sensors

MDPI

Article

# A Lightweight RFID Mutual Authentication Protocol Based on Physical Unclonable Function

**He Xu [1,2,*], Jie Ding [1,2], Peng Li [1,2], Feng Zhu [1,2] and Ruchuan Wang [1,2]**

[1] School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210023, China; 1016041216@njupt.edu.cn (J.D.); lipeng@njupt.edu.cn (P.L.); zhufeng@njupt.edu.cn (F.Z.); wangrc@njupt.edu.cn (R.W.)

[2] Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China

* Correspondence: xuhe@njupt.edu.cn; Tel.: +86-25-8586-6354

Received: 9 January 2018; Accepted: 22 February 2018; Published: 2 March 2018

**Abstract:** With the fast development of the Internet of Things, Radio Frequency Identification (RFID) has been widely applied into many areas. Nevertheless, security problems of the RFID technology are also gradually exposed, when it provides life convenience. In particular, the appearance of a large number of fake and counterfeit goods has caused massive loss for both producers and customers, for which the clone tag is a serious security threat. If attackers acquire the complete information of a tag, they can then obtain the unique identifier of the tag by some technological means. In general, because there is no extra identifier of a tag, it is difficult to distinguish an original tag and its clone one. Once the legal tag data is obtained, attackers can be able to clone this tag. Therefore, this paper shows an efficient RFID mutual verification protocol. This protocol is based on the Physical Unclonable Function (PUF) and the lightweight cryptography to achieve efficient verification of a single tag. The protocol includes three process: tag recognition, mutual verification and update. The tag recognition is that the reader recognizes the tag; mutual verification is that the reader and tag mutually verify the authenticity of each other; update is supposed to maintain the latest secret key for the following verification. Analysis results show that this protocol has a good balance between performance and security.

**Keywords:** RFID technology; Physical Unclonable Function; lightweight cryptography; mutual verification

## 1. Introduction

Radio frequency identification technology is able to recognize objects and people automatically, and it can also automatically obtain related data of recognized objects, which is the non-contract recognition technique [1]. Because of this, the recognition of radio frequency identification (RFID) does not only need the artificial interference, but also work well in the severe environment. Recently, most RFID systems are based on the electric induction [2]. Attaching a RFID tag on an object, which involves the information of this object, the dedicated recognition terminal can recognize this attached object through reading the tag. In addition, since RFID products read data does not require light source and can pass through external material, and the service life is durable, compared with bar code, RFID products have more advantages [3].

By now, RFID has been used into many areas, such as supply chain management, electric passport, credit card, driving license [4,5], vehicle system (charging system, keyless entry systems), entrance guard card (building gate, public transport) and health care. Especially, in the USA, Japan, and other developed countries, they have equipped with advanced and mature RFID systems [6]. Some retailers have invested RFID technology, and also authorized RFID producers to attach tag on their goods, so that the low-budget RFID tags are pervasively produced. Wal-Mart passed a resolution, which producers must sufficiently take advantage of the RFID, attaching RFID tags on all products to reduce manpower and material resources [7].

Generally, a typical RFID framework is composed of a reader, tag and a database [8], which is shown in Figure 1.

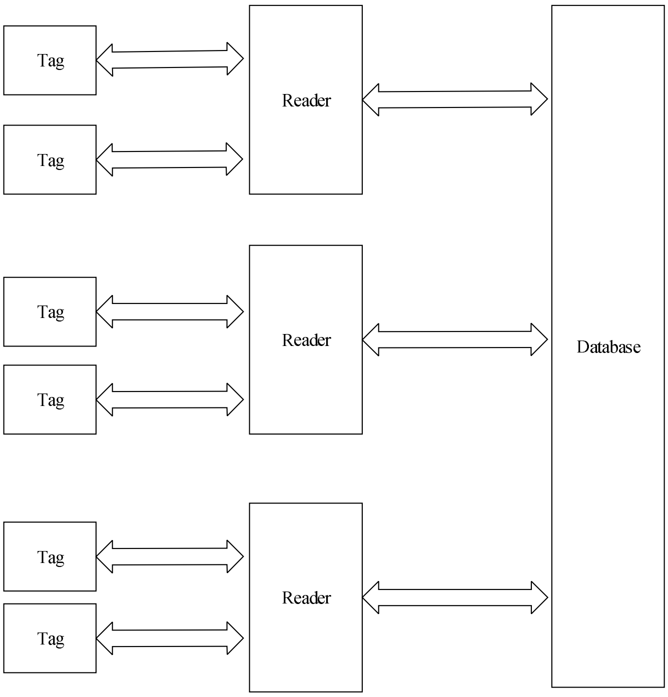


**Figure 1.** Radio frequency identification (RFID) system framework.

(1) Reader: The main function is to transfer energy to the tag via radio frequency and read data of tag or write data to tag [9]. In common RFID systems, reader still needs to exchange data with database. A reader is composed of the oscillating circuit, communication channel, and controller.

(2) Tag: According to the self-contained power or not, tag is classified into active tag, semi-passive tag and passive tag; based on the frequency, tag is classified into low-frequency tag, high-frequency tag and ultrahigh frequency tag [10]. By various applications, the proper tags are needed to be chosen.

(3) Database: It stores all information of tags which indicate all objects.

Mechanism of RFID systems: Firstly, reader sends signals via antenna, and tag receives signal and sends internal tag data. Then, reader receives and verifies the tag data. Finally, reader sends verification result to the host computer which is connected to a database.

Because RFID has the advantages of supporting dynamic real-time communication, fast recognition, easy to read, low cost, it is widely applied into various areas. Nevertheless, when RFID brings convenience, at the same time, its security problem is gradually exposed. In detail, during a simple write and read process, especially for the read or write on a passive tag, the read or write operation will happen when tag is close to a reader with the information exchange. Later, with the existence of a large number of fake and counterfeit RFID products, although researchers continuously find anti-fake measures, it is unable to completely eradicate them. Fake and counterfeit products make massive loss for world politics, economy, and culture, it has been a worldwide problem [11]. Clone tag is a kind of serious security threat for RFID systems. If attackers acquire the complete information of a tag, they could be able to obtain the unique identifier. In addition of this identifier, tag has no other identify, clone tag is therefore difficult to distinguish from the real one. In addition, as long as the legal tag data is obtained, attacker can launch such copy or clone attack.

In this case, researchers have taken measures to defend against clone attack, such as tag deactivation, encryption algorithms [12], identify verification [13] and hash code [14]. Reference [15] showed a mutual RFID verification protocol by using the elliptic curve cryptograph (ECC), which depends on ECC to enhance the ability of anti-clone. In the literature [16], authors gave a tag encryption algorithm without storage of the secret key. Instead, it depends on the hash value stored in the host computer. Reference [17] gave another RFID verification protocol based on hash, which provides already modified identifier to increase the privacy of verification. These mentioned protocols above rely on the security code which involve tag and the host computer, as well as one-way hash function.

Since the small and limited storage space in the tag, those solutions are based on the complex encryption algorithm which is similar to the hash function and it cannot be applied into the low-cost tag [18]. In such low-cost RFID systems, the biggest challenge to guarantee the security and privacy comes from the sparse tag resource [19]. To solve this problem, some lightweight encryption algorithms without hash function are put forward, and the method to identify clone tag as well [20,21]. Lehtonen raised a method based on statistics to detect clone tag from the RFID track [22]. Zannetti showed a method to detect clone attack with the protection of privacy [23].

Another anti-clone tag method is proposed by using the PUF technique. PUF technique is a breakthrough of the semi-conductor security techniques. PUF is a "biometric" identification technique in chips, which is also called the "chip DNA" technique [24]. PUF obtains the unique secrets from each chip. This secret information can be used to verify the authenticity of a chip, in the safe-guarding and anti-counterfeiting area, PUF has great application perspective. In the production of each chip, the tiny difference among chips unavoidably exist, although the chip design and production are the same, it is impossible to produce two chips which are completely same, so that PUF techniques can enhance anti-clone function [25]. The PUF judge circuit is shown as Figure 2 [26]. In Figure 2, the circuit is a decision-based PUF delay circuit. After inputting X[0], the circuit generates two delay paths whose length are equal, and the input signal is sent to the two paths at the same time. The decision module will choose the path that the input signal first reaches the destination. If the signal which is connected to the D point arrives firstly, it will output 1, otherwise it will output 0. Different physical manufacturing processes result in different outcomes which will give rather different outputs to the same input. PUF circuit receives a search command as an input signal and produces a unique serial feedback, which is verified by command/response mechanism. Different PUF circuits have various delay features, so that their transport speed is different and time of two signals passing PUF is different. The tail of a PUF circuit is an arbiter, and is able to judge the statue of signals before and after to output "1" or "0". A same signal passing two different PUF circuits can produce different outputs, and because that input signal determines the transport rout in PUF circuit, the different input signals are corresponding to different outputs.

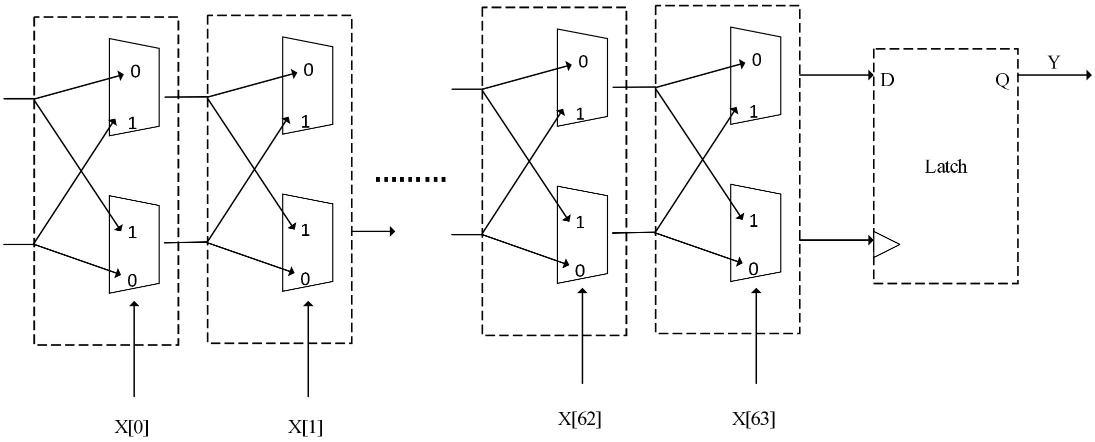


**Figure 2.** PUF subgrade circuit diagram.

In 2010, Kulseng proposed a lightweight mutual verification protocol based on PUF [27]. This lightweight protocol uses the PUF and linear feedback shift register (LFSR), rather than uses the complex plus and minus operations, which is very suitable for low-cost RFID tag, but it indicates the security problems of lightweight mutual verification [28].

