A Secure and Private RFID Authentication Protocol Based on Quadratic Residue

Süleyman Kardaş * †, Serkan Çelik * †, Mehmet Sariyüce * and Albert Levi†
*TÜBİTAK BİLGEM UEKAE, Kocaeli, Turkey
†Sabancı University, Faculty of Engineering and Natural Sciences, İstanbul, Turkey

***Abstract*—Radio Frequency IDentification based systems are getting pervasively deployed in many real-life applications in various settings for identification and authentication of remote objects. However, the messages that are transmitted over a insecure channel, are vulnerable to security and privacy concerns such as data privacy, location privacy of tag owner and etc. Recently, Yeh et al.'s proposed a RFID authentication protocol based on quadratic residue which is claimed to provide location privacy and prevent possible attacks. In this paper, we formally analyzed the protocol and we proved that the protocol provides destructive privacy according to Vaudenay privacy model. Moreover, we proposed a unilateral authentication protocol and we prove that our protocol satisfies higher privacy level such as narrow strong privacy. Besides, we proposed an enhanced version of our proposed protocol, which has same privacy level as Yeh at al protocol, but has reader authentication against stronger adversaries. Furthermore, the enhanced version of our protocol uses smaller number of cryptographic operations when compared to Yeh at al protocol and it is also cost efficient at the server and tag side and requires $\mathcal{O}(1)$ complexity to identify a RFID tag.**

***Index Terms*—RFID, Privacy, Security, Quadratic Residue.**

1. Introduction

Radio Frequency IDentification (RFID), which is one of the most important ubiquitous technologies, is widely adopted in the enterprises for inventory checking and management. It is a common way of remote object identification and authentication, uses radio wave signals.

A typical RFID system consists of three main components: the transponder or RFID tag, the transceiver or RFID reader, and the back-end database. RFID readers are commonly composed of an RF module, a control unit, and a coupling element to interrogate the tags by means of RF communication [11]. It is assumed that an attacker is able to monitor and intercept the communications between readers and tags, however, the interactions between the readers and the back-end database are secure. In RFID systems, the tagged object does not need to be in the line of sight but earlier technologies such as the barcode and smart cards do. This is a significant difference between RFID and the earlier technologies.

On account of the ease of deployment and low cost, RFID technology has been widely deployed into many daily life applications such as automation technology, supply chain management, transportation, and even passport identification [18]. Such use of RFID raises security and privacy concerns against strong adversary such as location privacy of tag owner, confidentiality, availability and etc. Since RFID labels used in daily life applications are low-cost devices and have limited resources, the challenge on addressing the security and privacy concerns are much harder than traditional technology.

Besides the passion of having secure authentication protocols, entire system performance has become an important issue. Since, designing authentication protocol without sacrificing security and privacy begets decreasing efficiency of whole system. However, achieving the security and privacy properties, the complexity in tag and server side can vary dramatically from one protocol to another. Hence, while handling security and privacy issues, it is also important to realize them with less computational complexity in the server and tag side.

In order to resolve these security and privacy issues, numerous RFID authentication protocols have been recently proposed in the literature [1], [5], [7]–[9], [12], [15]. Many of them are failed to provide security and privacy and the computation on the server side is also very high. Recently, Yeh *et al.* proposed an improvement of the RFID authentication protocol [6] which utilizes quadratic residue for security and privacy [17]. It requires constant time at the server side for identification; however, this proposal has lack of a formal security and privacy analysis.

Our Contributions. In this paper, we first present an analysis of Yeh et al. authentication protocol according to Vaudenay's model and prove that this protocol satisfies at most destructive privacy but the tag and reader authentication are secure against at most weak adversary. Then, we propose a unilateral authentication protocol which achieves narrow strong privacy. After that, we proposed an enhanced version of proposed protocol, which satisfies mutual authentication with reader authentication against stronger adversaries, achieves destructive privacy according to Vaudenay's model. Note that, our proposed protocol and enhanced version of it need constant-time complexity to identify and authenticate a tag.

The outline of the paper is as follows. In Section 2, we give a brief discussion on Vaudenay's security and privacy model, and formal model on security. Section 3 describes Yeh *et al.*'s proposed protocol and gives its security and privacy analysis. In Section 4, the first proposed protocol with security and privacy analysis is given in a detail. In Section 5, analysis of our second mutual protocol is given in a detail. In Section 6, we conclude the paper.

2. Formal Tools for Security and Privacy Analysis

We divide this section into two parts. In the first part, we summarize Vaudenay's privacy model. In the second part, we give brief information about ProVerif which is a tool used in security analysis.

