A New Framework for Privacy of RFID Path Authentication

Shaoying Cai[1], Robert H. Deng[1], Yingjiu Li[1], and Yunlei Zhao[2]

[1] Singapore Management University, 80 Stamford Road, Singapore 178902
{shaoyingcai.2009,robertdeng,yjli}@smu.edu.sg
[2] Fudan University, No. 825 Zhangheng Road, Shanghai, China 201203
ylzhao@fudan.edu.cn

Abstract. RFID-based path authentication enables supply chain managers to verify the exact path that a tag has taken. In this paper, we introduce a new oracle Move that models a tag's movement along a designed or an arbitrary path in a supply chain. With this oracle, we refine the existing security and privacy notions for RFID-based path authentication. In addition, we propose a new privacy notion, called path privacy, for RFID-based path authentication. Our privacy notion captures the privacy of both tag identity and path information in a single game. Compared to existing two-game based privacy notions, it is more rigorous, powerful, and concise. We also construct a new path authentication scheme. Our scheme does not require the entities in a supply chain to have any connection with each other except in the initial stage. It requires only 480 bits storage and no computational ability on each tag; thus it can be deployed on the standard EPCglobal Class 1 Generation 2 tags in the market.

1 Introduction

Supply chain is a network of multiple parties such as suppliers, transporters, storage facilities, distributors, and retailers that participate in the production, delivery, and sale of product [5]. It is difficult to monitor a supply chain since the involving parties are distributed at multiple locations or even across countries. So that supply chains are vulnerable to the counterfeiting problem, where an adversary injects fake goods into a supply chain. The counterfeiting problem has become a major threat to supply chains. According to the 2011 report of International Chamber of Commerce, it is estimated that the counterfeiting accounts for 5-7% of world trade, or about 600 billion U.S. dollars per year [6]. The ratio of counterfeiting is even higher in luxury market.

Radio Frequency IDentification (RFID) technology has been recently used to facilitate real-time monitoring of supply chains so as to thwart counterfeiting threats. In general RFID-enabled supply chains, each item is attached with a tag. The tag stores identity information of the item. A reader identifies an

F. Bao, P. Samarati, and J. Zhou (Eds.): ACNS 2012, LNCS 7341, pp. 473–488, 2012.

item through the interaction with the corresponding tag. Various tag authentication schemes (e.g.[2,5,8,9,10,11,12,13]) have been proposed to enable privacy-preserving identification of tags.[1] However, most of proposals require tags to have certain computational capability, which may incur unbearable cost in practice. Another common problem of deploying existing solutions in supply chain is that: to monitor a supply chain, the manager should have access to all the databases of the entities in the supply chain. This requires high-quality network connection and fine-grained access control, which may not be realistic in practice.

Recently, RFID-enabled path authentication was proposed by Blass, Elkhiyaoui and Molva [3,4], and extended later to be more practical [6,14], to tackle the counterfeiting problem in supply chains. In the proposal, which is named as TRACKER, the manager of a supply chain verifies the genuineness of tag by checking whether it has been processed by a series of reliable readers. Compared to the existing tag authentication schemes [2,5,8,9,10,11,12,13], the verification of a tag's genuineness is merely based on the credentials stored on the tag about the readers that have processed the tag along the path. TRACKER can be implemented with standard EPCglobal C1 G2 tags, which has several hundred bits storage and no computational ability. It does not require the entities in the supply chain to have any connection except in the initial stage. In this paper, we refine the privacy notions for path authentication and propose a more practical path authentication scheme. Our contributions include:

- We analyze the existing security and privacy notions for path authentication in RFID-enabled supply chain, including tag unlinkability and step unlinkability. We show that these two notions can be further refined to be more concise and formal.
- We propose a combined privacy notion that considers both tag unlinkability and step unlinkability for RFID-enabled supply chains. We analyze the relations among our new privacy notion, the tag unlinkability notion and the step unlinkability notion. We prove that our privacy notion implies tag unlinkability and step unlinkability.
- We propose a new path authentication solution using the standard EPC Class 1 Gen 2 tags without sharing the secret among supply chain parties. Compare to TRACKER, our solution is more efficient and requires less storage. We prove that our solution satisfies the security notion and the privacy notion.

2 Background

First, we model an RFID-enabled supply chain management system and the adversary in the system. Then we refine the security and privacy notions for RFID-enabled path authentication in supply chains.

[1] Most of the existing tag authentication schemes and their extensions are listed on http://www.avoine.net/rfid/

2.1 RFID-Enabled Supply Chain Management System

Supply chain is a network of multiple parties, which can be represented by a digraph $G = (V, E)$, where V is a set of vertices, E is a set of edges. Each vertex $v \in V$ represents one *step* in the supply chain. Note that each supply chain party may conduct several steps to process an item. Each directed edge $e \in E$, $e = \overrightarrow{v_i v_j}$, denotes that v_j is a possible next step to step v_i in the supply chain. A *path* is a finite sequence of steps $P = (v_0, \cdots, v_l)$, where $(v_i, v_{i+1}) \in E$, for $i \in \{0, l-1\}$. Every path shares the same source v_0. The last step v_l of a valid path $P_{valid_i} = (v_0, \cdots, v_l)$ represents a *check point*. Every item enters the supply chain from v_0, and goes through a path according to its own procedure. When it arrives at the check point, the manager will verify the item. Note that if a path consists of an empty set of steps (except v_0), we call it empty path, and denote it as "$-$".

An RFID-enabled supply chain system consists of an issuer I, a set of managers $\mathcal{M}$ and a set of normal readers $\mathcal{R}$. The issuer I is located at the source v_0 of the supply chain; a managers from $\mathcal{M}$ is placed at the end of each valid path and normal readers from $\mathcal{R}$ are placed at other places of a supply chain. The issuer I initializes a tag by storing certain information on the tag. While a tag goes through the supply chain, each reader in its path updates the content of the tag. Eventually, the tag arrives at a manager, the manager reads out the content of the tag and checks the validity of the tag. Formally, the system has the following functions:

- $\mathsf{Initialize}(\kappa)$: Given the security parameter κ, the system prepares a supply chain G, an issuer I and a set of l managers $\mathcal{M}$, a set of m readers $\mathcal{R}$ and a set of n tags $\mathcal{T}$, and a set of ν valid path $\mathcal{P}_{valid}$. We denote the content stored on any tag T_i as state S_{T_i}.
- $\mathsf{Read}(T_i)$: a function that returns back the current internal state S_{T_i} of T_i.
- $\mathsf{Write}(T_i)$: a function that writes a new state S'_{T_i} to T_i. Here we assume that the readers in each step are honest, that is, they update a tag only if the tag is authenticated.
- $\mathsf{PathCheck}(S^j_{T_i})$: a function that verifies whether tag T_i has gone through a valid path P_{valid}. If it is the case, it returns the valid path P_{valid}, else it returns $\emptyset$.