This paper is structured as follows: section I introduces related research results about RFID security. Section 2 describes Kulseng's lightweight mutual verification protocol and the shortcomings of this protocol. Section 3 introduces our lightweight mutual verification protocol. Section 4 uses GNY logic analysis to prove the availability of our protocol. Section 5 executes the security analysis on our protocol. Section 6 simulates our lightweight mutual verification protocol and evaluate it in the experiments. Section 7 gives the performance analysis. At last, we conclude the paper.

## 2. Kulseng's Mutual Verification Protocol and Its Security Analysis

This section specifically analyses the Kulseng's mutual verification protocol based on the PUF. In this paper, it is called K protocol in the following. Some definitions of K protocol are shown in Table 1.

**Table 1.** The definitions of K protocol.

| Symbol | Definition |
|---|---|
| ID | The ID of tag |
| FID | The false ID of tag |
| IDS | The index of tag in the database |
| PUF(X) | The result of the value X processed by the PUF module in the tag |
| $\oplus$ | The XOR operation |
| F(X) | The Permutation function |
| $K_n$ | The shared secret key by the reader and the tag |
| $P_n$ | The secret value of the tag |
| $<<<$ | The loop left movement operation |
| & | The AND operation |
| $\|\,\|$ | The JOIN operation |

### 2.1. Protocol Process

Figure 3 shows the Kulseng's mutual verification protocol. The specific steps are listed in the following.

S1: Reader firstly sends a search request to the tag.

S2: After receiving the request, the tag sends the IDS to reader.

S3: After receiving the IDS from the tag, reader searches this IDS in database. If this IDS can be found, it means that the database stores this IDS, and the tag is successfully recognized and reader sends ID $\oplus$ $G_n$ to the tag. If not, this tag can not be recognised and the protocol stops.

S4: After receiving ID $\oplus$ $G_n$, by calculating ID $\oplus$ $G_n$ $\oplus$ ID to obtain $G_n$ of ID $\oplus$ $G_n$ sent from reader, and tag compares this with its own $G_n$. If they are different, it means that the reader is not been verified, and the protocol stops. Otherwise, this reader is verified. The tag executes following operations:

① Calculating following value:

$$G_{n+1} = \mathrm{PUF}(G_n),\ G_{n+2} = \mathrm{PUF}(G_{n+1}),\ K_n = F(G_n),\ K_{n+1} = F(K_n)$$

② Calculating the following value and sending them to the reader.

$$K_n \oplus G_{n+1},\ K_{n+1} \oplus G_{n+2}$$

③ Update tag data:

$$\mathrm{IDS} = F(\mathrm{IDS} \oplus G_n),\ G_n = G_{n+1}$$

S5: After receiving $K_n \oplus G_{n+1}$, $K_{n+1} \oplus G_{n+2}$, reader judges whether $K_n \oplus G_{n+1} = F(G_n) \oplus G_{n+1}$ or not. If not, this tag is not verified, and the K protocol stops; if it is equal, tag is verified and reader executes the following operations:

① Calculating value of $G_{n+2}$:

$$K_{n+1} \oplus G_{n+2} \oplus F(F(G_n))$$

② Updating reader data by the calculated value of $G_{n+2}$:

$$\mathrm{IDS} = F(\mathrm{IDS} \oplus G_n),\ G_n = G_{n+1},\ G_{n+1} = G_{n+2}$$

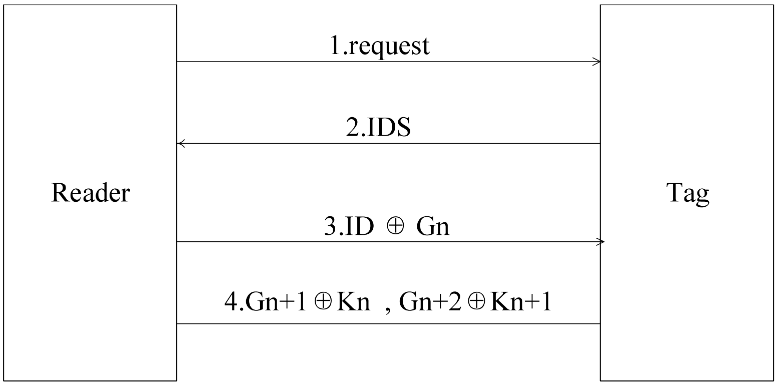


**Figure 3.** Kulseng's mutual verification protocol.

## 2.2. Security Analysis of K protocol

### 2.2.1. Data Confidentiality

Attacker firstly intercepts a complete verification process to obtain $\mathrm{ID} \oplus G_n$, $K_n \oplus G_{n+1}$ and $K_{n+1} \oplus G_{n+2}$. Tag updates its data after a complete verification, and $\mathrm{IDS} = F(\mathrm{IDS} \oplus G_n)$ is updated at the same time. In the next verification, attacker pretends to be a reader and sends a request to tag, and the tag sends its IDS to attacker after receiving the request from attacker, in which the $\mathrm{IDS} = F(\mathrm{IDS} \oplus G_n)$. The attacker has four values: $\mathrm{ID} \oplus G_n$, $K_n \oplus G_{n+1}$, $K_{n+1} \oplus G_{n+2}$ and $(\mathrm{IDS} \oplus G_n)$. Here, F is the function representing Linear Feedback Shifting Register (LFSR). According to the LFSR, if the attacker knows the LFSR's characteristic polynomial and output, the attacker can know the seed of LFSR via the matrix multiplication. In this way, combining the IDS in last verification and $F(\mathrm{IDS} \oplus G_n)$ obtained at this time, attacker can easily know the secret key $G_n$ used in the last verification, and use XOR between $G_n$ and $\mathrm{ID} \oplus G_n$, then attacker can know the ID of tag.

### 2.2.2. Desynchronized Attack

Attack one: Intercept the last information sent by tag to reader ( $K_n \oplus G_{n+1}$, $K_{n+1} \oplus G_{n+2}$).

Assume attacker is intercepting a verification process, after the tag confirms the reader, tag updates its data and sends $K_n \oplus G_{n+1}$, $K_{n+1} \oplus G_{n+2}$ to the reader. At this moment, the attacker can obtain this information without being received by the reader. Because reader cannot verify the tag, the database cannot update the corresponding secret key value to the tag, which causes the data of the tag and the database are not synchronized. In the following verification, due to the difference between $G_n$ of the tag and $G_n$ of the datab $K_n$ ase, tag would refuse service.

Attack two: Modify $K_{n+1} \oplus G_{n+2}$ of $(\oplus G_{n+1}$, $K_{n+1} \oplus G_{n+2})$ sent from tag to reader.

Assume an attacker is intercepting a verification process, after the tag confirms the reader, tag updates its own data and sends $K_n \oplus G_{n+1}$, $K_{n+1} \oplus G_{n+2}$ to the reader. Then, attacker modifies $K_{n+1} \oplus G_{n+2}$ of the information sent to the reader, such as modifying into ( $K_{n+1} \oplus G_{n+2} \oplus r_1$), which is finally sent to the reader. After reader receives $K_n \oplus G_{n+1}$, $K_{n+1} \oplus G_{n+2} \oplus r_1$, reader judges whether $K_n \oplus G_{n+1} = F(G_n) \oplus G_{n+1}$ or not to verify the tag. Because the attacker does not modify $K_n \oplus G_{n+1}$, reader successfully verifies the tag. After this, reader uses $K_{n+1} \oplus G_{n+2} \oplus r_1$ to execute the following operations as usual.

① Calculating $G_{n+2}$:

$$K_{n+1} \oplus G_{n+2} \oplus r_1 \oplus F(F(G_n))$$

Now, $G_{n+2}$ is updated by $G_{n+2} = G_{n+2} \oplus r_1$.

② Using $G_{n+2}$ to update reader information:

$$IDS = F(IDS \oplus G_n),\ G_n = G_{n+1},\ G_{n+1} = G_{n+2} \oplus r_1$$

This causes the desynchronization between tag data and database. In the following verification, due to the difference between $G_{n+1}$ of the database and $G_{n+1}$ of the tag, verification for the tag is failed.

## 3. Proposed Protocol

Due to the shortcoming of Kulseng's lightweight mutual verification protocol that is based on the PUF, we will introduce our proposed protocol in this section. Some definitions of proposed protocol are shown in Table 1 at the previous section.

In this protocol, RFID tag is embedded with the PUF module, which makes each RFID tag to produce a unique secret key based on its own circuit, so as to defend against the clone attack. Initially, each RFID tag stores three values: ① $P_n = PUF(challenge)$, ②FID and ③ $K_n$. The first value is the dedicated secret value of the tag; the second value is the IDS used during the mutual verification between the tag and the host computer; the third one is the shared secret key by the reader and the tag. After each verification, the secret key $P_n$, shared secret key $K_n$ and FID would be updated. For each tag, the database stores two set of values: $\left\{FID^{old},\ P_n^{old},\ P_{n+1}^{old},\ ID,\ K_n^{old}\right\}$ and $\{FID^{new}, P_n^{new}, P_{n+1}^{new}, K_n^{new}\}$. The later value set is the current tag value, and the prior value set is the tag value of the last communication. This verification protocol contains three parts: tag recognition, mutual verification and update. Tag recognition is to verify the tag's authenticity; mutual verification is to use the verified tag to verify reader's authenticity; update is to store the latest secret key for the following verification. Figure 4 shows the certification process of proposed protocol.