A. Vaudenay's privacy model

Vaudenay's privacy model [16] is one of the most systematic and generic models, so we apply this model for our privacy analysis. In Vaudenay model, one can see the boundaries of a strong malicious adversary who can monitor all communications, trace tags within a limited period of time, corrupt tags, and get side channel information on the reader output [16].

Vaudenay defines an RFID scheme by following procedures.

- SetupReader(1^s) : This algorithm first creates a public key pair (K_P, K_S) and initializes its database $\mathcal{DB}$.
- SetupTag$_{K_P}$(ID): This algorithm produces a tag secret K and the initial state S of a tag with ID. If it is a valid tag, the pair (ID,K) is added to $\mathcal{DB}$.

An adversary $\mathcal{A}$ communicates with the RFID system with generic eight oracles defined in [16].

Vaudenay also defines eight adversarial classes with different capabilities.

Definition 2.1. *(Adversary Classes [16]) An adversary $\mathcal{A}$ is a p.p.t. algorithm which has arbitrary number of access to all oracles described above. Weak $\mathcal{A}$ uses all oracles except* Corrupt *oracle. Forward $\mathcal{A}$ is allowed to use only* Corrupt *oracle after her first call to the oracle. Destructive $\mathcal{A}$ cannot use any oracles against a tag after an* Corrupt *oracle on the tag. Strong $\mathcal{A}$ uses all oracles defined above without any restrictions. Narrow $\mathcal{A}$ has no access to* Result *oracle.*

An RFID scheme is given with three cryptographic properties such as correctness, security, and privacy. Correctness is implicitly assumed. The security definition is already defined in [16]. Here, we present the privacy game of Vaudenay as follows.

Definition 2.2. *(Privacy [16]). The adversaries who start with an attack phase allowing oracle queries then pursuing an analysis phase with no oracle query. In between phases, the adversary receives the hidden table T of the DrawTag oracle then outputs either true or false. The adversary wins if the output is true. We say that the RFID scheme is P-private if all such adversaries which belong to class P are trivial following Definition 2.3.*

Definition 2.3. *(Blinder [16]) A blinder $\mathcal{B}$ is a simulator which simulates the* Launch, SendReader, SendTag, *and* Result *oracles without having access to the secret keys and the database. When a blinded adversary $\mathcal{A}^B$ makes the* Launch, SendReader, SendTag, *and* Result *queries, she is answered through the blinder $\mathcal{B}$.*

Remark 1. *The blinder $\mathcal{B}$ is consistent and acts like a real reader in a way that if a protocol transcript's inputs are derived as a result of usage of oracles to $\mathcal{B}$, the answer given by $\mathcal{B}$ to the* Result *oracle on this protocol transcript is 1. If all inputs of a protocol transcript are not derived as a result of usage of oracles to $\mathcal{B}$, then the answer given by $\mathcal{B}$ to the* Result *oracle on this protocol transcript depends on the appearance probability of missing inputs on protocol transcript. Besides, $\mathcal{B}$ holds all its answers to the oracles used by $\mathcal{A}$ in its database and answers the new oracles depending on its database.*

Note that, in this paper, in all protocol descriptions, tags only include T_{ID} as a tag related information. Hence, when Result oracle is applied, for the current protocol run, the notion of privacy is meaningless. Thus,we look for privacy for protocol runs where Corrupt oracle takes place. As a reference, following remark can be given.

Remark 2. *In this paper, the adversary is not allowed to distinguish between the real system and the blinder at protocol runs where* Corrupt *oracle takes place.*

B. Security Analysis

Securing a system is a complex problem since it requires a careful analysis of the underlying assumptions about cryptographic functions and trusted parties, and an accurate implementation of hardware and software. Satisfying all these requirements is virtually impossible without the use of formal analytical techniques [13] which are invaluable tools for identifying weaknesses in security protocols.

In order to verify formally whether an authentication protocol achieves a certain security property, we first create a model which specifies the capability of an adversary. Then, we describe the interactions of the adversary in this model and the definition of the security property within the model. Finally, by using this model, a formal tool checks whether the goals in the security protocol are achieved or not. Recently, several different symbolic formal models have been proposed in the literature [2], [3], [14]. In our analysis, we use ProVerif [3] which is automatic tool to verify a wide range of security of cryptographic protocols.

Properties of the processes described in the applied pi-calculus can be proved by automated tools ProVerif [4]. ProVerif first translates the applied pi-calculus process into a set of Horn clauses. These clauses account for the initial knowledge of the attacker and the inference rules she can apply to broaden her knowledge pool for the messages. ProVerif can prove reachability properties that are typical of model checking tools such as correspondence assertions, and observational equivalence. ProVerif can also reconstruct an execution trace that falsifies the desired property: when a desired property cannot be proved. Furthermore, in ProVerif analysis, protocol analysis is considered in accordance with an infinite number of sessions, an unbounded message space and parallel sessions.