2.2 Adversary Model

We use the following the notations. If $A(\cdot, \cdot, \cdots)$ is a randomized algorithm, then $y \leftarrow A(x_1, x_2, \cdots; \rho)$ means that y is assigned with the unique output of algorithm A on inputs x_1, x_2, $\cdots$ and coins ρ, while $y \leftarrow A(x_1, x_2, \cdots)$ is a shorthand for first picking ρ at random and then setting $y \leftarrow A(x_1, x_2, \cdots)$. $y \leftarrow A^{O_1, \cdots, O_n}(x_1, x_2, ...)$ denotes that y is assigned with the output of algorithm A which takes $x_1, x_2, ...$ as inputs and has oracle accesses to $O_1, ..., O_n$. If S is a set, then $s \in_R S$ indicates that s is chosen uniformly at random from set S. Let $\Pr[E]$ denote the probability that an event E occurs. Let $\mathcal{N}$ denote the set

of all integers. Let $\mathcal{R}$ denote the set of all real numbers. A function $f : \mathcal{N} \to \mathcal{R}$ is said to be *negligible* if for every $c > 0$ there exits a number $n_0 \in N$ such that $f(n) < \frac{1}{n^c}$ holds for all $n > n_0$.

An adversary $\mathcal{A}$, against RFID path authentication, is given accesses to four oracles $\mathcal{O} = \{O_1, O_2, O_3, O_4\}$. O_1, O_2, O_3 denote Read, Write, PathCheck functions, respectively. O_4 denotes a function $\mathsf{Move}(T_i, k, \mathcal{K}, b)$, where $k \in \mathcal{N}$, $\mathcal{K} \in \{P, G\}$, $b \in \{0, 1\}$. Move $(T_i, k, \mathcal{K}, b)$ is defined as follows:

- If $\mathcal{K} = G$, no matter whether $b = 0$ or $b = 1$, starting from the current step of T_i with internal state $S^j_{T_i}$, move the tag T_i forward $k \geq 1$ steps arbitrarily in the supply chain system G.
- If $\mathcal{K} = P$, works as follows: If $b = 1$, from the current step of T_i with internal state $S^j_{T_i}$, move the tag T_i forward $k \geq 1$ steps through the designated path $\mathcal{P}$ (the length of P is at least k steps). If $b = 0$, move tag T_i forward $k \geq 1$ steps according to any path that does not have a common step with P. The reader in each step updates the tag's state. Finally, $\mathsf{Move}(T_i, k, P, b)$ returns back the state transcript $\{S^{j+1}_{T_i}, \cdots, S^{j+k}_{T_i}\}$ of T_i from step $j + 1$ to $j + k$.

Note that oracle O_4 is a new oracle introduced in this paper. It is critical to precisely model various kinds of tag movement. In [4,3], the concept of path is not explicitly defined, and the operations on tag movement are specified through step-level oracles; thus, it is difficult to describe the tag movement at path level. While using O_4, any tag movement can be precisely represented by adjusting the parameters of **Move** function. The introducing of O_4 facilitates defining clear security and privacy notions.

The four oracles capture the adversary's ability to read from a tag, write into a tag, check the validity of a tag, and follow a tag through a designated path P (for the case of $\mathcal{K} = P$) or simply update the state of the tag by forwarding it arbitrarily in the system G (for the case of $\mathcal{K} = G$). We denote by $\mathcal{A}^{\mathcal{O}}(para)$ a probabilistic polynomial-time (PPT) algorithm $\mathcal{A}$ that, on input of some system public parameters *para*, runs a supply chain system via the four oracles in $\mathcal{O}$. An adversary is a (t, n_1, n_2, n_3, n_4)-adversary if it works in time t and makes oracle queries to O_μ without exceeding n_μ times, where $1 \leq \mu \leq 4$.

2.3 Existing Security and Privacy Notions

Security Notion. The security goal of our system is to prevent an adversary from inserting counterfeited goods to the supply chain. As the manager checks the authenticity of a tag merely based on the state stored on a tag, the system should prevent an adversary from forging a tag's internal state with a valid path that has not been actually taken by the tag in the supply chain. Since standard EPC C1 G2 tags have no computation capability, no reader authentication is performed. If a tag's state has been changed by an adversary, even if it has gone through a valid path, it is not considered as a valid tag by a manager.

The security for RFID path authentication means, it is infeasible for any probabilistic polynomial-time adversary $\mathcal{A}$ to create a state $S^l_{T_i}$ for a tag T_i

such that given $S^l_{T_i}$, a manager M outputs a valid path $P_{valid} = \{v_0, \cdots, v_l\}$ which T_i has not gone through. It is formalized by an experiment $\mathbf{Exp}^{\mathbf{Security}}_{\mathcal{A}}[\kappa]$ shown in Figure 1. The adversary $\mathcal{A}$ consists of two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$ which run in two phases, learning phase and challenge phase. Firstly, given parameter κ, the experiment setups the system through $\mathsf{Setup}(\kappa)$, and passes the public system parameters *para* to $\mathcal{A}_1$. In the learning phase, $\mathcal{A}_1$ is allowed to collect information by querying the four oracles without exceeding n_1, n_2, n_3, n_4 times, respectively. Then it generates a transcript *st* which contains the information about the system it gathered during the learning phase. In the challenge phase, $\mathcal{A}_2$ creates a tag T with state s^j_T using *st*. The tag T may have a new ID or an existing ID in the system. Then the game checks the validity of T_i through $\mathsf{Check}(s^j_T)$. $\mathbf{Exp}^{\mathbf{Security}}_{\mathcal{A}}[\kappa]$ outputs 1 if both of the following two conditions hold: $\mathsf{Check}(s^j_T)$ returns a valid path P_{valid}; and there exists $z \in \{1, \ldots, l\}$ such that the tag has not passed v_z in its z-th step, where l denotes the length of the path and v_z denotes the z-th step in P_{valid}. $\mathbf{Exp}^{\mathbf{Security}}_{\mathcal{A}}[\kappa]$ outputs 0, otherwise.

Experiment $\mathbf{Exp}^{Security}_{\mathcal{A}}[\kappa]$
1. run $\mathsf{Setup}(\kappa)$ to setup $I, \mathcal{R}, \mathcal{T}, \mathcal{M}$.
2. $\{st\} \leftarrow \mathcal{A}_1^{\mathcal{O}}(para)$. // the learning phase
3. $T \leftarrow \mathcal{A}_2(st)$. //the challenge phase
4. $s^j_T \leftarrow \mathsf{Read}(T)$.
5. output 1, if $\mathcal{P}_{valid} \longleftarrow \mathsf{PathCheck}(S^j_T)$, and there is a step $v_z \in P_{valid}$ which T has not gone through in its z-th step; output 0, otherwise.