In this paper, the backstage database cannot meet the requirements because it should store all the tag information resulting in a very large amount of data. In order to solve the above problems, the HBase of big data technology is used to store the data. HBase which is a distributed and column-oriented is an open source database. A table in HBase can have hundreds of millions of rows and millions of columns, and the data in each unit can have multiple versions for backup.

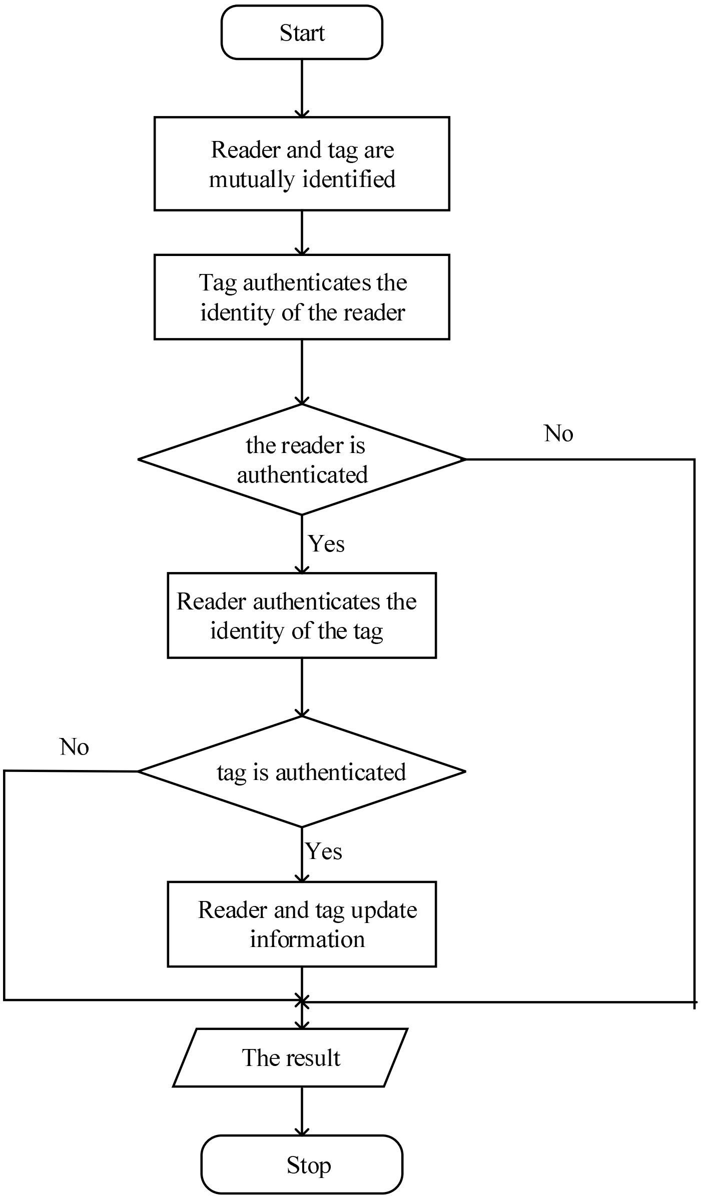

Start
Reader and tag are mutually identified
Tag authenticates the identity of the reader
the reader is authenticated
No
Yes
Reader authenticates the identity of the tag
tag is authenticated
No
Yes
Reader and tag update information
The result
Stop

**Figure 4.** Single tag certification process.

### 3.1. Tag Verification

In the tag verification process, reader firstly sends a search request to tag, after receiving the request, tag produces a random number $r_1$, and sends $r_1$ with FID to the reader. After receiving the random number $r_1$ and FID, reader uses FID to search data in the host database. If $\mathrm{FID} = \mathrm{FID}^{\mathrm{new}}$, it means that the database contains this FID, and this tag can be verified without being attacked previously. In addition, the reader uses $\left\{\mathrm{FID}^{\mathrm{new}}, \mathrm{P}_{\mathrm{n}}^{\mathrm{new}}, \mathrm{P}_{\mathrm{n+1}}^{\mathrm{new}}, \mathrm{K}_{\mathrm{n}}^{\mathrm{new}}\right\}$ to execute mutual verification with the tag. However, in the previous verification, tag might be intercepted, resulting in the update of the database, while the tag data is not updated. In this case, reader uses $\left\{\mathrm{FID}^{\mathrm{old}}, \mathrm{P}_{\mathrm{n}}^{\mathrm{old}}, \mathrm{P}_{\mathrm{n+1}}^{\mathrm{old}}, \mathrm{ID}, \mathrm{K}_{\mathrm{n}}^{\mathrm{old}}\right\}$ to execute the mutual verification with this tag. If both of the new and old FID cannot match with the FID of the reader, this protocol stops.

### *3.2. Mutual Verification*

The purposes of Mutual verification is to verify the identity of both tag and reader. Reader firstly produces a random number $r_2$, combining FID, secret key $P_{n+1}$, shared key $K_n$ and random number $r_1$ according to the Equations (1) and (2) to calculate A and B. Then, reader sends A | | B to the RFID tag.

$$A = FID \oplus P_{n+1} \oplus K_n \oplus r_1 \oplus r_2 \tag{1}$$

$$B = (r_1 <<< 8)\ \&\ (r_2 <<< 1) \oplus (P_{n+1} <<< 2) \tag{2}$$

After receiving A | | B, tag calculates D and E by Equations (3) and (4), Equation (5) deduces the random number $r_2'$ of the reader and further uses $r_2'$ to obtain $B'$ by Equation (6).

$$D = PUF(P_n) \tag{3}$$

$$E = PUF(D) \tag{4}$$

$$r_2' = A \oplus FID \oplus D \oplus K_n \oplus r_1 \tag{5}$$

$$B' = (r_1 <<< 8)\ \&\ \left(r_2' <<< 1\right) \oplus (D <<< 2) \tag{6}$$

Then, tag compares whether $B$ and $B'$ are equal or not. If not, the reader is fake, and verification terminates; otherwise, the reader passes the verification, and the protocol goes on for the next step. At the same time, with previously calculated D, E and deduced random number $r_2'$, by Equation (7) F is calculated, which is also used to deduce the random number $r_2'$, F and FID of the tag. By Equation (8). H is calculated and F | | H is sent to reader.

$$F = D \oplus E \oplus r_2' \tag{7}$$

$$H = (FID <<< 8)\ \&\ (F <<< 1) \oplus \left(r_2' <<< 2\right) \tag{8}$$

After receiving F | | H, by Equation (9), reader deduces secret key $E'$ of the tag, and this deduced key is used to obtain $E'$ random number $r_2$ and secret key $P_{n+1}$. With Equation (10), $F'$ is calculated. Then, using the FID of this tag, $F'$, random number $r_2$, by Equation (11). $H'$ is calculated.

$$E' = F \oplus r_2 \oplus P_{n+1} \tag{9}$$

$$F' = E' \oplus r_2 \oplus P_{n+1} \tag{10}$$

$$H' = (FID <<< 8)\ \&\ \left(F' <<< 1\right) \oplus (r_2 <<< 2) \tag{11}$$

At this moment, reader compares whether H and $H'$ are equal or not. If it is equal, the tag has been verified before and the mutual verification terminates. If not, this tag is fake, and the verification stops.

### *3.3. Update Process*

After verifying the tag, update starts. During this process, firstly, the host database updates, then the tag updates. The process of updating the database is classified into two cases. In the tag update, if the database uses $FID^{oid}$ and FID to match, the host database does not need to update; if the database uses $FID^{new}$ and $FID$ of the tag to match, the database should be updated in following way:

$$P_n^{new} = P_{n+1}^{new},\ P_{n+1}^{new} = E'$$

$$K_n^{new} = (K_n^{new} <<< 8)\ \&\ \left(r_2' <<< 1\right) \oplus (r_1 <<< 2)$$

$$FID^{new} = (FID^{new} <<< 8)\ \&\ \left(r_2' <<< 1\right) \oplus (r_1 <<< 2)$$