3. Yeh et al.'s Proposed Protocol and Its Privacy Analysis

In this section, we first present Yeh et al.'s authentication protocol [17] by considering the server and the reader as

a single entity, just reader, since the channel between these two entities is assumed to be secure. Then, we analyze the protocol according to Vaudenay privacy model. We prove that this protocol satisfies destructive privacy. The protocol steps are described as follows.

Let $h : \{0,1\}^* \rightarrow \{0,1\}^\alpha$ be a hash function and $PRNG : \{0,1\}^\alpha \rightarrow \{0,1\}^\alpha$ be a pseudo-random number generator. Let $r, s, t, n \in \{0,1\}^\alpha$. Each tag $\mathcal{T}$ is equipped with a unique $\mathcal{T}_{ID}$ and stores the value n and r. These values are given by reader in the initialization phase. Reader stores the values $h(\mathcal{T}_{ID}), \mathcal{T}_{ID}, r, r_{old}$ where $r_{old} = r$ at the beginning.

In the protocol, the reader $\mathcal{R}$ first sends a random challenge $s \in_R \{0,1\}^\alpha$ to a tag $\mathcal{T}$. Once $\mathcal{T}$ receives the challenge, $\mathcal{T}$ picks another random challenge $t \in_R \{0,1\}^\alpha$. $\mathcal{T}$ constructs x, y, X, R and T respectively, then sends $X, R, T, h(x), h(y)$ and $h(t)$ to $\mathcal{R}$. Then, $\mathcal{R}$ gets (x_1, x_2, x_3, x_4) and (t_1, t_2, t_3, t_4) by solving $X = x^2 \mod n$ and $T = t^2 \mod n$ by using the factors of n, which are p and q. After that $\mathcal{R}$, determines correct values of x and t by comparing $h(x_i) \stackrel{?}{=} h(x)$ and $h(t_i) \stackrel{?}{=} h(t)$. Then, $\mathcal{R}$ determines the correct value of r in a similar way. $\mathcal{R}$ computes $h(\mathcal{T}_{ID})$ and seeks $\mathcal{T}_{ID}$ from database and compares received r with r or r_{old}. If received r is valid, then computes acknowledgement message $x_{ack} = \mathcal{T}_{ID} \oplus t \oplus r$ or r_{old}, sends $h(x_{ack})$ to $\mathcal{T}$ and updates r_{old} as r as $PRNG(r)$. Then $\mathcal{T}$ checks whether $h(x_{ack}) \stackrel{?}{=} h(\mathcal{T}_{ID}) \oplus r \oplus t$. If it is valid, $\mathcal{T}$ updates r as $PRNG(r)$, otherwise the protocol aborts.

Before starting the security and privacy analysis of the protocol, we can assume, without loss of generality, there are one reader and one tag in the system. Since the variables which change tag to tag at calculation steps are $h(T_{ID})$ and r which have same bit length as s. Thus, by deriving more s values, i.e. more protocol runs, we can recover the advantage loss due to working with one tag instead of many tags.

Theorem 3.1. *Yeh et al.'s Proposed Protocol achieves tag authentication and reader authentication if the adversary $\mathcal{A}_w$ belongs to weak class.*

Proof: Let the adversary $\mathcal{A}_w$ observes n protocol runs between the reader and the tag. Let us assume that $\mathcal{A}_w$ tries to impersonate the tag at $n+1$th run. If the value of s sent by the reader is equal to the one of the s values sent at one of the previous protocol runs, $\mathcal{A}_w$ impersonates the tag with success probability 1. Otherwise, $\mathcal{A}_w$ has to guess the values of $h(T_{ID})$ and r for corresponding run correctly. Thus, the success probability for $\mathcal{A}_w$ to impersonate the tag is $\frac{n}{2^\alpha} + (1 - \frac{n}{2^\alpha})\frac{1}{2^{2m}}$, which is negligible. Hence, the system achieves tag authentication if the adversary is weak.