Fig. 1. Security Experiment

Definition 21. *The advantage of $\mathcal{A}$, denoted $Adv^{Security}_{\mathcal{A}}(\kappa)$, in the security experiment is*

$$\left|\Pr[\mathbf{Exp}^{\mathbf{Security}}_{\mathcal{A}}[\kappa] = 1]\right|$$

Definition 22. *We say an RFID path authentication scheme is $(t, n_1, n_2, n_3, n_4, \epsilon)$-secure, if for any t-time adversary $\mathcal{A}$ who makes at most n_1, n_2, n_3, n_4 queries to O_1, O_2, O_3, O_4 respectively, $Adv^{Security}_{\mathcal{A}}(k) < \epsilon$ holds. The probability is taken over coins of $\mathcal{A}$ and the oracles.*

Privacy Notions. For an RFID-enabled supply chain system, Blass, Elkhiyaoui and Molva [3] considered two privacy notions: tag unlinkability and step unlinkability. Tag unlinkability corresponds to the privacy of a tag's identity. Step unlinkability corresponds to the privacy of a tag's path. Note that in the older version of TRACKER [4], there is another path privacy notion, namely path unlinkability, which is proven to be weaker than step unlinkability in [3].

Tag Unlinkability Briefly, tag unlinkability requires that no efficient adversary can link the state information stored in a tag to the tag's identity. In [4], the

tag unlinkability is defined through a formal experiment. The experiment contains two phases: the learning phase and the challenge phase. An adversary $\mathcal{A}$ is provided with two tags T_0 and T_1. In the learning phase, the adversary can access the system and gather information without exceeding the constraints set by the game. In the challenge phase, the game updates the tags by moving them one more step further in the supply chain. The experiment then flips a coin $\delta \in_R \{0,1\}$, and provides the updated state of T_δ to the adversary. The adversary guesses the value of δ. The adversary wins the game if it can successfully guess δ with probability $1/2$ plus a non-negligible quantity.

We slightly modify the experiment to $Exp_{\mathcal{A}}^{Tag-Unlinkability}[\kappa]$. In the learning phase, the adversary is allowed to access the oracles O_1, O_2, O_3, O_4 without exceeding n_1, n_2, n_3, n_4 times, respectively. Then, the adversary outputs two tags T_0 and T_1 together with a transcript st, where st is the information it has gathered. In the challenge phase, the experiment tosses a coin $\delta \in_R \{0,1\}$. The experiment moves the tag T_δ one step forward arbitrarily in the system G, and provides the updated state S_δ of tag T_δ to the adversary. With S_δ and the transcript st, the adversary guesses the value of δ, then outputs the guessed value δ'. If $\delta = \delta'$, the experiment outputs 1; else, the experiment outputs 0. The adversary wins the game if the experiment outputs 1 with probability $1/2$ plus a non-negligible quantity.

A key difference between the original tag unlinkability notion [3] and our refined one is that, in the original notion, the challenge tags T_0 and T_1 are selected by the experiment, while in our notion, the challenge tags T_0 and T_1 are selected by the adversary; therefore, the adversary in our notion is stronger then the adversary in [3]. We depict $Exp_{\mathcal{A}}^{Tag-Unlinkability}[\kappa]$ in Figure 2.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{Tag-Unlinkability}[\kappa]$
1. run $\mathsf{Setup}(\kappa)$ to setup $I, \mathcal{R}, \mathcal{T}, \mathcal{M}$.
   Denote by $para$ the public system parameter.
2. $\{T_0, T_1, st\} \leftarrow \mathcal{A}_1^{\mathcal{O}}(para)$.
3. $\delta \longleftarrow_R \{0,1\}$.
4. $S_\delta \leftarrow$Move$(T_\delta, 1, G, 1)$, i.e., move T_δ one step arbitrarily forward in the system G.
   Denote by $\mathcal{S}_\delta$ the updated state of T_δ.
5. $\delta' \leftarrow \mathcal{A}_2^{\mathcal{O}}(S_\delta, st)$.
6. output 1 if $\delta' = \delta$, 0 otherwise.

Fig. 2. Tag Unlinkability Experiment

Definition 23. *The advantage of $\mathcal{A}$, denoted $Adv_{\mathcal{A}}^{Tag-Unlinkability}(\kappa)$, in the tag unlinkability experiment is* $\left|\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathbf{Tag-Unlinkability}}[\kappa] = 1] - \frac{1}{2}\right|$

Definition 24. *An RFID path authentication scheme is $(t, n_1, n_2, n_3, n_4, \epsilon)$-tag-unlinkable, if for any t-time adversary $\mathcal{A}$ who makes at most n_1, n_2, n_3, n_4 queries to O_1, O_2, O_3, O_4, respectively, we have $Adv_{\mathcal{A}}^{Tag-Unlinkability}(\kappa) < \epsilon$. The probability is taken over the choice of δ, coins of $\mathcal{A}$ and the oracles.*

Step Unlinkability Step unlinkability requires that no efficient adversary is feasible to tell whether the two paths of any two different tags share a common step or not. In [3], the step unlinkability game is defined as follows. Firstly, the experiment randomly chooses a tag T for the adversary. In the learning phase, the adversary arbitrarily queries the oracle without exceeding the constraints. The adversary may gather information from the system. It may follow T, so that it knows the path of the targeted tag. In the challenge phase, the experiment provides the adversary with another tag T_c, the adversary lets T_c move forward along its path for several steps and then reads the state of T_c. Finally, the adversary is asked to guess whether T and T_c have a step in common besides v_0. The adversary breaks the path privacy if the probability of correct guessing is non-negligibly more than $\frac{1}{2}$.

The above experiment defined in [4] is based on the assumption that every tag passes through every step with the same probability. However, given a tag, in case that the probabilities of the tag to pass by different steps are not even, then, an adversary can trivially win the game. We give an example to illustrate the situation. Suppose there are four paths in the system, P_a, P_b, P_c and P_d and every tag will go through the fours paths with equal probability. P_a, P_b, P_c shares a common step v besides v_0, while P_d have no common step with the other three paths besides v_0. In case that the adversary learns that tag T has gone through path P_a, for any T_c the probability that it has a common step v with T is 75%. Thus the adversary will win the game with non-trivial advantage.