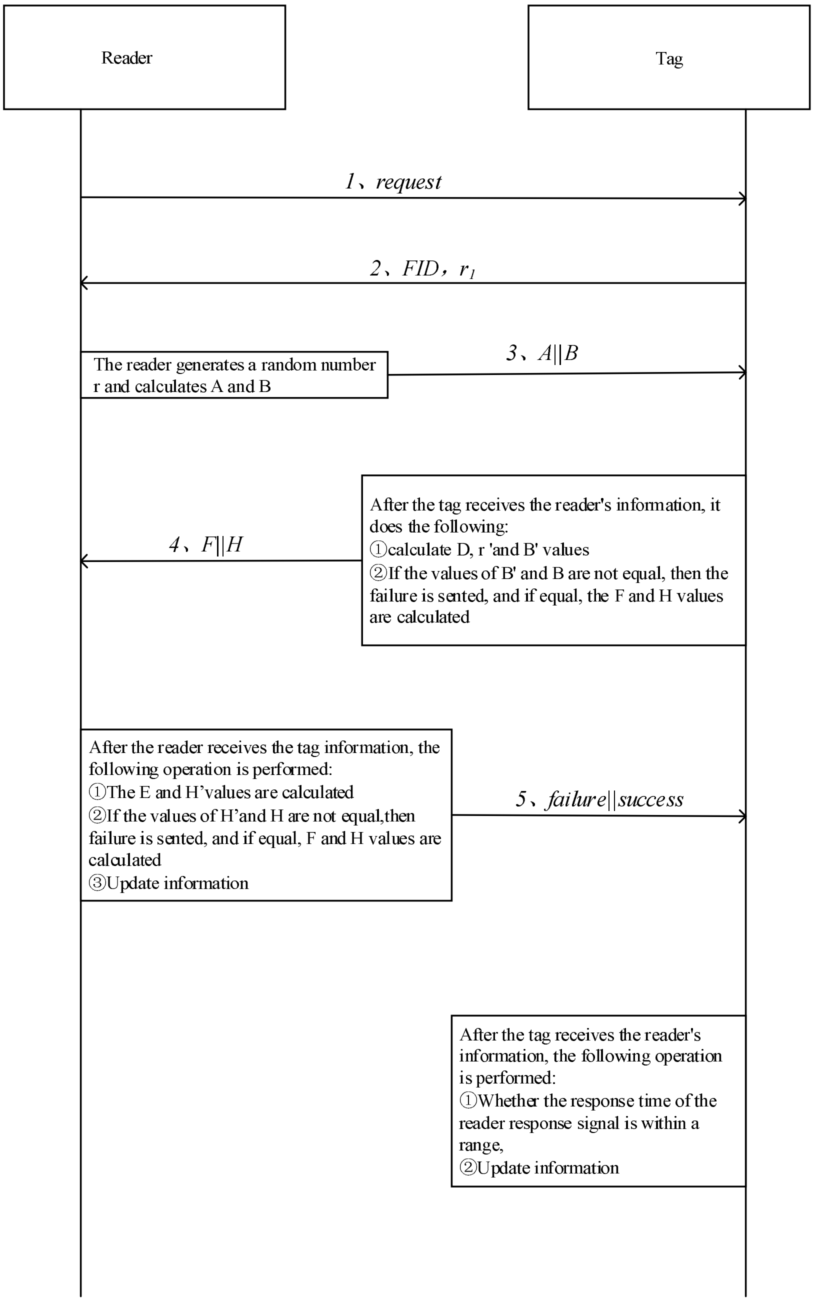
$$FID^{old} = FID^{new},\ P_n^{old} = P_n^{new},\ P_{n+1}^{old} = P_{n+1}^{new},\ K_n^{old} = K_n^{new}$$

If reader sends success command to the tag, and the processing time of the reader is within the acceptable range, the tag would be updated in the following ways, which is shown in Figure 5.

$$\mathrm{P_n} = \mathrm{D}$$

$$\mathrm{FID} = (\mathrm{FID} <<< 8)\ \&\ \left(\mathrm{r}_2' <<< 1\right)\ \oplus (\mathrm{r}_1 <<< 2)$$

$$\mathrm{K_n} = (\mathrm{K_n} <<< 8)\ \&\ \left(\mathrm{r}_2' <<< 1\right)\ \oplus\ (\mathrm{r}_1 <<< 2)$$



The reader generates a random number r and calculates A and B

After the tag receives the reader's information, it does the following:
①calculate D, r 'and B' values
②If the values of B' and B are not equal, then the failure is sented, and if equal, the F and H values are calculated

After the reader receives the tag information, the following operation is performed:
①The E and H'values are calculated
②If the values of H'and H are not equal,then failure is sented, and if equal, F and H values are calculated
③Update information

After the tag receives the reader's information, the following operation is performed:
①Whether the response time of the reader response signal is within a range,
②Update information

**Figure 5.** Time sequence diagram of single tag authentication.

## 4. Proof of Security

In this section, we use GNY logic [29] to prove our protocol. The GNY logic was proposed by Gong, Needham and Yahalom in 1990. It is a logic that analyzes the authentication protocol. The GNY logic uses a completely different state search tool, which includes a collection of beliefs maintained by each subject and a collection of inference rules that get new beliefs from the old beliefs. BAN logic has a very simple, intuitive set of rules, so it is easy to use. As it is pointed out in the reference [29], GNY logic can be used to find serious errors in the protocol, which has attracted widely attention by security researchers. The application of GNY logic has epoch-making significance. It greatly promoted the development of formal verification of security protocols, and inspired many methods of formal verification of security protocols.

### 4.1. GNY Logic Expression

$A \mid \equiv B$: A trusts message B.

$A \mid \nabla B$: A receives message B.

$A \mid \sim B$: A sends message B.

$A \ni B$: A has message B.

(B, C): connecting message B and message C.

#(B): message B is the latest, which means that message B has never been sent before.

$\{B\}_k$: message B is encrypted by secret key k.

$A \overset{k}{\leftrightarrow} D$: A and D share secret key k to communicate, which means only A and D or the trusted third party know the secret key k.

### 4.2. Principle of the GNY Deduction

Principle 1: $\frac{A|\equiv\#(B)}{A|\equiv\#(B,C),A|\equiv\#(F(B))}$

Principle 1 indicates: if A trusts that message B is the latest, then it can be deduced that A also trusts the connection between B and C is the latest, and the mapping is trusted as the latest by A whose image is B.

Principle 2: $\frac{A\ni B,A\ni C}{A\ni(B,C),A\ni(F(B,C))}$

Principle 2 indicates: If A has message B and C, then it can be deduced that A not only has the connection between B and C, but also the mapping whose image is message B and message C.

Principle 3: $\frac{A\ni B,A|\equiv\#(B)}{A|\equiv\#\{H(B)\}}$

Principle 3 indicates: If A not only has message B, but also trusts that B is the latest message, then, it can be deduced that A also trusts that the equation containing B is the latest.

Principle 4: $\frac{A|\nabla(B)}{A\ni B}$

Principle 4 indicates: If A received message B, then A has message B can be deduced.

Principle 5: $\frac{A|\equiv D\overset{k}{\leftrightarrow}A,\ A|\nabla H(B,\langle K\rangle)\ ,\ A\ni(B,K),\ A|\equiv\#(B,K)}{A|\equiv D|\sim(B,\langle K\rangle),A|\equiv D|\sim H(B,\langle K\rangle)}$

Principle 5 indicates: A trusts that the secret key K is shared by A and D, and A previously received encrypted B by secret key K. A has the connection between message B and secret key K. A trusts that the connection between message B and secret key K is the latest, then, A trusts that D previously sent the connection between message B and the secret key K, and A also trusts that D used to send encrypted B by K.

### 4.3. Protocol Process

Protocol (1): R→T: request

Protocol (2): T→R: FID

Protocol (3): $R\rightarrow T{:}FID \oplus P_{n+1} \oplus K_n \oplus r_1 \oplus r_2||((r_1 <<< 8)\ \&\ (r_2 <<< 1) \oplus (P_{n+1} <<< 2))$

Protocol (4): $T \rightarrow R{:}\ (PUF(P_n) \oplus PUF(PUF(P_n)) \oplus r'_2,\ (FID <<< 8)\ \&\ (PUF(P_n) \oplus PUF(PUF(P_n)) \oplus r'_2 <<< 1) \oplus (r'_2 <<< 2))$

Protocol (5): R→T: success/failure

### 4.4. Specifically Describe the Protocol above by Using the GNY Logic Language

Protocol (1): $\mathrm{T} \mid \nabla$ request

Protocol (2): $\mathrm{R} \mid \nabla$ FID

Protocol (3): $\mathrm{T}|\nabla \mathrm{FID} \oplus \mathrm{P}_{\mathrm{n}+1} \oplus \mathrm{K}_{\mathrm{n}} \oplus \mathrm{r}_1 \oplus \mathrm{r}_2||((\mathrm{r}_1 <<< 8) \,\&\, (\mathrm{r}_2 <<< 1) \oplus (\mathrm{P}_{\mathrm{n}+1} <<< 2))$

Protocol (4): $\mathrm{R} \mid \nabla(\mathrm{PUF}(\mathrm{P}_{\mathrm{n}}) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_{\mathrm{n}})) \oplus \mathrm{r}'_2,\ (\mathrm{FID} <<< 8) \,\&\, (\mathrm{PUF}(\mathrm{P}_{\mathrm{n}}) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_{\mathrm{n}})) \oplus \mathrm{r}' <<< 1) \oplus (\mathrm{r}'_2 <<< 2))$

Protocol (5): $\mathrm{T} \mid \nabla$ success/failure

### 4.5. Assumption

(1) $\mathrm{T} \ni (\mathrm{FID}, \mathrm{P}_{\mathrm{n}}, \mathrm{K}_{\mathrm{n}}, \mathrm{ID})$

(2) $\mathrm{R} \ni \left(\mathrm{FID}^{\mathrm{new}}, \mathrm{P}^{\mathrm{new}}_{\mathrm{n}}, \mathrm{P}^{\mathrm{new}}_{\mathrm{n}+1}, \mathrm{K}^{\mathrm{new}}_{\mathrm{n}}, \mathrm{FID}^{\mathrm{old}}, \mathrm{P}^{\mathrm{old}}_{\mathrm{n}}, \mathrm{P}^{\mathrm{old}}_{\mathrm{n}+1}, \mathrm{ID}, \mathrm{K}^{\mathrm{old}}_{\mathrm{n}}\right)$

(3) $\mathrm{T} \ni (r_1)$

(4) $\mathrm{T}| \equiv \#(r_1)$

(5) $\mathrm{R} \ni (r_2)$

(6) $\mathrm{R}| \equiv \#(r_2)$

(7) $\mathrm{T}| \equiv \mathrm{R} \xleftrightarrow{\mathrm{P}_{\mathrm{n}},\, \mathrm{K}_{\mathrm{n}},\, \mathrm{P}_{\mathrm{n}+1}} \mathrm{T}$

(8) $\mathrm{R}| \equiv \mathrm{T} \xleftrightarrow{\mathrm{P}_{\mathrm{n}},\, \mathrm{K}_{\mathrm{n}},\, \mathrm{P}_{\mathrm{n}+1}} \mathrm{R}$