Similarly, if $\mathcal{A}_w$ tries to impersonate the reader, then $\mathcal{A}_w$ sends a challenge s to the tag. Upon receiving the challenge, the tag responses with $X, R, T, h(x), h(y), h(t)$ according to which t value the tag chooses. However, as $\mathcal{A}_w$ does not know the value of r, $\mathcal{A}_w$ can not figure out the value of t. Moreover, since $\mathcal{A}_w$ does not know the factors of n, which are p and q, $\mathcal{A}_w$ can not the roots of X and R and T. Besides, $\mathcal{A}_w$ has to guess correct value of T_{ID}. Thus, the probability that $\mathcal{A}_w$ sends correct $h(x_{ack})$ to the tag is $\frac{1}{2^{2m}}$, which is negligible. Therefore, the system achieves the reader authentication if the adversary is in class of weak. ■

Theorem 3.2. *Yeh et al.'s proposed protocol achieves destructive privacy but does not achieve narrow strong privacy.*

Proof: Let there are one reader and one tag in the system and let $\mathcal{A}_d$ be a destructive adversary. Assume to the contrary, the protocol does not achieve destructive privacy. That is, the adversary $\mathcal{A}_d$ can distinguish between the real RFID system and the system simulated by the $\mathcal{B}$ with non negligible probability.

Let start with how $\mathcal{B}$ evaluates oracles:

- **Launch()**: Evaluated in a trivial way.
- **SendReader(π)**: The output is $s \in_R \{0,1\}^\alpha$.
- **SendTag(s, π)**: The output is $X, R, T, h(x), h(y), h(t)$.
- **SendReader(($X, R, T, h(x), h(y), h(t)$),$\pi$)**: The output is $h(x_{ack})$.
- **Result(π)**: This oracle works as defined in Remark 1

Let the system is run n times only by the real RFID system or $\mathcal{B}$ and let $\mathcal{A}_d$ applies CORRUPT oracle at $n+1$th protocol run. $\mathcal{A}_d$ gets the values of T_{ID}, n and r_{n+1}, t_{n+1}, x_{n+1}, y_{n+1} as a result of CORRUPT oracle usage.

There are three ways for $\mathcal{A}_d$ to distinguish between the real reader from the blinder. The first way is $\mathcal{A}_d$'s guessing the correct value of r at any protocol run. If this is the case, then by using the relation $R = (r^2 mod n) \oplus t$ formula, $\mathcal{A}_d$ gets the value of t for the corresponding round. Moreover, $\mathcal{A}_d$ gets the values of x, y, X, T values of the corresponding round. Furthermore, as $\mathcal{A}_d$ can calculate next rounds' r value, in a similar way $\mathcal{A}_d$ gets the values of t, x,y, X, T values for each advancing protocol run. Therefore, if $\mathcal{A}_d$ correctly guesses r value at least 1 protocol run, then $\mathcal{A}_d$ can check correctness of the protocol at next protocol runs. Therefore, in this case, the adversary distinguishes the real system from the blinder. However, realization of this case has probability at most $1 - (1 - \frac{1}{2^\alpha})$, which is negligible. The next way for $\mathcal{A}_d$ is to guess the correct value of $h(ack)$ at any protocol run. Similarly, the realization of this case has probability at most $1 - (1 - \frac{1}{2^\alpha})$, which is negligible.

The last way is $\mathcal{A}_d$'s determining the value that is produced by $Result$ oracle is right or wrong. By contradiction assumption, $\mathcal{A}_d$'s success probability at this case is non-negligible as the success probability of previous two ways are negligible. However, this contradicts with the Theorem 3.1 as in our case, for past protocol runs, destructive adversary acts like weak adversary as r values of previous protocol runs can not be deduced from the knowledge of r_{n+1}. Thus, the protocol achieves destructive privacy.

Let $\mathcal{A}_s$ be a narrow strong adversary. In this case, let $\mathcal{A}_s$ corrupts the tag before starting any protocol run. As indicated above, $\mathcal{A}_s$ gets the value of r, and due to the nature of PRNG functions, $\mathcal{A}_s$ can calculate the value of r in any advancing run. Therefore, she can calculate the value of t, x,y, X and T at each protocol run. Hence, $\mathcal{A}_s$ can distinguish the real

system from the blinder. Thus, the protocol does not achieve narrow strong privacy. ■

4. The Proposed Protocol

In this section, we first present a novel scalable RFID authentication protocol which is based on quadratic residue. Then, we give security and privacy analysis of it according to Vaudenay model.

Let $h : \{0,1\}^* \rightarrow \{0,1\}^\kappa$ be a hash function. Let $s, n, t \in \{0,1\}^\alpha$. Each tag $\mathcal{T}$ is equipped with a unique $\mathcal{T}_{ID}$ and stores the value n. These values are given by reader $\mathcal{R}$ in the initialization phase. $\mathcal{R}$ stores the values $h(\mathcal{T}_{ID})$ and $\mathcal{T}_{ID}$. The authentication protocol is summarized in Figure 1, the upper part of dashed line.

