suppose that the adversary intercepts the messages $\{R_1\}$ and $\{s\}$ and replays them to the tag, where $R_1 = r_1 P$, $TK_S = x_S R_2$, $X_T = Auth_T \oplus H_1(R_1, TK_S)$, $e = H_2(R_1, R_2, X_T)$ and $s \equiv x_S e + r_1 \bmod n$. The tag could discover the attack by verifying whether $s_P \equiv eP_S + R_1 \bmod n$. The reason is that the tag produces a new random value r_2 for each new session.

Suppose that the adversary intercepts the message $\{R_2, Auth_T\}$ and replays it to the server, where $R_2 = r_2 P$, $TK_T = r_2 P_S$, $Auth_T = X_T \oplus H_1(R_1, TK_T)$. The server could discover the attack by verifying the correctness of $Auth_T$. The reason is that it produces a new random value r_1 for each new session. Therefore, the proposed scheme could withstand the replay attack.

Tag impersonation attack resistance

Suppose that the adversary wants to impersonate the tag to the server after receiving the message R_1 sent by the server. It has to produce a legitimate message $\{R_2, Auth_T\}$ where $R_2 = r_2 P$, $TK_T = r_2 P_S$, $Auth_T = X_T oplus H_1(R_1, TK_T)$. However, since it does not know the tag's identifier X_T and the random value r_2, it cannot generate the legitimate message $\{R_2, Auth_T\}$. Thus, the proposed scheme could overcome the tag impersonation attack.

Server spoofing attack resistance

Suppose that the adversary wants to impersonate the server to the tag. It could produce a random value $r_1 \in Z_n^*$, computes $R_1 = r_1 P$ and sends R_1 to the tag. However, it cannot generate a legitimate message s since it does not obtain the tag's identifier X_T and the server's private key x_S, where $TK_S = x_S R_2$, $X_T = Auth_T \oplus H_1(R_1, TK_S)$, $e = H_2(R_1, R_2, X_T)$ and $s \equiv x_S e + r_1 \bmod n$. Thus, the adversary cannot impersonate the server to the tag and the proposed scheme could withstand the server spoofing attack.

DoS attack resistance

In the proposed scheme, we know that there is no need to synchronously update the tag's identifier X_T after the scheme is executed since the tag's identifier X_T is protected well. Therefore, the proposed scheme could overcome DoS attack.

Location tracking attack resistance

Suppose that the adversary could get the tag's identifier X_T and intercept the messages $\{R_1\}$, $\{R_2, Auth_T\}$ and $\{s\}$ transmitted between the server and the tag, where $R_1 = r_1P$, $R_2 = r_2P$, $TK_T = r_2P_S$, $Auth_T = X_T \oplus H_1(R_1, TK_T)$, $TK_S = x_SR_2$, $X_T = Auth_T \oplus H_1(R_1, TK_S)$, $e = H_2(R_1, R_2, X_T)$ and $s \equiv x_Se + r_1 \bmod n$. The adversary does not have the capabilities to obtain the server's private key x_S and two random values r_1 and r_2, so it cannot confirm whether those messages are transmitted between the server and the tag. Therefore, the proposed scheme could overcome the location tracking attack.

Cloning attack resistance

In the proposed scheme, we know that every tag has its own identifier X_T which is a random point on the ECC. Suppose that the adversary could obtain some tags' identifiers, but it cannot get other tags' identifiers since there is no relationship between these tags. Therefore, the proposed scheme could overcome cloning attack.

De-synchronization attack resistance

In the proposed scheme, since the tag's identifier is protected well, it does not need to be updated after the proposed scheme is executed. Thus, the proposed protocol could withstand the de-synchronization attack.

The man-in-the-middle attack resistance

The proposed scheme could provide mutual authentication between the tag and the server. Thus, the proposed scheme could overcome the man-in-the-middle attack.

3 The Proposed Protocol

Taking Zhao's Protocol as an example or standard we propose a protocol of our own which takes less computation time than Zhao's.

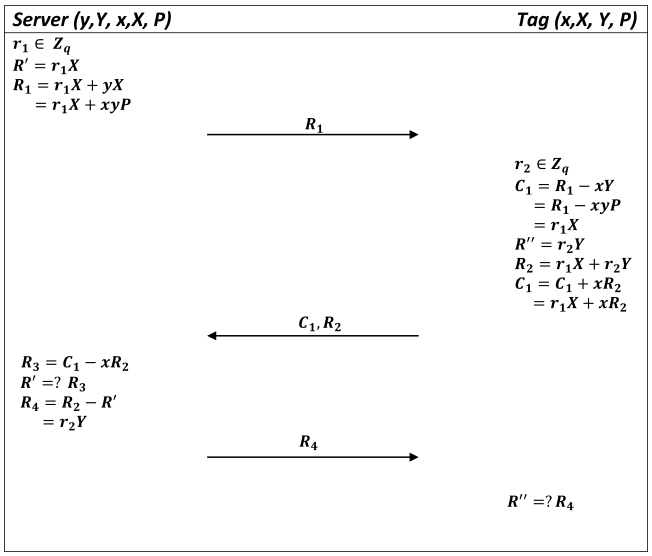

Figure 15: The Proposed Scheme

The public and private keys of the server and the tag is known to the server whereas, the tag does not have access to the server's private key.

The server computes two values $R^{'}$ and R_1. It sends R_1 to the tag.

Upon receiving, tag performs operation on R_1 which results in C_1, computes $R^{''}$ and R_2, updates C_1, sends C_1 and R_2 to the server.