### 4.6. Target Formulas to Be Proven

(1) $\mathrm{T}|\equiv \mathrm{R}| \sim \#\{\mathrm{FID} \oplus \mathrm{P}_{\mathrm{n}+1} \oplus \mathrm{K}_{\mathrm{n}} \oplus \mathrm{r}_1 \oplus \mathrm{r}_2,\ (\mathrm{r}_1 <<< 8) \,\&\, (\mathrm{r}_2 <<< 1) \oplus (\mathrm{P}_{\mathrm{n}+1} <<< 2)\}$

(2) $\mathrm{R}|\equiv \mathrm{T}| \sim \#\{\mathrm{PUF}(\mathrm{P}_{\mathrm{n}}) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_{\mathrm{n}})) \oplus \mathrm{r}'_2,\ (\mathrm{FID} <<< 8) \,\&\, (\mathrm{PUF}(\mathrm{P}_{\mathrm{n}}) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_{\mathrm{n}})) \oplus \mathrm{r}'_2 <<< 1) \oplus (\mathrm{r}'_2 <<< 2)\}$

(1) Prove $\mathrm{T}|\equiv \mathrm{R}| \sim \#\{\mathrm{FID} \oplus \mathrm{P}_{\mathrm{n}+1} \oplus \mathrm{K}_{\mathrm{n}} \oplus \mathrm{r}_1 \oplus \mathrm{r}_2,\ (\mathrm{r}_1 <<< 8) \,\&\, (\mathrm{r}_2 <<< 1) \oplus (\mathrm{P}_{\mathrm{n}+1} <<< 2)\}$

S1: by principle: $\frac{\mathrm{A}|\equiv\#(\mathrm{B})}{\mathrm{A}|\equiv\#(\mathrm{B,C}),\mathrm{A}|\equiv\#(\mathrm{F(B)})}$ and assumption (4) $\mathrm{T}| \equiv \#(\mathrm{r}_1)$, we can deduce:

$$\mathrm{T}| \equiv \#\{\mathrm{FID} \oplus \mathrm{P}_{\mathrm{n}+1} \oplus \mathrm{K}_{\mathrm{n}} \oplus \mathrm{r}_1 \oplus \mathrm{r}_2\}$$

$$\mathrm{T}| \equiv \#\{(\mathrm{r}_1) \,\&\, (\mathrm{r}_2) \oplus (\mathrm{P}_{\mathrm{n}+1})\}$$

S2: by principle: $\frac{\mathrm{A}\ni\mathrm{B},\mathrm{A}\ni\mathrm{C}}{\mathrm{A}\ni(\mathrm{B,C}),\mathrm{A}\ni(\mathrm{F(B,C)})}$ and assumption (3) $\mathrm{T} \ni (\mathrm{r}_1)$, we can deduce:

$$\mathrm{T} \ni \{(\mathrm{r}_1) \,\&\, (\mathrm{r}_2) \oplus (\mathrm{P}_{\mathrm{n}+1})\}$$

S3: by principle: $\frac{\mathrm{A}\ni\mathrm{B},\mathrm{A}|\equiv\#(\mathrm{B})}{\mathrm{A}|\equiv\#\{\mathrm{H(B)}\}}$ and the above corollary, we can deduce:

$$\mathrm{T}| \equiv \#\{(\mathrm{r}_1 <<< 8) \,\&\, (\mathrm{r}_2 <<< 1) \oplus (\mathrm{P}_{\mathrm{n}+1} <<< 2)\}$$

S4: by principle: $\frac{\mathrm{A}|\nabla(\mathrm{B})}{\mathrm{A}\ni\mathrm{B}}$ and protocol (3), we can deduce:

$$\mathrm{T} \ni \{\mathrm{FID} \oplus \mathrm{P}_{\mathrm{n}+1} \oplus \mathrm{K}_{\mathrm{n}} \oplus \mathrm{r}_1 \oplus \mathrm{r}_2\}$$

$$\mathrm{T} \ni \{(\mathrm{r}_1 <<< 8) \,\&\, (\mathrm{r}_2 <<< 1) \oplus (\mathrm{P}_{\mathrm{n}+1} <<< 2)\}$$

S5: by principle: $\frac{\mathrm{A}|\equiv\mathrm{D}\overset{\mathrm{k}}{\leftrightarrow}\mathrm{A},\ \mathrm{A}|\nabla\mathrm{H}(\mathrm{B},\langle\mathrm{K}\rangle)\ ,\ \mathrm{A}\ni(\mathrm{B,K}),\ \mathrm{A}|\equiv\#(\mathrm{B,K})}{\mathrm{A}|\equiv\mathrm{D}|\sim(\mathrm{B},\langle\mathrm{K}\rangle),\mathrm{A}|\equiv\mathrm{D}|\sim\mathrm{H}(\mathrm{B},\langle\mathrm{K}\rangle)}$ and assumption (7) and (8), we can deduce:

$$\mathrm{T}|\equiv \mathrm{R}| \sim \{\mathrm{FID} \oplus \mathrm{P}_{\mathrm{n}+1} \oplus \mathrm{K}_{\mathrm{n}} \oplus \mathrm{r}_1 \oplus \mathrm{r}_2\}$$

$$\mathrm{T}|\equiv \mathrm{R}|\sim \{(\mathrm{r}_1 < < < 8)\ \&\ (\mathrm{r}_2 < < < 1) \oplus (\mathrm{P}_{n+1} < < < 2)\}$$

S6: then, there is:

$$\mathrm{T}|\equiv \mathrm{R}|\sim \#\{\mathrm{FID} \oplus \mathrm{P}_{n+1} \oplus \mathrm{K}_n \oplus \mathrm{r}_1 \oplus \mathrm{r}_2\}$$

$$\mathrm{T}|\equiv \mathrm{R}|\sim \#\{(\mathrm{r}_1 < < < 8)\ \&\ (\mathrm{r}_2 < < < 1) \oplus (\mathrm{P}_{n+1} < < < 2)\}$$

S7: by the definition, we can also deduce:

$$\mathrm{T}|\equiv \mathrm{R}|\sim \#\{\mathrm{FID} \oplus \mathrm{P}_{n+1} \oplus \mathrm{K}_n \oplus \mathrm{r}_1 \oplus \mathrm{r}_2,\ (\mathrm{r}_1 < < < 8)\ \&\ (\mathrm{r}_2 < < < 1) \oplus (\mathrm{P}_{n+1} < < < 2)\}$$

(2) prove
$\mathrm{R}|\equiv \mathrm{T}|\sim \#\{\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2,\ (\mathrm{FID} < < < 8)\ \&\ (\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2 < < < 1) \oplus (\mathrm{r}'_2 < < < 2)\}$

S1: by principle: $\frac{\mathrm{A}|\equiv\#(\mathrm{B})}{\mathrm{A}|\equiv\#(\mathrm{B,C}),\mathrm{A}|\equiv\#(\mathrm{F(B)})}$ and assumption (6) $\mathrm{R}|\equiv\#(\mathrm{r}_2)$, there is:

$$\mathrm{R}|\equiv\#\{\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2\}$$

$$\mathrm{R}|\equiv\#\{(\mathrm{FID})\ \&\ (\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2) \oplus (\mathrm{r}'_2)\}$$

S2: by principle: $\frac{\mathrm{A}\ni\mathrm{B},\mathrm{A}\ni\mathrm{C}}{\mathrm{A}|\ni(\mathrm{B,C}),\mathrm{A}|\ni(\mathrm{F(B,C)})}$ and assumption (5) $\mathrm{R}\ni(\mathrm{r}_2)$, there is:

$$\mathrm{R}\ni\{(\mathrm{FID})\ \&\ (\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2) \oplus (\mathrm{r}'_2)\}$$

S3: by principle: $\frac{\mathrm{A}\ni\mathrm{B},\mathrm{A}|\equiv\#(\mathrm{B})}{\mathrm{A}|\equiv\#\{\mathrm{H(B)}\}}$ and the corollary above, there is:

$$\mathrm{R}|\equiv\#\{(\mathrm{FID} < < < 8)\ \&\ (\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2 < < < 1) \oplus (\mathrm{r}'_2 < < < 2)\}$$

S4: by principle: $\frac{\mathrm{A}|\nabla(\mathrm{B})}{\mathrm{A}\ni\mathrm{B}}$ and protocol (4), there is:

$$\mathrm{R}\ni\{\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2\}$$

$$\mathrm{R}\ni\{(\mathrm{FID} < < < 8)\ \&\ (\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2 < < < 1) \oplus (\mathrm{r}'_2 < < < 2)\}$$

S5: by principle: $\frac{\mathrm{A}|\equiv\mathrm{D}\overset{k}{\leftrightarrow}\mathrm{A},\ \mathrm{A}|\nabla\mathrm{H}(\mathrm{B},\langle\mathrm{K}\rangle)\ ,\ \mathrm{A}\ni(\mathrm{B,K}),\ \mathrm{A}|\equiv\#(\mathrm{B,K})}{\mathrm{A}|\equiv\mathrm{D}|\sim(\mathrm{B},\langle\mathrm{K}\rangle),\mathrm{A}|\equiv\mathrm{D}|\sim\mathrm{H}(\mathrm{B},\langle\mathrm{K}\rangle)}$ and assumption (7) and (8), we can deduce:

$$\mathrm{R}|\equiv \mathrm{T}|\sim\{\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2\}$$

$$\mathrm{R}|\equiv \mathrm{T}|\sim\{(\mathrm{FID} < < < 8)\ \&\ (\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2 < < < 1) \oplus (\mathrm{r}'_2 < < < 2)\}$$

S6: according to the definition, we can deduce:

$$\mathrm{R}|\equiv \mathrm{T}|\sim\#\{\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2\}$$

$$\mathrm{R}|\equiv \mathrm{T}|\sim\#\{(\mathrm{FID} < < < 8)\ \&\ (\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2 < < < 1) \oplus (\mathrm{r}'_2 < < < 2)\}$$

S7: by definition, there is:

$$\mathrm{R}|\equiv \mathrm{T}|\sim\#\{\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2,\ (\mathrm{FID} < < < 8)\ \&\ (\mathrm{PUF}(\mathrm{P}_n) \oplus \mathrm{PUF}(\mathrm{PUF}(\mathrm{P}_n)) \oplus \mathrm{r}'_2 < < < 1) \oplus (\mathrm{r}'_2 < < < 2)\}$$

## 5. Security Analysis

This section analyses the possible attacks and the security of this protocol.