In the protocol, $\mathcal{R}$ first sends a random challenge $s \in_R \{0,1\}^\alpha$ to a tag $\mathcal{T}$. Once $\mathcal{T}$ receives the challenge, $\mathcal{T}$ picks another random challenge $t \in_R \{0,1\}^\alpha$. $\mathcal{T}$ constructs x, X, T and M respectively, then sends X, T and M to $\mathcal{R}$. Once $\mathcal{R}$ receives X, T and M, it gets (x_1, x_2, x_3, x_4) and (t_1, t_2, t_3, t_4) by solving $X = x^2 \mod n$ and $T = t^2 \mod n$ by the help of factors on n. After that $\mathcal{R}$, determines correct values of x and t by comparing $h(x_i||t_j) \stackrel{?}{=} M$. Now, $\mathcal{R}$ can compute $h(\mathcal{T}_{ID})$ and then check existence of $\mathcal{T}_{ID}$ in the database.

A. Privacy Analysis

Before starting the security analysis of the proposed protocol, Note that, we can assume there is one reader and one tag in the system. Since the variables which change tag to tag at calculation steps are $h(TID)$ which has same bit length as s. Thus, by deriving more s values, i.e. more protocol runs, we can recover the advantage loss due to working with one tag instead of many tags.

Theorem 4.1. *The proposed RFID protocol achieves tag authentication if the adversary $\mathcal{A}_w$ belongs to the weak class.*

Proof: Let the adversary $\mathcal{A}_w$ observes n protocol run between the reader and the tag. First of all, let us assume that $\mathcal{A}_w$ tries to impersonate the tag at $n+1$st run. There are two cases to consider. If the challenge value s sent by the reader is equal to the one of the s values sent at previous protocol run, then with 1 success probability, $\mathcal{A}_w$ impersonates the tag. However, the probability of realization of this scenario is $\frac{n}{2^\alpha}$. If this is not the case, then the only way for $\mathcal{A}_w$ to impersonate the tag is to guess the value of $h(T_{ID})$ correctly. The success probability in this case $\frac{1}{2^\alpha}$. Hence, $\mathcal{A}_w$ impersonates the tag with probability $\frac{n}{2^\alpha} + (1 - \frac{n}{2^\alpha})\frac{1}{2^\alpha}$, which is negligible. Therefore, the system achieves tag authentication if the adversary is weak. ■

Theorem 4.2. *The proposed RFID protocol achieves narrow strong privacy.*

Proof: Before starting the proof steps, note that, for proposed protocol, in terms of privacy analysis, there is no real difference between the adversary's applying CORRUPT oracle only one time and more than one time. Since, at each CORRUPT oracle usage, the adversary gets the values of T_{ID} and n, which do not changes among protocol runs and session specific t and x values and there is no real connection between any of two protocol runs' corresponding values. Therefore, in the proof, the adversary applies the CORRUPT oracle only once.

Let there are one reader and one tag in the system and let $\mathcal{A}_s$ be a narrow strong adversary. Assume to the contrary, the protocol does not achieve narrow strong privacy. That is, the adversary $\mathcal{A}_s$ can distinguish between the real RFID system and the system simulated by the $\mathcal{B}$ with non negligible probability.

Let start with how $\mathcal{B}$ evaluates oracles:

- **Launch**(): Evaluated in a trivial way.
- **SendReader**(π): The output is $s \in_R \{0,1\}^m$.
- **SendTag**(s, π): The output is X, T, M.

Let the system is run n times only by the real RFID system or $\mathcal{B}$. Let $\mathcal{A}_s$ applies CORRUPT oracle at $n+1$st protocol run and after that oracle usage, the system run k more times. Note that, $\mathcal{A}_s$ gets the values of T_{ID}, n, t_{n+1} and x_{n+1} as a result of CORRUPT oracle usage.

Note that, there are 2 ways for $\mathcal{A}_w$ to distinguish the real system from the blinder. The first one is to guess t value correctly at any of previous n protocol runs or next k runs. The other way is to guess one of the X, T and M value correctly. Hence, the total success probability of the adversary is $\frac{n+k}{2^\alpha} + (1 - \frac{n+k}{2^\alpha})\frac{3}{2^\alpha}$, which is negligible. Of course, one can run this process defined above polynomially bounded time and increase the adversary's chance but the resulting success probability will be at most negligible. ■

5. An Enhanced Version of the Proposed Protocol

In this section, we propose an enhanced version of the proposed protocol(see Figure 1) which satisfies reader authentication against strong adversary and has destructive privacy level.