We modify the step unlinkability experiment to make it more rigorous. The new step unlinkability experiment $Exp_{\mathcal{A}}^{Step-Unlinkability}[\kappa]$ is shown in Figure 3. The experiment starts by setting the system $I, \mathcal{R}, \mathcal{T}, \mathcal{M}$ through $\mathsf{Setup}(\kappa)$. An adversary $\mathcal{A}$ runs two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively in the two phases. In the learning phase, $\mathcal{A}_1$ queries the oracle set $\mathcal{O}$ and outputs a tag T and transcript st. In the challenge phase, the experiment creates a new tag T_c, and then tosses a coin $\delta \in_R \{0,1\}$. The experiment sets a path P as follows: if $\delta = 0$, the path P does not have any common step with T's path; else the path P have certain common steps with T's path. After getting the path P, the experiment moves T_c along path P in k steps. $\mathcal{A}_2$ reads the state S_{T_c} of T_c, guesses the value of δ, and outputs the guessed value δ'. Note that S_{T_c} contains the states updated by the readers in path P. If the probability of $\delta' = \delta$ is non-negligibly more than $\frac{1}{2}$, the adversary wins the game.

Definition 25. *The advantage of $\mathcal{A}$, denoted $Adv_{\mathcal{A}}^{Step-Unlinkability}(k)$, in the step unlinkability experiment is* $\left|\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathbf{Step-Unlinkability}}[\kappa] = 1] - \frac{1}{2}\right|$

Definition 26. *An RFID path authentication scheme is $(t, n_1, n_2, n_3, n_4, \epsilon)$-step unlinkable, if for any t-time adversary $\mathcal{A}$ who makes at most n_1, n_2, n_3, n_4 queries to O_1, O_2, O_3, O_4, respectively, we have $Adv_{\mathcal{A}}^{Step-Unlinkability}(k) < \epsilon$. The probability is taken over the choice of δ, coins of $\mathcal{A}$ and oracles.*

Experiment $\mathbf{Exp}_{\mathcal{A}}^{Step-Privacy}[\kappa]$
1. run $\mathsf{Setup}(\kappa)$ to setup $T, \mathcal{R}, \mathcal{T}, \mathcal{M}$. Denote by $para$ the public system parameter.
2. $\{T, k, st\} \leftarrow \mathcal{A}_1^{\mathcal{O}}(para)$.
3. create a new tag T_c.
4. randomly selects a bit $\delta \in \{0, 1\}$.
5. if $\delta = 0$, selects a path P that dose not have any common step with T' path; else, select a path P that has one or more common steps with T's path. The length of the path is at least k.
6. $S_{T_c} \leftarrow$Move$(T_c, k, P, 1)$.
7. $\delta' \leftarrow \mathcal{A}_2(S_{T_c}, st)$.
8. output 1 if $\delta' = \delta$, 0 otherwise.

Fig. 3. Step Unlinkability Experiment

3 A New RFID Privacy Notion for Path Authentication

In this section, we propose a new privacy notion, named path privacy, for path authentication. This notion captures the privacy of tag identity and path information in a single game. We show that path privacy implies tag unlinkability and step unlinkability.

3.1 Path Privacy

In [3], two privacy notions, tag-unlinkability and step-unlinkability, should be used together to analyze the privacy of a path authentication scheme. These two notions are formulated separately (via four algorithms). We present a single game-based privacy notion, path-privacy, which implies tag unlinkability and step unlinkability.

The experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathbf{Path-Privacy}}[\kappa]$ of path privacy is shown in Figure 4 and formalized as follows. The experiment consists of two phases: the learning phase and the challenge phase. An adversary $\mathcal{A}$ runs two algorithms $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively in the two phases. The experiment sets up the system $I, \mathcal{R}, \mathcal{T}, \mathcal{M}$ through $\mathsf{Setup}(\kappa)$. In the learning phase, $\mathcal{A}_1$ queries the four oracles without exceeding n_1, n_2, n_3, n_4 times, respectively. $\mathcal{A}_1$ outputs two tags T_0, T_1, a path P that has at least k steps left for both tags, and state information st. In the challenge phase, the experiment firstly flips a coin δ. If $\delta = 1$, the experiment moves T_1 k steps along the path P, and T_1 is updated by k readers in the path. Let the state of T_1 be denoted as S_1. If $\delta = 0$, the experiment moves T_0 k steps without going through the path P (T_0 is updated by k readers that are not in the path). Let the state of T_0 be denoted as S_0. The Move operations are performed by the game challenger, and the adversary has no access to the readers and the tag during the Move operations. In the challenge phase, the experiment provides $\mathcal{A}_2$ with S_δ and st. $\mathcal{A}_2$ guesses the value of δ as δ'. If $\delta' = \delta$, the experiment outputs 1; else the experiment outputs 0. If the experiment outputs 1 with probability non-negligibly more than $\frac{1}{2}$, the adversary wins the game.

Experiment $\mathbf{Exp}_{\mathcal{A}}^{Path-Privacy}[\kappa]$
1. run $\mathsf{Setup}(\kappa)$ to setup $I, \mathcal{R}, \mathcal{T}, \mathcal{M}$. Denote by *para* the public system parameter.
2. $\{T_0, T_1, P, k, st\} \leftarrow \mathcal{A}_1^{\mathcal{O}}(para)$, where P is a path of length at least k, st is state information.
3. $\delta \leftarrow \{0, 1\}$.
4. $S_\delta \leftarrow$Move(T_δ, k, P, δ). Denote by $\mathcal{S}_\delta$ the state of $\mathcal{T}_\delta$.
5. $\delta' \leftarrow \mathcal{A}_2^{\mathcal{O}}(S_\delta, st)$.
6. output 1 if $\delta' = \delta$, 0 otherwise.

Fig. 4. Path Privacy Experiment

Definition 31. *The advantage of $\mathcal{A}$, denoted $Adv_{\mathcal{A}}^{Path-Privacy}(k)$, in the path privacy experiment is* $\left|\Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathbf{Path-Privacy}}[\kappa] = 1] - \frac{1}{2}\right|$

Definition 32. *A RFID path authentication scheme is $(t, n_1, n_2, n_3, n_4, \epsilon)$-private, if for any t-time adversary $\mathcal{A}$ who makes at most n_1, n_2, n_3, n_4 queries to O_1, O_2, O_3, O_4, respectively, we have $Adv_{\mathcal{A}}^{Privacy}(k) < \epsilon$. The probability is taken over the choice of δ, coins of $\mathcal{A}$ and oracles.*

3.2 Relations among Privacy Notions

Now, we analyze the relations among our new privacy notion and the two existing privacy notions. We show that path-privacy is stronger than tag unlinkability and step unlinkability.

Theorem 1. *Path-privacy implies tag unlinkability.*

Proof. Path privacy implies that S_0 and S_1 in the path privacy experiment are computationally indistinguishable, even if the adversary $\mathcal{A}$ has full control over the supply chain system via the four oracle access *except that the random bit δ is blinded to $\mathcal{A}$*. Intuitively, tag unlinkability is implied by path privacy, as the ability of linking tag's state to tag's identity can be *directly* used to break path privacy.