### 5.1. Mutual Verification

Mutual verification: Reader verifies the tag, and tag reversely verifies the reader. For some verification protocols, they just verify the tag, without the verification of the reader, which results in a security issue. If attacker uses a reader which is not been verified by a tag, the RFID system data would be leaked, or even the system would be damaged irreparably, such as desynchronization attack and DDos attack.

(1) Reader forgery

The third step of the proposed protocol is to send message A||B to RFID tag, according to the Equations (3) and (4), tag calculates the values of D and E, and tag deduces the random number $r_2'$ of the reader with Equation (5), where deduced $r_2'$ is also used to generate $B'$ by Equation (6). Then, $B'$and B of the reader are compared to verify the reader, in which value of A and value of B are calculated by the random number $r_1$ of the tag, random number $r_2$ of the reader, FID of the tag and the shared secret key $P_{n+1}$ and secret key $K_n$. These values would be updated after each round to prepare for the next round, and this is because the original values should be secreted so that the attacker cannot produce a fake reader.

(2) Tag forgery

The fourth step of our protocol is to send message F||H to the reader, reader deduces the secret key of the tag by Equation (9), and uses deduced secret key $E'$, its random number $r_2$, secret key $P_{n+1}$ and Equation (10) to calculate $F'$. Then reader uses FID of the tag, $F'$, and random number $r_2'$ to produce $H'$ by Equation (9), by which reader compares H and the calculated $H'$ to complete the verification of the reader. In this process, F and H are calculated by the deduced random number $r_2$ by the tag, the secret key produced by PUF and the tag signature FID. After each round verification, these values would be updated for the following verification. As long as the initial values are in secret, attacker cannot produce a fake tag.

Above all, this protocol can guarantee the mutual verification between reader and tag.

### 5.2. Data Confidentiality and Tag Anonymity

During the verification process, the internal data of either tag or reader should be transferred in cipher text, and the data about tag identity should also be cipher text in transition. Although attacker has intercepted some data of the tag or the reader, attacker is not allowed to deduce any message about the identity of reader and tag.

The critical steps of our protocol are the second, third and fourth step. In the second step, tag sends its FID to the reader, which is just alias and has no relation with the real identity of the tag, and it would be updated after each verification round. Although attacker intercepts this alias, it cannot obtain any effective identity message of the tag. The third and the fourth steps send the A||B of the reader to the tag, and send F||H of the tag to the reader respectively. The previous mutual verification has analyzed the random number $r_1$ of the tag, random number $r_2$ of the reader, FID of the tag and the shared secret key $P_{n+1}$ and secret key $K_n$ obtained by calculation. As long as the initialization is in secret, attacker cannot calculate those values. Although attacker can intercept them, those intercepted values cannot be used to deduce the secret key and deduce the tag identity.

Above all, this protocol guarantees the data confidentiality and the integrity of the whole RFID system.

### 5.3. Steal Attack

Attacker steals the data of the reader and the tag to make the data cannot be received by either tag or reader, as a way to damage the RFID system.

Assume that attacker steals the last message (failure/success) from reader to the tag, and promotes reader to update the secret key of the database except for the secret key of the tag. In this way, the data of tag and the database are not synchronous. Our protocol solves this problem by maintaining the FID and secret key $P_n$ and $K_n$ of the last verification. Even the tag cannot update its secret key value, the database can find its FID.

In summary, this protocol can defend against steal attack.

### *5.4. Replay Attack*

Attacker iteratively sends the message that has been received by the reader to cheat reader, which would make reader to response so that the RFID system is damaged.

Assume the attacker intercepts the message during a mutual verification: ①FID, $r_1$; ②A | | B; ③F | | H. Then, attacker intercepts the last message from reader to the tag (failure/success), and the reader updates the secret value of the database as a result, except for the secret key of tag. In the next verification round, attacker firstly sends FID, $r_1$ that are intercepted previously to the reader. After receiving those message, reader searches this FID and finds that it matches to a $\text{FID}^{\text{old}}$ . Later, reader generates the random number $r_2$ and produce A | | B by equation. If attacker can send the previously obtained F | | H to activate a forgery verification, however, during this verification, the random number generated by the reader is different from the random number produced in the last round, which results in the difference between H and $H'$, so that the tag verification will fail. Above all, this protocol defends against replay attack.

### *5.5. Backtracking Attack*

When a RFID tag is deciphered, attacker cannot deduce previous verification information by this current verification.

Assume that an attacker cracks the current verification process, and obtains the FID of the tag and the secret value ($P_n$, $K_n$). However, attacker cannot deduce previous secret key values in verification because these secret key values are updated with the random number $r_1, r_2$ after each verification round.

Therefore, this protocol guarantees the safety against the backtracking.

### *5.6. Clone Attack*

Attacker copies the legal tag to obtain a clone tag, which contains all information of the copied tag, including the unique ID, data and algorithms stored in tag. In other words, when reader sends a random number and a request to the tag, clone tag can response to the reader in a same way as the legal tag. Thus, simple encryption algorithms and increase of the complexity of the verification algorithms cannot prevent against clone attack.

Our protocol is based on PUF technique to achieve the mutual verification. In the previous introduction of the PUF technique, we have analyzed its safety. Although attacker launches attack on reader or tag, and obtains the secret key of the tag: {$P_n$, FID, $K_n$}, attacker cannot produce a clone tag with these values. This is because PUF technique is by deriving the difference among chips during manufacture, which causes the different outputs. In this way, clone attack can be prevented against by our protocol.

### *5.7. Desynchronization Attack*

Attacker triggers the update of tag by some means except for the update of the reader or vice-versa. As a result, the desynchronization between the data of the database and the tag data exists, and tag cannot be verified by the reader in following verification.

Attack one: Modifying message A or message B

Assume attacker is intercepting a verification process, until the reader sending A | | B to the tag, attacker modifies this message into $A''||B''$, and then sends it the tag. After receiving $A''||B''$, tag firstly deduces the random number $r_2$ of the reader by $A''$, by which tag calculates $B'$. Then, tag judges if $B' = B''$ as a way to verify reader. Because attacker modifies message B into $B''$, reader cannot be successfully verified, which defends against the desynchronization caused by modifying message A or message B.

Attack two: Modifying message F or message H.

Assume attacker is intercepting a verification process, until confirmation of the identity of the reader, tag calculates message F||H and sends it to reader to verify. At this moment, attacker modifies this message into $F^{''}||H^{''}$, and sends the modified message to reader. After receiving $F^{''}||H^{''}$, reader firstly deduces the secret key $E'$ by $F^{''}$, then reader calculates $H'$. After this, reader judges if $H' = H^{''}$ or not to verify the tag. Because attacker modifies message H into $H^{''}$, reader cannot verify tag successfully, so that both tag and reader cannot update their own data, which thus solves the problem of desynchronization caused by modifying F or modifying H.

Compared with previous Kulseng's lightweight mutual verification protocol, Table 2 shows the advantages of our protocol.

**Table 2.** Comparison of security analysis between the proposed protocol and Kulseng's lightweight mutual verification protocol.

| Types of Attack | Proposed Protocol | K Protocol |
|---|---|---|
| Mutual verification | √ | √ |
| Data confidentiality and Tag anonymity | √ | × |
| Steal attack | √ | × |
| Replay attack | √ | √ |
| Backtracking attack | √ | × |
| Clone attack | √ | √ |
| Desynchronization attack | √ | × |

## 6. Experiments

This section shows the secure tag verification in a RFID system according to our protocol. We simulate to construct a vehicle cargo invading detection system, and all vehicles of this system carry RFID tags. With ultrahigh frequency RFID reader, the tag of a vehicle would be recorded, which information is recorded by the database and operating platform executes the secure verification via our protocol.

Ultrahigh frequency reader: Impinj R420. Air interface protocol: EPC global UHF Class 1 Gen 2/ISO 18000-6C, using Autopilot technique of the Speedway Revolution, which is able to automatically optimize the performance of the reader in any environment to maintain the best performance. Autopilot technique has following features:

1. Automatically setting—Optimize the deployment of reader to achieve the best performance.
2. Low load circulation—Reduce radio-frequency interference, power dissipation, energy cost.
3. Dynamic antennae switch—Increase handling capacity, enhance the reader to more effective work.

Speedway Revolution RFID reader also supports power over Ethernet (PoE), increases the flexibility of its application, so that the deployment is simplified and does not need AC power supply. As a result, the cost is dramatically reduced. Reader is equipped with the capability of the best flexibility in receiving, anti-interference and product and carrier offset capability.

RFID tag: Impinj H47, which supports air interface protocol EPC Class 1 Gen 2/ISO18000-6C; working frequency: 860~960 MHz, read distance is decided by the reader emit power. The tag can be fixed on the object in a sticker manner, and is easy to be read for obtaining information.

Platform construction tools: Java development platform.

Operating system: Windows 10.

In this section, we mainly discuss the RFID readers of what frequency we choose. Different frequency readers are limited by different distance when readers read tags' information. If the distance is too long, the reader cannot read the information of the tag, which will lead to the occurrence of missed reading. The counterfeiters can mix the real and fake tags together and put the fake tags out of the range to allow them to cheat the reader. Therefore, it is important to discuss how to use this protocol in terms of different frequencies and ranges of RFID readers.