The protocol steps of this protocol consists of the unilateral authentication protocol and the last message sent by reader to the tag. The reader prepares $x_{ack} = TID||t||s$ and sends $h(x_{ack})$ to the tag. The tag checks validity of $h(x_{ack})$ by comparing its value with $h(TID||t||s)$. All the steps of the second protocol are summarized in Figure 1.

A. Privacy Analysis by using Vaudenay's Model

Theorem 5.1. *The protocol depicted in Figure 1 satisfies tag authentication against weak adversary and satisfies reader authentication against narrow strong adversary.*

Theorem 5.2. *The protocol demonstrated at Figure 1 achieves destructive privacy.*

Proof: Let there are one reader and one tag in the system and let $\mathcal{A}_d$ be a destructive adversary. Assume to the contrary, the protocol does not achieve destructive privacy. That is, the adversary $\mathcal{A}_d$ can distinguish between the real RFID system and the system simulated by the $\mathcal{B}$ with non negligible probability.

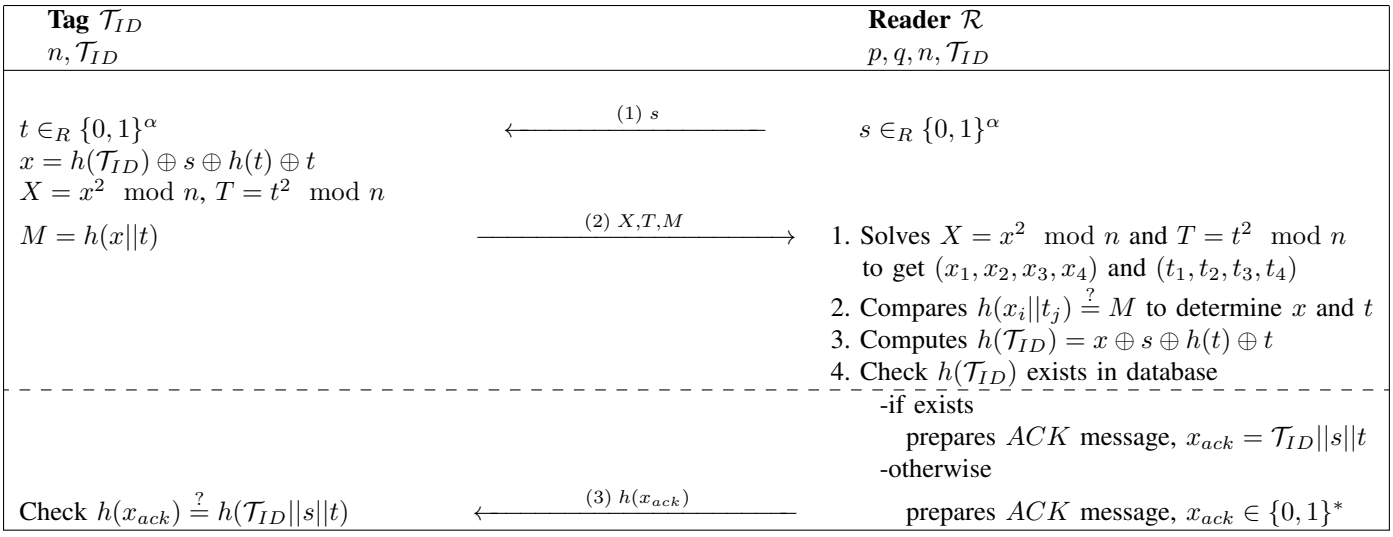

Fig. 1. Our proposed narrow strong private scheme.

$\mathcal{B}$ evaluates oracles in the same way as indicated at the proof of Theorem 4.2 with addition:

- **SendReader**$((X, T, M), \pi)$: The output is $h(x_{ack})$.
- **Result**(π): This oracle works as defined in Remark 1

Let the system is run n times only by the real RFID system or $\mathcal{B}$ and let $\mathcal{A}_d$ applies CORRUPT oracle at $n+1$st protocol run. $\mathcal{A}_d$ gets the values of TID, n and t_{n+1}, x_{n+1} as a result of CORRUPT oracle usage.

There are three cases to consider. The first case is $\mathcal{A}_d$'s guessing the value of t in any of previous n protocol runs. However, as there is no connection between t_{n+1} and previously chosen t values, the realization of first case is negligible. The second case is $\mathcal{A}_d$'s guessing the correct value os $h(x_{ack})$. Similarly, the probability of realization of this case is negligible.