The server performs operation on C_1 and checks whether the result matches with $R^{'}$. If it does not match the communication stops otherwise server performs operation on R_2 resulting in R_4 which is sent to the tag.

The tag, upon receiving the value checks whether it matches with $R^{''}$. If it matches then the communication process continues, that is both sides have been authenticated, otherwise no communication takes place.

3.1 Security Analysis of the proposed protocol

Mutual Authentication

An adversary cannot generate the legal message R_1 without the knowledge of r_1 and y where r_1 is a random number generated by the server and y is the private key of the server. Both of these are only know to the server and not directly passed during the communication which allows these two values to remain a secret, only to be known by the server.

An adversary cannot generate the legal message R_2 without the knowledge of r_2 and x where r_2 is a random number generated by the tag and x is the private key of the tag.

Thus the proposed protocol could provide Mutual Authentication between the tag and the server.

Anonymity

The private key of the tag x is known by the server and the tag itself. This value is not passed directly in any step of the authentication process and neither can it be retrieved by any other methods. Therefore, an adversary cannot get the private key: x of the tag and the proposed protocol could provide anonymity.

Availability

From the description of the proposed protocol, we know that no synchronously update of the secret key is needed in the execution of the protocol. Therefore, the proposed protocol could be executed between the server and the tag. Therefore, the proposed protocol could provide availability.

Forward Security Suppose that the adversary could get the tag's private key x. However, he cannot determine whether the messages R_1, R_2 are transmitted between the tag and the server since he does not know the random numbers r_1 and r_2. Therefore, the adversary cannot trace the tag and the proposed protocol could provide forward security.

Replay Attack

Suppose that the adversary intercepts the message R_1 and replays it to the tag. However, he cannot generate R_4 upon receiving the message C_1, R_2 since the tag generates a new random number r_2 and he does not know the tag's private key x. Then the tag could find the attack by checking the correctness of R_4. From the same method, we could show the server could find the replay attack by checking the correctness of R_3.Therefore,the proposed protocol could withstand replay attack.

Impersonation Attack

To impersonate the tag to the server, the adversary has to generate a legal message C_1, R_1. However, the adversary cannot generate C_1 and R_1 since he does not know r_2 and x. Therefore, the proposed protocol could withstand the impersonation attack.

Server Spoofing Attack

To impersonate the server to the tag, the adversary could generate a random number r_1, compute $R^{'}$ but he cannot generate R_2 as he does not know the server's private key y, neither does he know the tag's private key x. Therefore, the adversary cannot impersonate the server to the tag and the proposed protocol could withstand server spoofing attack.

DoS Attack

In the proposed scheme, we know that there is no need to synchronously update the tag's private key x after the scheme is executed since the tag's private key is well protected. Therefore, the proposed scheme could overcome DoS attack.

Location Tracking Attack

Suppose that the adversary could get the tag's private key x and intercept the messages . The adversary does not have the capabilities to obtain the server's private key y and two random values r_1 and r_2, so it cannot confirm whether those messages are transmitted between the server and the tag. Therefore, the proposed scheme could overcome the location tracking attack.

Cloning Attack

According to the description of the proposed protocol, we know that each tag has its own private key x. Suppose the adversary could get the private keys of several tags. However, he cannot get the private key of another tag since there is no relation among tags' private keys. Therefore, the proposed protocol could withstand cloning attack.

4 Conclusion

RFID tags can be attached to cash, clothing, and possessions, or implanted in animals and people, the possibility of reading personally-linked information without consent has raised serious privacy concerns. Because of that many RFID authentication protocols have been proposed to ensure security and privacy. In this paper we talked about the weaknesses and strengths of some of the authentication protocols based on ECC.

References

[1] P. Tuyls and L. Batina, "Rfid-tags for anti-counterfeiting," in Topics in cryptology–CT-RSA 2006. Springer, 2006, pp. 115–131.

[2] Y. K. Lee, L. Batina, and I. Verbauwhede, "Ec-rac (ecdlp based randomized access control): Provably secure rfid authentication protocol," in RFID, 2008 IEEE International Conference on. IEEE, 2008, pp. 97–104.

[3] 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, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on. IEEE, 2007, pp. 217– 222.

[4] 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, 2007. PerCom Workshops' 07. Fifth Annual IEEE International Conference on. IEEE, 2007, pp. 217– 222.

[5] M. S. Farash, "Cryptanalysis and improvement of an efficient mutual authentication rfid scheme based on elliptic curve cryptography," The Journal of Supercomputing, vol. 70, no. 2, pp. 987–1001, 2014.

[6] Z. Zhang and Q. Qi, "An efficient rfid authentication protocol to enhance patient medication safety using elliptic curve cryptography," Journal of medical systems, vol. 38, no. 5, pp. 1–7, 2014.

[7] Y.-P. Liao and C.-M. Hsiao, "A secure ecc-based rfid authentication scheme integrated with id-verifier transfer protocol," Ad Hoc Networks, vol. 18, pp. 133–146, 2014.

[8] Z. Zhao, "A secure rfid authentication protocol for healthcare environments using elliptic curve cryptosystem," Journal of medical systems, vol. 38, no. 5, pp. 1–7, 2014.

[9] D. He, N. Kumar, N. Chilamkurti, and J.-H. Lee, "Lightweight ecc based rfid authentication integrated with an id verifier transfer protocol," Journal of medical systems, vol. 38, no. 10, pp. 1–6, 2014.

[10] D. He, N. Kumar, N. Chilamkurti, and J.-H. Lee, "Lightweight ecc based rfid authentication integrated with an id verifier transfer protocol," Journal of medical systems, vol. 38, no. 10, pp. 1–6, 2014.