In the experiments, to verify the impact of using high frequency reader and the ultrahigh frequency reader on the system alarm's success rate, in the cases with different distance between the tag and the reader, through experiencing 100 times experimental result analysis, the results are shown by Figure 6, we can conclude: within 10 cm from high frequency RFID reader, the number of successfully finding the clone tag are: 100, 100, 100, 99, 99, 100, 98, 97, 94. The distance between the reader and tag is over the best reading distance 8 cm, because the Mifare card data cannot be effectively read and the success rate dramatically decreases.

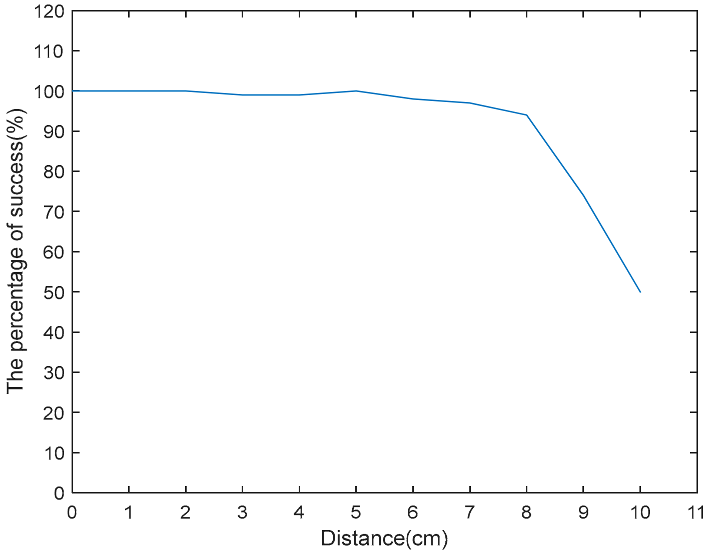


**Figure 6.** High frequency experiments.

As shown in Figure 7, in conclusion: within 110 cm from the RFID ultrahigh frequency reader, times of the reader being attacked are respectively: 100, 100, 100, 100, 99, 97, 98, 96, 95, 94, 91, 78. When the effective distance is bigger than the nominal range 100 cm, because reader cannot obtain the tag data, success rate obviously reduces. Experimental sample data is given in Figure 7.

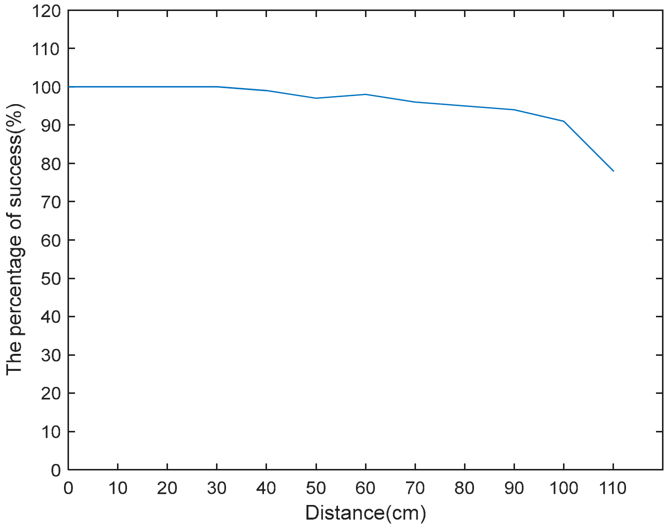


**Figure 7.** Ultra-high frequency experiments.

This experiment indicates when the high frequency reader is located over the nominal reading range, which is more than 8 cm, the performance dramatically falls, the rate of invading tag dramatically reduces. With our experimental sample, comprehensive success rate is 91.90%; while the ultrahigh frequency reader's performance slightly reduces when the distance is over the nominal reading distance, when the distance is over 90 cm, the performance of the reader begins to reduce. In comparison, ultrahigh frequency RFID reader of the original system performs better in detecting fake tag.

## 7. Performance Analysis

By calculating the costs of time and storage space for a complete certification process of backstage database, reader and tag are used to judge the performance of the proposed protocol. In this section, the reader and the backend database are viewed as a whole for the convenience of analysis, which is called the reader side. For the protocol presented in this paper, the data stored in the tag, reader, and back-end database is a unit of 96 bits, let L substitute for 96 bits. $T_P$ defines as the number of PUF function operations. $T_R$ defines as the number of times when a random number is generated. $T_{XOR}$ defines as the number of XOR operation. $T_{LEFT}$ defines as the number of times of the loop left movement, $T_{OR}$ defines as the number of join operation, $T_{AND}$ defines as the number of AND operation, and $T_F$ defines as the number of F function operation. Then the complete certification process will be analyzed in the term of its complexity.

Firstly, we analyze the time complexity.

For the RFID tag, in tag verification phase, a random number of $r_1$ is generated in the tag recognition phase, so the time complexity of the tag is $T_R$; in the process of authenticating the reader of Mutual Verification phase, there are four operations by a tag, such as $D = PUF(P_n)$, $E = PUF(D)$, $r'_2 = A \oplus FID \oplus D \oplus K_n \oplus r_1$, $B' = (r_1 <<< 8)\ \&\ (r'_2 <<< 1) \oplus (D <<< 2)$. So in the process of authenticating the reader of Mutual Verification phase, the time complexity of the tag is $(2T_P + 5T_{XOR} + 3T_{LEFT} + T_{AND})$. In the process of authenticating the tag of Mutual Verification phase, there are three operations by a tag, such as $F = D \oplus E \oplus r'_2$, $H = (FID <<< 8)\ \&\ (F <<< 1) \oplus (r'_2 <<< 2)$and $F||H$. So in the process of authenticating the tag of Mutual Verification phase, the time complexity of the tag is $(3T_{XOR} + 3T_{LEFT} + T_{AND} + T_{OR})$. In the whole Mutual Verification phase, the time complexity of the tag is $(2T_P + 8T_{XOR} + 6T_{LEFT} + 2T_{AND} + T_{OR})$. In the update phase, there are two operations by a tag, such as $FID = (FID <<< 8)\ \&\ (r'_2 <<< 1) \oplus (r_1 <<< 2)$ and $K_n = (K_n <<< 8)\ \&\ (r'_2 <<< 1) \oplus (r_1 <<< 2)$. Thus, the time complexity of the tag is $(2T_{XOR} + 6T_{LEFT} + 2T_{AND})$ in the update phase.

For the reader side, in the process of authenticating the reader of Mutual Verification phase, there are three operations by the reader side, such as $A = FID \oplus P_{n+1} \oplus K_n \oplus r_1 \oplus r_2$, $B = (r_1 <<< 8)\ \&\ (r_2 <<< 1) \oplus (P_{n+1} <<< 2)$ and $A||B$. So in the process of authenticating the reader of Mutual Verification phase, the time complexity of the reader side is $(T_R + 5T_{XOR} + 3T_{LEFT} + T_{AND} + T_{OR})$. In the process of authenticating the tag of Mutual Verification phase, there are three operations by the reader side, such as $E' = F \oplus r_2 \oplus P_{n+1}$, $F' = E' \oplus r_2 \oplus P_{n+1}$, $H' = (FID <<< 8)\ \&\ \left(F' <<< 1\right) \oplus (r_2 <<< 2)$. So in the process of authenticating the tag of Mutual Verification phase, the time complexity of the reader side is $(5T_{XOR} + 3T_{LEFT} + T_{AND})$. So in the whole Mutual Verification phase, the time complexity of the reader side is $(T_R + 10T_{XOR} + 6T_{LEFT} + 2T_{AND} + T_{OR})$. In the update phase, there are two operations by the reader side, such as $K_n^{new} = (K_n^{new} <<< 8)\ \&\ (r'_2 <<< 1) \oplus (r_1 <<< 2)$ and $FID^{new} = (FID^{new} <<< 8)\ \&\ (r'_2 <<< 1) \oplus (r_1 <<< 2)$. So in the update phase, the time complexity of the reader side is $(2T_{XOR} + 6T_{LEFT} + 2T_{AND})$. Tables 3 and 4 give the results of the time cost analysis of our protocol and the K protocol.

**Table 3.** Results of the time cost analysis of K protocol.

| Device | Tag Verification Phase | Mutual Verification Phase | Update Phase |
|---|---|---|---|
| Tag | - | $2T_P + 3T_{XOR} + 2T_F$ | $T_{XOR} + T_F$ |
| Reader | - | $2T_{XOR} + 2T_F$ | $T_{XOR} + T_F$ |

**Table 4.** Results of the time cost analysis of the proposed protocol.

| Device | Tag Verification Phase | Mutual Verification Phase | Update Phase |
|---|---|---|---|
| Tag | $T_R$ | $2T_P + 8T_{XOR} + 6T_{LEFT} + 2T_{AND} + T_{OR}$ | $2T_{XOR} + 6T_{LEFT} + 2T_{AND}$ |
| Reader | - | $T_R + 10T_{XOR} + 6T_{LEFT} + 2T_{AND} + T_{OR}$ | $2T_{XOR} + 6T_{LEFT} + 2T_{AND}$ |

Secondly, we analyze the space complexity.

For the RFID tag, there are some data in a tag, such as $\mathrm{P_n}$, FID, $\mathrm{K_n}$. Thus, the space complexity of a tag is 3L.

For the reader side, there are some data in the reader side, such as $\mathrm{FID^{old}}$, $\mathrm{P_n^{old}}$, $\mathrm{P_{n+1}^{old}}$, ID, $\mathrm{K_n^{old}}$, $\mathrm{FID^{new}}$, $\mathrm{P_n^{new}}$, $\mathrm{P_{n+1}^{new}}$, $\mathrm{K_n^{new}}$. So the space complexity of the reader side is 9 L. The comparison results are shown in the Table 5.