In more details, we show that it is possible to construct an adversary $\mathcal{B}$ that $(t, n_1, n_2, n_3, n_4, \epsilon)$-breaks path privacy using $\mathcal{A}$ as a subroutine, where $\mathcal{A}$ is an adversary which can $(t, n_1, n_2, n_3, n_4, \epsilon)$-break tag unlinkability. Adversary $\mathcal{B}$ plays the path privacy game using adversary $\mathcal{A}$ as a subroutine; it is $\mathcal{A}$ who conducts the attacks to the system, while A aims to win the tag-unlinkability game. Firstly, $\mathbf{Exp}_{\mathcal{B}}^{\mathbf{Path-Privacy}}[\kappa]$ sets up the system $I, \mathcal{R}, \mathcal{T}, \mathcal{M}$ and publishes the public system parameter *para*. Then $\mathcal{B}$ passes *para* to $\mathcal{A}$. $\mathcal{A}$ plays the tag-unlinkability game. In the learning phase, when $\mathcal{A}_1$ queries the oracles $\mathcal{O}$, the queries are transferred to $\mathcal{B}_1$, and $\mathcal{B}_1$ queries the oracles $\mathcal{O}$ for $\mathcal{A}_1$ in the path-privacy experiment. Then $\mathcal{A}_1$ outputs $\{T_0, T_1, st\}$. Upon receiving $\mathcal{A}_1$'s output, $\mathcal{B}_1$ chooses a path P and submits $\{T_0, T_1, P, 1, st\}$ to the path-privacy experiment. The experiment $\mathbf{Exp}_{\mathcal{B}}^{\mathbf{Path-Privacy}}[\kappa]$ chooses $\delta \in_R \{0, 1\}$, and returns

$S_\delta \leftarrow \text{Move}(\mathcal{T}_\delta, 1, P, 1)$ to $\mathcal{B}_2$. $\mathcal{B}_2$ transfers S_δ to $\mathcal{A}_2$. When $\mathcal{A}_2$ stops, $\mathcal{B}_2$ outputs whatever output by $\mathcal{A}_2$. It is clear that if $\mathcal{A}$ wins the tag unlinkability game, then $\mathcal{B}$ wins the path privacy game. We have:

$$\Pr[\mathbf{Exp}_{\mathcal{B}}^{\mathbf{Path-Privacy}}[\kappa] = \mathbf{1}] = \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathbf{Tag-Unlinkability}}[\kappa] = \mathbf{1}] \quad (1)$$

If $\mathcal{A}$ $(t, n_1, n_2, n_3, n_4, \epsilon)$-breaks tag-unlinkability, then $\mathcal{B}$ also $(t, n_1, n_2, n_3, n_4, \epsilon)$-breaks path privacy. □

Theorem 2. *Path privacy implies step unlinkability.*

Proof. Assuming that a system is not step-unlinkable, there exists an adversary $\mathcal{A}$ which can $(t, n_1, n_2, n_3, n_4, \epsilon)$-break its step unlinkability. We can construct an adversary $\mathcal{B}$ that breaks the path privacy using $\mathcal{A}$ as a subroutine.

$\mathbf{Exp}_{\mathcal{B}}^{\mathbf{Path-Privacy}}[\kappa]$ sets up the system $I, \mathcal{R}, \mathcal{T}, \mathcal{M}$ and publishes the public system parameter *para*. $\mathcal{B}$ passes *para* to $\mathcal{A}$. If $\mathcal{A}$ can break the step unlinkability in $\mathbf{Exp}_{\mathcal{A}}^{\mathbf{Step-Unlinkability}}[\kappa]$. Then $\mathcal{B}$ can use $\mathcal{A}$ as a subroutine to break path-privacy. In the learning phase, $\mathcal{A}_1$ gathers the information of the system. In this process, $\mathcal{A}_1$ cannot query the oracles directly; instead, it submits the queries to $\mathcal{B}_1$ and then $\mathcal{B}_1$ queries the oracles $\mathcal{O}$ for $\mathcal{A}_1$. Then $\mathcal{A}_1$ outputs $\{T, st\}$. As $\mathcal{A}_1$ fully controls the system during the learning phase, then $\mathcal{A}_1$ knows the path of T. We denote the path by P, which is contained in st. Then $\mathcal{A}_1$ passes $\{T, k, st\}$ to $\mathcal{B}_1$. $\mathcal{B}_1$ creates two new tags T_0 and T_1 and outputs $\{T_0, T_1, P, k, st\}$. $\mathbf{Exp}_{\mathcal{B}}^{\mathbf{Path-Privacy}}[\kappa]$ tosses a coin δ. If $\delta = 0$, then the experiment moves T_0 without going through path P in k step, and the state of T_0 is denoted as S_0; else, the experiment moves T_1 through path P in k step, and the state of T_1 is denoted as S_1. The experiment returns S_δ to $\mathcal{B}_2$. $\mathcal{B}_2$ transfers S_δ to $\mathcal{A}_2$, and outputs whatever output by $\mathcal{A}_2$.

In the above path-privacy game, $\mathcal{B}_2$ is provided with the state S_δ. If $\delta = 0$, then the tag with state S_0 does not have any common step with T. If $\delta = 1$, then the tag with state S_1 has at least one common step with T. Given S_δ, $\mathcal{A}_2$ guesses whether the tag has a common step with T or not. $\mathcal{B}_2$ can directly use the result of $\mathcal{A}_2$. It is clear that:

$$\Pr[\mathbf{Exp}_{\mathcal{B}}^{\mathbf{Path-Privacy}}[\kappa] = \mathbf{1}] = \Pr[\mathbf{Exp}_{\mathcal{A}}^{\mathbf{Step-Unlinkability}}[\kappa] = \mathbf{1}] \quad (2)$$

Hence, if $\mathcal{A}$ $(t, n_1, n_2, n_3, n_4, \epsilon)$-breaks the step-unlinkability, then $\mathcal{B}$ also $(t, n_1, n_2, n_3, n_4, \epsilon)$-breaks the path privacy. □

4 A New RFID Path Authentication Protocol

We propose a new RFID-based path authentication scheme under the path privacy notion. Our path authentication scheme is suitable for a supply chain that where the paths of products are pre-determined. We use pseudorandom function and elliptic curve ElGamal encryption scheme as building blocks.

4.1 Building Blocks

Pseudorandom function Given a security parameter κ, let $m(\cdot)$ and $l(\cdot)$ be two positive polynomials in κ. We say that

$$\{F_k : \{0,1\}^{m(\kappa)} \longrightarrow \{0,1\}^{l(\kappa)}\}_{k \in_R \{0,1\}^\kappa} \tag{3}$$

is a PRF ensemble if the following two conditions hold:

1. Efficient evaluation: There exists a polynomial-time algorithm that on input k and $x \in \{0,1\}^{m(\kappa)}$ returns $F_k(x)$.
2. Pseudorandomness: A PPT oracle machine $\mathcal{A}(t, \varepsilon)$-breaks the PRF ensemble, if
$$|Pr[\mathcal{A}^{F_\kappa}(\kappa) = 1] - Pr[\mathcal{A}^{H_\kappa}(\kappa) = 1]| \geq \varepsilon \tag{4}$$
where F_κ is a random variable uniformly distributed over the multi-set $F_k, k \in_R \{0,1\}^\kappa$, H_κ is uniformly distributed among all functions mapping $m(\kappa)$-bit-long strings to $l(\kappa)$-bit-long strings, and the running time of $\mathcal{A}$ is at most t (here each oracle query accounts for one unit operation).