The last way is $\mathcal{A}_d$'s determining the value that is produced by $Result$ oracle is right or wrong. By contradiction assumption, $\mathcal{A}_d$'s success probability at this case is non-negligible as the success probability of previous two ways are negligible. However, this contradicts with the Theorem 5.1 as in our case, for past protocol runs, destructive adversary acts like weak adversary. Thus, the protocol achieves destructive privacy. ■

B. Formal Analysis

In this section, we use ProVerif tool in order to formally prove the security property of our enhanced protocols such as reader authentication and tag authentication.

To encode the protocol into the pi-calculus, we first determine the required cryptographic primitives with function symbols, and rewrite rules and equations over terms. Let $hash()$ be a universal hash function. Let xor be the function which satisfies $\forall x, y \in \{0,1\}^\alpha$, $xor(x, y) = x \oplus y$. Note that, ProVerif cannot evaluate XOR functions properly and so we provide all possible reduction functions $(xor_1, \ldots, xor_8)$ which help ProVerif simulate XOR function. Let two large primes, (P,Q) be a factors of a common modulus N. Then, let *smodulus* denote a type of pair of (P,Q) and *pmodulus* denote a type of public modulus (N=PQ). The reader stores factors of a public modulus N ***P_and_Q*** and tag stores the modulus, *publicmod(P_and_Q)*.

We also simulate quadratic residue functions, one for taking modulo square, one for taking modulo square root. $\forall x, X \in \{0,1\}^\alpha$ and *pmodulus* $N \in \{0,1\}^\alpha$, $square(x, N)$ is equal to $x^2 \mod N$ and $ssquare(X, N)$ gives all possible solutions to $X^{-2} \mod N$.

The public channel between reader and tags are described as $free\ c : channel$. The adversary is also allowed to use this channel for her attack.

Our mutual authentication protocol is expected to satisfy (informally) the following properties:

- Authentication of tag to reader: if the reader identifies tag, it responds so that at the end of the protocol, tag has approval to engage with reader in a session, only if reader permits it.
- Authentication of reader to tag: similar to the above.
- Secrecy of session keys (combination of s and t).

In our model, we assume $secret$ is a private key shared between tag and reader which is unknown by the adversary. Our interest in this model is to verify the secrecy of the bitstring (t) generated by tag. Therefore, as soon as tag authenticates reader, tag broadcasts $secret$ XORed with the generated t $(out(c, secret \oplus t))$. If there is no way that an adversary can derive $secret$ by applying the rules, then the protocol is safe. Namely, the authentication procedure has not been compromised. In order to challenge the adversary, we write the query syntax, as the following: **query attacker(secret)**.

The behaviour of the reader is encoded into following process, *Reader*. In this process, the reader waits any message from tag on channel $in(c : channel, data)$. It sends any message to tag through the same channel $(out(c : channel, data))$.

```
1.  let Reader(TID:bitstring ) = new s:bitstring;
2.  (* Message 1 *) out(c, s);
3.  (* Message 2 *)
4.  in(c, (X:bitstring, T:bitstring, M:bitstring));
5.  let x = ssquare(X,P_and_Q) in
6.  let t = ssquare(T,P_and_Q) in
7.  let (=M) = hash((x,t)) in
8.  let HTID = hash(TID) in let HT = hash(t) in
9.  let (=HTID) = xor1(xor1(xor1(x,HT),t),s)
10. in event readerAuthTag(s,t);(* Message 3 *)
11. out(c, hash((TID,s,t))); 0.
```

The behaviour of the tag is encoded into following process:

```
12. let Tag(TID:bitstring, N : pmodulus) =
13. (* Message 1 *)
14. in(c, s:bitstring); new t:bitstring ;
15. let HT = hash(t) in let HTID = hash(TID) in
16. let x = ssquare(X,P_and_Q) in
17. let X = square(x,N) in let T = square(t,N) in
18. let M = hash((x,t)) in
19. (* Message 2 *) out(c,(X,T,M)); (* Message 3 *)
20. in(c, ack:bitstring);
21. let (=ack) = hash((TID,s,t)) in
22. event tagAuthReader(s,t);
23. out(c, xor(secret,t)) ;0.
```

These two processes are executed multiple times in parallel using the following syntax:

```
24. process
25.     let N = publicmod(P_and_Q) in out (c,N);
26.     new TID:bitstring;
27.     (!Reader(TID) | !Tag(TID,N ) | phase 1; out(c,TID))
```

In this process, we first created a public modulus N, which is sent through channel c. Then we create a new TID for a tag identifier. This TID and the private products of N (P_and_Q) are given to reader. ProVerif first converts these processes and adversary actions into a set of Horn clauses [10] so as to automatically prove queries. Then, it runs the processes and searches for a valid security gap based on requested queries. The output of ProVerif confirms that the attacker cannot derive the term $(secret)$ so the authentication procedure can be performed successfully without being compromised. Also, the attacker is not be able to cheat both reader and tag even if we provide TID of the victim tag to adversary in phase 1.