**Table 5.** Results of space cost analysis between the proposed protocol and K protocol.

| Protocol | Tag | Reader |
|---|---|---|
| K protocol | 3 L | 4 L |
| Our protocol | 3 L | 9 L |

In our protocol, it only uses some simple operations, such as XOR operation, the loop left movement operation, JOIN operation, and AND operation. In addition to PUF circuit by the arbiter judged by two signal circuit who arrives firstly, the amount of calculation is very small. So our protocol meets the requirements of low cost in computational overhead. To sum up, although the protocol proposed in this paper is more complex than the K protocol, its security is much higher than that of K protocol. Therefore, this protocol has found a good balance between performance and security.

## 8. Conclusions

In this paper, a lightweight RFID security protocol based on the Physical Unclonable Function (PUF) is proposed to achieve efficient verification of a single tag. The protocol includes three process: Tag recognition, mutual verification and update. The tag recognition is that the reader recognizes the tag; mutual verification is that the reader and tag mutually verify the authenticity of each other; update is supposed to maintain the latest secret key for the following verification. The results of security show that the presented protocol is efficient to protect RFID systems. Implementation of the presented protocol in real RFID systems are our future work.

**Acknowledgments:** This work is financially supported by the National Natural Science Foundation of P. R. China (No.: 61572260, No.: 61572261, No.: 61672296, No.: 61602261), Natural Science Foundation of Jiangsu Province (No.: BK20140888), Jiangsu Natural Science Foundation for Excellent Young Scholars (No.: BK20160089), Scientific & Technological Support Project of Jiangsu Province (No.: BE2015702, BE2016185, No.: BE2016777), China Postdoctoral Science Foundation (No.: 2014M561696), Jiangsu Planned Projects for Postdoctoral Research Funds (No.: 1401005B), Postgraduate Research and Practice Innovation Program of Jiangsu Province (No.: SJLX16_0326, No.: KYCX17_0798), Project of Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks (No.: WSNLBZY201509).

**Author Contributions:** He Xu and Jie Ding conceived and designed the experiments, and wrote the paper; Peng Li performed the experiments and analyzed the data; Ruchuan Wang contributed RFID devices and guided their use. Feng Zhu helped with language modification. All authors contributed to the final version.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Ahuja, S.; Potti, P. An Introduction to RFID Technology. *Commun. Netw.* **2010**, *2*, 183–186. [CrossRef]
2. Zomorrodi, M.; Karmakar, N.C.; Bansal, S.G. Introduction of electromagnetic image-based chipless RFID system. In Proceedings of the IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing, Melbourne, Australia, 2–5 April 2013; pp. 443–448.
3. Domdouzis, K.; Kumar, B.; Anumba, C. Radio-Frequency Identification (RFID) applications: A brief introduction. *Adv. Eng. Inform.* **2007**, *21*, 350–355. [CrossRef]
4. Laran, R. A Basic Introduction to RFID Technology and its Use in the Supply Chain. *J. Control. Release* **2009**, *136*, 79–85.
5. Gadh, R.; Roussos, G.; Michael, K.; Huang, G.Q.; Prabhu, B.S.; Chu, P. Guest Editors Introduction: RFID—A Unique Radio Innovation for the 21st Century. *Proc. IEEE* **2010**, *98*, 1546–1549. [CrossRef]
6. Miragliotta, G.; Perego, A.; Tumino, A. A quantitative model for the introduction of RFID in the fast moving consumer goods supply chain: Are there any profits. *Int. J. Oper. Prod. Manag.* **2009**, *29*, 1049–1082. [CrossRef]
7. Coltman, T.; Gadh, R.; Michael, K. RFID and Supply Chain Management: Introduction to the Special Issue. *J. Theor. Appl. Electron. Commer. Res.* **2008**, *3*, 3–6.
8. Shen, Y.; Wang, Z.; Zhou, C. RFID Principle and Its Application in Vehicle. *Storage Transp. Preserv. Commod.* **2008**, *30*, 44–46.
9. Ferrero, R.; Gandino, F.; Montrucchio, B.; Rebaudengo, M.; Zhang, L. A novel simulator for RFID reader-to-reader anti-collision protocols. In Proceedings of the International EURASIP Workshop on RFID Technology, Rosenheim, Germany, 22–23 October 2015; pp. 59–64.
10. Want, R. *An Introduction to RFID Technology*; IEEE Educational Activities Department: Piscataway, NJ, USA, 2006.
11. Ha, J.C.; Moon, S.J.; Nieto, J.M.G.; Boyd, C. Low-Cost and Strong-Security RFID Authentication Protocol. In *Emerging Directions in Embedded and Ubiquitous Computing, Proceedings of the EUC 2007 Workshops: TRUST, WSOC, NCUS, UUWSN, USN, ESO, and SECUBIQ, Taipei, Taiwan, 17–20 December 2007*; Springer: Berlin, Germany, 2012; pp. 795–807.
12. Sharma, R.; Singh, P.K. The Simulation and Analysis of RC4 and 3DES Algorithm for Data Encryption in RFID Credit Card. *Int. J. Appl. Eng. Res.* **2015**, *10*, 4265–4273.
13. Hsu, C.H.; Wang, S.; Zhang, D.; Chu, H.C.; Lu, N. Efficient identity authentication and encryption technique for high throughput RFID system. *Secur. Commun. Netw.* **2016**, *9*, 2581–2591. [CrossRef]
14. Gódor, G.; Imre, S. Hash-Based Mutual Authentication Protocol for Low-Cost RFID Systems. In *Information and Communication Technologies*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 76–87.
15. Jin, C.; Xu, C.; Zhang, X.; Li, F. A Secure ECC-based RFID Mutual Authentication Protocol to Enhance Patient Medication Safety. *J. Med. Syst.* **2016**, *40*, 12–17. [CrossRef] [PubMed]
16. Dehkordi, M.H.; Farzaneh, Y. *Improvement of the Hash-Based RFID Mutual Authentication Protocol*; Kluwer Academic Publishers: Alphen, The Netherlands, 2014; Volume 75, pp. 219–232.
17. Sun, D.Z.; Zhong, J.D. Cryptanalysis of a Hash Based Mutual RFID Tag Authentication Protocol. *Wirel. Pers. Commun.* **2016**, *91*, 1085–1093. [CrossRef]
18. D'Arco, P.; Santis, A.D. On Ultralightweight RFID Authentication Protocols. *IEEE Trans. Dependable Secure Comput.* **2011**, *8*, 548–563. [CrossRef]
19. Peng, Q.; Zhang, C.; Zhao, X.; Sun, X.; Li, F.; Chen, H.; Wang, Z. A Low-Cost UHF RFID System with OCA Tag for Short-Range Communication. *IEEE Trans. Ind. Electron.* **2015**, *62*, 4455–4465. [CrossRef]
20. Gao, L.; Ma, M.; Shu, Y.; Wei, Y. An ultralightweight RFID authentication protocol with CRC and permutation. *J. Netw. Comput. Appl.* **2014**, *41*, 37–46. [CrossRef]
21. Tewari, A.; Gupta, B.B. *Cryptanalysis of a Novel Ultra-Lightweight Mutual Authentication Protocol for IoT Devices Using RFID Tags*; Kluwer Academic Publishers: Alphen, The Netherlands, 2017.
22. Lehtonen, M.; Michahelles, F.; Fleisch, E. How to Detect Cloned Tags in a Reliable Way from Incomplete RFID Traces. In Proceedings of the IEEE International Conference on RFID, Orlando, FL, USA, 27–28 April 2009; pp. 257–264.
23. Zanetti, D.; Fellmann, L.; Capkun, S. Privacy-preserving Clone Detection for RFID-enabled Supply Chains. In Proceedings of the IEEE International Conference on RFID, Orlando, FL, USA, 14–16 April 2010; pp. 37–44.

24. Ray, B.; Chowdhury, M.; Abawaiy, J. PUF-based secure checker protocol for Networked RFID Systems. In Proceedings of the IEEE Conference on Open Systems, Subang, Malaysia, 26–28 October 2015; pp. 78–83.
25. Gao, Y.; Li, G.; Ma, H.; Al-Sarawi, S.F.; Kavehei, O.; Abbott, D.; Ranasinghe, D.C. Obfuscated challenge-response: A secure lightweight authentication mechanism for PUF-based pervasive devices. In Proceedings of the IEEE International Conference on Pervasive Computing and Communication Workshops, Sydney, Australia, 14–18 March 2016; pp. 1–6.
26. Devadas, S.; Suh, E.; Paral, S.; Sowell, R.; Ziola, T.; Khandelwal, V. Design and Implementation of PUF-Based "Unclonable" RFID ICs for Anti-Counterfeiting and Security Applications. In Proceedings of the IEEE International Conference on RFID, Las Vegas, NV, USA, 16–17 April 2008; pp. 58–64.
27. Kulseng, L.; Yu, Z.; Wei, Y.; Guan, U. Lightweight mutual authentication and ownership transfer for RFID systems. In Proceedings of the IEEE Conference on INFOCOM, San Diego, CA, USA, 14–19 March 2010; IEEE Press: Piscataway, NJ, USA, 2010; pp. 251–255.
28. Kardas, S.; Akgun, M.; Kiraz, M.S.; Demirci, H. Cryptanalysis of Lightweight Mutual Authentication and Ownership Transfer for RFID Systems. In Proceedings of the Workshop on Lightweight Security & Privacy: Devices, Protocols, and Applications, Istanbul, Turkey, 14–15 March 2011; IEEE Computer Society: Washington, DC, USA, 2011; pp. 20–25.
29. Gong, L.; Needham, R.; Yahalom, R. Reasoning about belief in cryptographic protocols. In Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, 7–9 May 1990; pp. 234–248.