The PRF ensemble is pseudorandom, if for all sufficiently large κ, there exists no algorithm A that can (t, ϵ)-break the PRF ensemble, for any t that is polynomial in κ and any ϵ that is nonnegligible in κ [15].

Elliptic Curve ElGamal Cryptosystem An elliptic curve ElGamal cryptosystem provides the following usual set of operations:

- *Setup*: The system outputs an elliptic curve $\mathcal{E}$ over a finite field $\mathbb{F}_p$, where p is a large prime. Let P be a point on $\mathcal{E}(\mathbb{F}_p)$ of a large prime order q such that the discrete logarithm problem is intractable for $\mathcal{G} =< P >$.
- *Key generation*: The secret key is $sk \in \mathbb{F}_p$. The corresponding public key pk is the pair of points $(P, Y = sk \cdot P)$.
- *Encryption*: To encrypt a point $M \in \mathcal{E}$, one randomly selects $r \in \mathbb{F}_q$ and computes $E(M) = (U, V) = (r \cdot P, M + r \cdot Y)$. The ciphertext is $c = (U, V)$.
- *Decryption*: To decrypt a ciphertext $c = (U, V)$, one computes $D(c) = U - sk \cdot V = M$.

To encrypt message m, we need a point mapping algorithm to transform $m \in \mathbb{F}q$ to a point in the elliptic curve $\mathcal{E}$. $\mathcal{M}(m) = m \cdot P$ is a simple additively homomorphic and unreversed mapping $\mathcal{M} : \mathbb{F}_q \rightarrow \mathcal{E}$, where P is a point in $\mathcal{E}$ of large prime order q. This mapping is a one-to-one mapping from $\mathbb{F}_q$ to $G =< P >$: if $\exists m_1, m_2 \in \mathbb{F}_q$ such that $\mathcal{M}(m_1) = \mathcal{M}(m_2)$, then $m_1 = m_2 \bmod q$.

ElGamal system supports re-encryption operation denoted as ReE. Given a ciphertext $c = (U, V)$ under a public key $pk = (P, Y = sk \cdot P)$, and the public key pk, ReE re-randomizes the ciphertext c to c', where $c' = (U', V') = (U + r \cdot P, V + r \cdot Y)$, for $r \in_R \mathbb{F}_q$. ElGamal system preserves the semantic security property under re-encryption [7]. Let $O_{re-encrypt}$ be an oracle that, provided with two ciphertexts c_0, c_1, randomly chooses $b \in \{0,1\}$, re-encrypts c_b using ElGamal and public key pk, and returns the resulting ciphertext c_b. The semantic security of ElGamal under re-encryption implies that guessing the value of b is as difficult for $\mathcal{A}$ as the decisional Diffie-Hellman (DDH) problem [7].

4.2 Protocol

Assume that an RFID-enabled supply chain path authentication system consists of a set of n tags, an issuer I, a set of l managers $\mathcal{M}$, and a set of m normal readers $\mathcal{R}$. Our protocol has three steps: initialization, updating and verification. In the initialization step, the issuer and the managers setup the system together and initialize the tags. When the tags enter the supply chain, the corresponding reader updates the tags on each step. Finally, when a tag reaches a manager in $\mathcal{M}$, the manager reads out the content of the tag and checks the validity of the tag. Each tag stores an encrypted ID and an encrypted credential generated by the readers in its path.

Initialization: The managers $\mathcal{M}$ generate $(sk, pk) = (x, y = g^x)$ for ElGamal encryption and send pk to the issuer and the readers. The underlying elliptic curve of the ElGamal system is denoted as $\mathcal{E}$. The issuer $\mathcal{I}$ selects a secret-key $k_0 \in \{0,1\}^\kappa$, where κ is the system parameter. $\mathcal{I}$ sets for each reader R_j a secret key k_j, where $1 \leq j \leq m$. $\mathcal{I}$ distributes k_j to R_j. The issuer selects a pseudorandom function PRF, and sends PRF to all the normal readers.

Each tag T_i has an unique identity ID_i, where $ID_i \in \mathcal{E}$. For each T_i, the issuer $\mathcal{I}$ sets its initial state to be $\{c_i = E(ID_i), t_i = PRF_{k_0}(ID_i)\}$. We denote the path which T_i will go through as P_i. Suppose $P_i = (R_{i_0}, R_{i_1}, R_{i_2}, \cdots, R_{i_l})$, for any $0 \leq j \leq l$, where i_j denotes the reader ID in the position j of path P_i. Then for T_i, the issuer $\mathcal{I}$ computes $v_i = PRF_{k_{i_l}}(PRF_{k_{i_{l-1}}}(\cdots(PRF_{k_0}(ID_i))))$, and stores a copy of (ID_i, v_i) on the databases of the managers $\mathcal{M}$.

Interaction between Reader and Tag: When tag T_i reaches R_j, reader R_j reads out T_i's current state $S_{T_i} = \{c_i, t_i\}$. R_j computes the new state $\{c_i', t_i'\}$, where c_i' is re-randomization of c_i under the public key pk and $t_i' = PRF_{k_j}(t_i)$, and then writes $\{c_i', t_i'\}$ to the tag.

Check the Validity of Tag: Only the managers $\mathcal{M}$ can check the validity of tags. Upon the arrival of a tag at a check point, with state $\{c_i, t_i\}$, M decrypts c_i to get ID_i, and searches its database; if and only if it can find a tuple (ID_i, v_i) that satisfies $t_i = v_i$, then T_i is considered as a valid tag.

4.3 Security and Privacy Analysis

The security and privacy of the proposed protocol are based on the pseudorandomness of PRF and the semantic security of Elgamal Encryption scheme under re-encryption. In the following, we provide a formal security and privacy analysis.

Suppose PRF is a pseudorandom function that mapps $m(\kappa)$-bit-long strings to $l(\kappa)$-bit-long strings. We call the function $CPRF(m) = PRF_{k_l}(PRF_{k_{l-1}}(\cdots(PRF_{k_0}(m)))$ as "cascaded" pseudorandom function, where $k_0, ..., k_l$ are randomly chosen keys for the pseudorandom function PRF. If for all k_i, $0 \leq i \leq l$, PRF_{k_i} is a pseudorandom function, $CPRF(m)$ is a pseudorandom function (formal proof please refer to [4]).