6. Conclusions

In this article, we first give a formal security and privacy analysis of Yeh et al.'s authentication protocol. We proved that this protocol provides at most destructive privacy according Vaudenay model whereas the tag and reader authentication is secure against at most weak adversary. Then, we introduced an unilateral authentication protocol and we formally proved that this protocol achieves narrow strong adversary. We also proposed the enhanced version of the protocol that provides reader authentication. We proved that the second protocol satisfies destructive privacy and the reader authentication is secure against narrow strong adversary.

References

[1] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran. Scalable rfid systems: a privacy-preserving protocol with constant-time identification. *Dependable Systems and Networks, International Conference on*, 0:1–10, 2010.

[2] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron. The avispa tool for the automated validation of internet security protocols and applications. In *Proceedings of the 17th international conference on Computer Aided Verification*, CAV'05, pages 281–285, Berlin, Heidelberg, 2005. Springer-Verlag.

[3] B. Blanchet. An efficient cryptographic protocol verifier based on prolog rules. In *Proceedings of the 14th IEEE workshop on Computer Security Foundations*, CSFW '01, pages 82–, Washington, DC, USA, 2001. IEEE Computer Society.

[4] B. Blanchet and B. Smyth. Proverif 1.86pl3: Automatic cryptographic protocol verifier, user manual and tutorial. http://www.proverif.ens.fr/manual.pdf, 2012.

[5] M. Burmester, B. de Medeiros, and R. Motta. Anonymous rfid authentication supporting constant-cost key-lookup against active adversaries. *IJACT*, 1(2):79–90, 2008.

[6] Y. Chen, J.-S. Chou, and H.-M. Sun. A novel mutual authentication scheme based on quadratic residues for rfid systems. *Computer Networks*, 52(12):2373 – 2380, 2008.

[7] A. Fernandez-Mir, R. Trujillo-Rasua, and J. Castella-Roca. Scalable RFID Authentication Protocol Supporting Ownership Transfer and Controlled Delegation. In *Workshop on RFID Security – RFIDSec'11*, Amherst, Massachusetts, USA, June 2011.

[8] J. Ha, S.-J. Moon, J. M. G. Nieto, and C. Boyd. Low-cost and strong-security rfid authentication protocol. In *EUC Workshops*, pages 795–807, 2007.

[9] S. Kardaş, A. Levi, and E. Murat. Providing Resistance against Server Information Leakage in RFID Systems. In *New Technologies, Mobility and Security – NTMS'11*, pages 1–7, Paris, France, February 2011. IEEE, IEEE Computer Society.

[10] R. Küsters and T. Truderung. Using proverif to analyze protocols with diffie-hellman exponentiation. In *Proceedings of the 2009 22nd IEEE Computer Security Foundations Symposium*, CSF '09, pages 157–171, Washington, DC, USA, 2009. IEEE Computer Society.

[11] P. Lopez. *Lightweight Cryptography in Radio Frequency Identification (RFID) Systems*. PhD thesis, Computer Science Department, Carlos III University of Madrid, 2008.

[12] M. Ohkubo, K. Suzuki, and S. Kinoshita. Cryptographic Approach to "Privacy-Friendly" Tags. In *RFID Privacy Workshop*, MIT, Massachusetts, USA, November 2003.

[13] S. Older and S.-K. Chin. Formal methods for assuring security of protocols. *Comput. J.*, 45(1):46–54, 2002.

[14] P. Ryan and S. Schneider. *The modelling and analysis of security protocols: the csp approach*. Addison-Wesley Professional, first edition, 2000.

[15] B. Song and C. J. Mitchell. Scalable RFID Security Protocols supporting Tag Ownership Transfer. *Computer Communication, Elsevier*, March 2010.

[16] S. Vaudenay. On privacy models for rfid. In *Proceedings of the Advances in Crypotology 13th international conference on Theory and application of cryptology and information security*, ASIACRYPT'07, pages 68–87, Berlin, Heidelberg, 2007. Springer-Verlag.

[17] T.-C. Yeh, C.-H. Wu, and Y.-M. Tseng. Improvement of the rfid authentication scheme based on quadratic residues. *Computer Communications*, 34(3):337 – 341, 2011.

[18] K. Y. Yu, S.-M. Yiu, and L. C. K. Hui. Rfid forward secure authentication protocol: Flaw and solution. In *CISIS'09*, pages 627–632, 2009.