Lemma 1. *Producing a new valid pair $\{c_i, t_i\}$ contradicts with the pseudorandomness property of $CPRF$. Here a new pair of $\{c_i, t_i\}$ means that c_i is a ciphertext of a new ID_i under the public key of the system, or c_i is a ciphertext of an existing ID_i in the system while t_i is a new value that has not appeared in the system.*

Proof (sketch). The security of our system is based on the pseudorandomness of $CPRF(m)$. Suppose there is an oracle $O_{CPRF}^{distinguish}$, given a message m, the oracle randomly returns the value of $CPRF(m)$ or $H(m)$, denoted as m', where $H()$ is an arbitrarily selected function among all functions mapping $m(\kappa)$-bit-long strings to $l(\kappa)$-bit-long strings. After getting m', the adversary outputs 1 if it guesses $m' = CPRF(m)$, else he outputs 0. $Pr[\mathcal{A}^{CPRF}(\kappa) = 1]$ denotes the probability that the adversary outputs 1 when the oracle $O_{CPRF}^{distinguish}$ returns value $CPRF(m)$. $Pr[\mathcal{A}^{H}(\kappa) = 1]$ denotes the probability that the adversary outputs 1 when the oracle $O_{CPRF}^{distinguish}$ returns value $H(m)$. Since $CPRF$ is a pseudorandom function, given $\mathcal{A}$ with limited access to the function $CPRF$, we have $|\Pr[\mathcal{A}^{CPRF}(\kappa) = 1] - \Pr[\mathcal{A}^{H}(\kappa) = 1]| \geq \varepsilon$, where ε is negligible. We will show that if an adversary $\mathcal{A}'$ can successfully forge a new pair $\{c_i, t_i\}$, then using $\mathcal{A}'$ as a subroutine, there exists an adversary $\mathcal{A}$ that breaks $CPRF(m)$'s pseudorandomness, namely the value of $|Pr[\mathcal{A}^{CPRF}(\kappa) = 1] - Pr[\mathcal{A}^{H}(\kappa) = 1]|$ will be non-negligible.

$\mathcal{A}$ sets up a supply chain system with public key pk, private key sk for Elgamal encryption system, and a valid path in which the readers have the keys $k_0, \cdots, k_l$, respectively, where l is the length of the path. $\mathcal{A}$ does not know the keys $k_0, \cdots, k_l$, while it is provided with $PRF_{k_0}, \cdots, PRF_{k_i}$ by the oracle $O_{CPRF}^{distinguish}$. $\mathcal{A}$ transfers the public system parameters to $\mathcal{A}'$ which runs two algorithms $\mathcal{A}'_1$ and $\mathcal{A}'_2$ in $\mathbf{Exp}_{\mathcal{A}'}^{\mathbf{Path-Privacy}}[\kappa]$, . In the learning phase, $\mathcal{A}'_1$ accesses the supply chain system without exceeding the constraints defined in $\mathbf{Exp}_{\mathcal{A}'}^{\mathbf{Path-Privacy}}[\kappa]$. In the challenge phase, $\mathcal{A}'_2$ outputs a new pair $\{c_i, t_i\}$. $\mathcal{A}$ decrypts c_i to get ID. Then $\mathcal{A}$ queries $O_{CPRF}^{distinguish}$ with ID. $O_{CPRF}^{distinguish}$ returns a message mes_{ID}. In case $\{c_i, t_i\}$ is valid, then by checking whether $mes_{ID} = t_i$, $\mathcal{A}$ knows whether $O_{CPRF}^{distinguish}$ has chosen the function $CPRF$ or a random function $H()$. As a result, if $\mathcal{A}'$ $(t, n_1, n_2, n_3, n_4, \epsilon)$-breaks the the security of path authentication, then $\mathcal{A}$ (t, ϵ)-breaks the pseudorandomness of function $CPRF$.

Theorem 3. *If PRF is pseudorandom, then our system has path privacy property under the semantic security of ElGamal re-encryption.*

Proof (sketch). Assume that our system is not path private, namely, there exists an adversary $\mathcal{A}$ that breaks the path privacy of our system. Then we can construct an adversary $\mathcal{B}$ to break the semantic security of ElGamal encryption system under re-encryption. $\mathcal{B}$ uses $\mathcal{A}$ as a subroutine and maintains a list L to answer $\mathcal{A}$'s queries as follows.

Suppose the public key of an ElGamal encryption cryptosystem is pk, and its corresponding private key is sk. Adversary $\mathcal{B}$ can break the semantic security of the system under re-encryption. $\mathcal{B}$ firstly simulates a path authentication

system; it initializes the system the same as Initialization step defined in Section 4.2, except that the public and private keys of the manager are set to pk and sk, respectively. Note that $\mathcal{B}$ knows all the secret keys of the readers, but it does not know the value of sk. Then an adversary $\mathcal{A}$ starts the path-privacy experiment. In the learning phase of $\mathcal{A}_1$, when $\mathcal{A}_1$ queries the oracles, $\mathcal{B}$ answers the queries. $\mathcal{B}$ can answer the queries to O_1, O_2 and O_4 directly. However, $\mathcal{B}$ does not have the private key sk, hence in case $\mathcal{A}_1$ queries the O_3 with a state $\{c_i, t_i\}$, $\mathcal{B}$ cannot decrypt c_i to get ID_i and compare the value of t_i with v_i in the database. In order to answer the queries to $\mathcal{O}_3$, $\mathcal{B}$ maintains a list L that records the history of each oracle's operations. Firstly, $\mathcal{B}$ inserts the tuples (ID_i, c_i, t_i, v_i) for $i = 1$ to n into list L. Then, each time a tag's state is changed, $\mathcal{B}$ adds a link between the tag's new state and old state. With the list L, given a tag's state, even through $\mathcal{B}$ cannot decrypt the ciphertext, it can get the tag's ID through the records of the tag's state in list L. Thus $\mathcal{B}$ can answer the queries to $\mathcal{O}_3$ by searching the database and comparing t_i with v_i. At the end of the learning phase, $\mathcal{A}_1$ outputs two tags T_0 and T_1, a path P with no less than k steps, st. Suppose that the state of T_0 is $\{c_0, t_0\}$, the state of T_1 is $\{c_1, t_1\}$. $\mathcal{B}$ firstly submits the two messages $\{c_0, c_1\}$ to $\mathcal{O}_{re-encrypt}$. $\mathcal{O}_{re-encrypt}$ randomly chooses $b \in \{0, 1\}$, and re-encrypts c_b to c'_b under the public key pk. Then $\mathcal{B}$ sends $S = \{c'_b, r\}$ to $\mathcal{A}_2$, where r is a random string. Note that actually, $\mathcal{B}$ should provide $\{c'_b, t_b\}$ to $\mathcal{A}_2$, where t'_b is the new value of t_b after been processed by k readers in path P. We argue that $\{c'_b, r\}$ and $\{c'_b, t'_b\}$ contain same information that can be used by $\mathcal{A}_2$. $\mathcal{A}_2$ cannot get any information from t'_b since the function PRF is a pseudorandom function. $\mathcal{A}_2$ guesses the value of b by analyzing $\{c'_b, r\}$. $\mathcal{B}$ outputs whatever output by $\mathcal{A}$.

Assuming the pseudorandomness of PRF, the advantage of $\mathcal{B}$ to break the semantic security of ElGamal under re-encryption is the same as the advantage of $\mathcal{A}$ to break the path privacy of the system. Since the ElGamal encryption scheme under re-encryption is semantic secure, hence our system is path private. □

4.4 Performance

Computational requirement: Our scheme does not require the tags to perform any computation. All the computation will be performed at the reader side. To update a tag, each reader requires one re-encryption operation and one computation on PRF. For a manager to verify a tag's validity, it requires one decrypting operation and one comparison.

Storage requirement: Each tag T_i's state S_i consists of $\{c_i, t_i\}$. c_i is ElGamal ciphertext on ID_i which requires $2 \cdot 160$ bits. t_i is the path mark, generated by the PRF, thus 160 bits is sufficient. Therefore 480 bits storage is required for each tag. The protocol can thus be implemented with the standard EPC Class 1 Gen 2 tag with an extensible EPC memory bank (scalable between 16-480 bits), a scalable user memory bank (64-512 bits), which are available in the market [1].

On the reader side, the issuer stores a copy of system parameters including pk and k_j, for $0 \leq j \leq m$, m is the number of normal readers. So the

storage requirement for the issuer is $O(1)$. Each normal reader R_j at step v_j needs to store the public key pk of the system and its own key k_j, the storage requirement for each normal reader is $O(1)$. Each manager stores a copy of sk. It also maintains a database DB, for each tag T_i, DB stores the verification information (ID_i, v_i). The storage requirement for a manager is $O(n)$, n is the number of the tags. As a tag's record takes 480 bits, a manager with 1GB storage can stores more than 17 million tags' records.

Compare to TRACKER [3,4], our system is more practical. Since the tags' paths are predetermined in the initial stage, there is no need to store the path information on tag. A manage can perform path verification by simple comparison. Consequently, the storage and computational requirements on updating tags are reduced. The comparisons of storage and computational requirements between our protocol and TRACKER are shown in Table 1. Note that in comparing the computational load, we omit the cheap operations such as hash operation, computing PRF, and point addition on elliptic curve etc. We only count the relative expensive operations such as point multiplication on elliptic curve.

Table 1. Comparisons of TRACKER and Our Protocol

	TRACKER [3]	Our protocol
storage requirement		
tag	960 bits	480 bits
issuer	$O(1)$	$O(1)$
normal reader	$O(1)$	$O(1)$
manager	$O(n + vp)$, vp is the number of valid paths n is the number of tags	$O(n)$, n is the number of tags
computational requirement of processing a tag (operation on elliptic curve)		
issuer	8 point multiplication	2 point multiplication
normal reader	10 point multiplication	2 point multiplication
manager	5 point multiplication	1 point multiplication

5 Conclusions

In this paper, we analyzed the existing security and privacy notions for RFID-enabled path authentication in [3]. We provided refined versions of the notions. We proposed the first single-game-based privacy notion for path authentication which implies the existing notions. We also proposed a path authentication protocol that satisfies the privacy notion. Our protocol can be implemented on standard EPC class 1 Generation 2 tags, and it outperforms the existing path authentication solutions [3,4] for RFID-enabled supply chains.

References

1. `http://www.alientechnology.com/tags/index.php`
2. Arbit, A., Oren, Y., Wool, A.: Toward Practical Public Key Anti-Counterfeiting for Low-Cost EPC Tags. In: IEEE RFID 2011, Orlando, FL, pp. 184–191 (April 2011)

3. Blass, E.O., Elkhiyaoui, K., Molva, R.: Tracker: Security and Privacy for RFID-Based Supply Chains. Cryptology ePrint Archive, Report 2010/219 (2010)
4. Blass, E.O., Elkhiyaoui, K., Molva, R.: Tracker: Security and Privacy for RFID-Based Supply Chains. In: NDSS 2011, San Diego, California, USA, pp. 455–472 (2011)
5. Cai, S., Li, Y., Li, T., Deng, R.H.: Attacks and Improvements to an RFID Mutual Authentication Protocol and its Extensions. In: WiSec 2009, Zurich, Switzerland, pp. 51–58 (2009)
6. Cai, S., Li, Y., Zhao, Y.: Distributed Path Authentication for Dynamic RFID-Enabled Supply Chains. In: IFIP SEC 2012, Crete, Greece (2012)
7. Golle, P., Jakobsson, M., Juels, A., Syverson, P.: Universal Re-encryption for Mixnets. In: Okamoto, T. (ed.) CT-RSA 2004. LNCS, vol. 2964, pp. 163–178. Springer, Heidelberg (2004)
8. Li, Y., Ding, X.: Protecting RFID Communications in Supply Chains. In: ASIACCS 2007, New York, NY, USA, pp. 234–241 (2007)
9. Molnar, D., Wagner, D.: Privacy and Security in Library RFID: Issues, Practices, and Architectures. In: CCS 2004, New York, NY, USA, pp. 210–219 (2004)
10. Peris-Lopez, P., Hernandez-Castro, J.C., Tapiador, J.M.E., Li, T., Li, Y.: Vulnerability Analysis of RFID Protocols for Tag Ownership Transfer. Computer Networks 54(9), 1502–1508 (2010)
11. Piramuthu, S.: RFID Mutual Authentication Protocols. Decision Support Systems (2010), http://dx.doi.org/10.1016/j.dss.2010.09.005
12. Rizomiliotis, P., Rekleitis, E., Gritzalis, S.: Security Analysis of the Song-Mitchell Authentication Protocol for Low-Cost RFID Tags. IEEE Communications Letters 13(4), 274–276 (2009)
13. Song, B., Mitchell, C.J.: RFID Authentication Protocol for Low-Cost Tags. In: WiSec 2008, Alexandria, Virginia, USA, pp. 140–147 (2008)
14. Wang, H., Li, Y., Zhang, Z., Cao, Z.: Two-Level Path Authentication in EPCglobal Network. In: IEEE RFID 2012, Orlando, Florida, pp. 24–31 (2012)
15. Yao, A.C., Yung, M., Zhao, Y.: Adaptive Concurrent Non-Malleability with Bare Public-Keys. CoRR, abs/0910.3282 (2009)