11-2016

Efficient Tag Path Authentication Protocol with Less Tag Memory

Hongbing WANG

Yingjiu LI
Singapore Management University, yjli@smu.edu.sg

Zongyang ZHANG

Yunlei ZHAO

Efficient Tag Path Authentication Protocol with Less Tag Memory

No Author Given

No Institute Given

Abstract. Logistical management has been advanced rapidly in these years, taking advantage of the broad connectivity of the Internet. As it becomes an important part of our lives, it also raises many challenging issues, e.g., the counterfeits of expensive goods pose a serious threat to supply chain management. As a result, path authentication becomes especially important in supply chain management, as it helps us maintain object pedigree and supply chain integrity. Meanwhile, a tag path authentication must meet a series of security requirements, such as authentication, privacy, and unlinkability. In addition, the authentication protocol must be efficient.

In 2011, the first tag path authentication protocol in an RFID-based supply chain, named "Tracke", is proposed by Blass *et al.* in NDSS'11. They have made an important breakthrough in this research area. In this paper, we improve their work and propose a more efficient tag path authentication protocol in an RFID-based supply chain, which meets all the above mentioned security requirements. Our result shows that the proposed protocol can significantly reduce both computational overhead and memory requirement on tags, compared with the previous work.

Keywords: RFID, Tag path authentication, Security and privacy,Unlinkability.

1 Introduction

With the help of radio-frequency identification (RFID) system, object identification and tracking can be easily achieved in a supply chain. An object's identification is stored in a tag which is embedded in an object. The tag can be interrogated by a tag reader via a wireless communication channel in an RFID-based system. As a result, the location of an object and its shipping path can be tracked. The system with tracking capability has been widely adopted in supply chain management. Both participants and beneficiaries of a supply chain concern the genesis of an object,and whether an object is being cloned in conveyance in a supply chain. In todays RFID applications, one of the most challenging problems is tag security and privacy. Considering the limited memory of a tag and its lack of computing capability, to develop an efficient path authentication protocol has always been regarded as a challenging topic.

Currently,logistic management is mainly represented by RFID-based supply chain management, and it is widely adopted and becomes an important part of our daily lives. RFID technology brings convenience to logistic network, and it

has been widely used in numerous applications,including manufacturing, logistics, transportation, warehouse inventory control, supermarket checkout counters, etc. [8]. As forecasted,the overall RFID market will pass 6 billion in 2011 [1]. It not only covers traditional applications such as access control, automobile immobilization, and electronic toll collection, but also includes emerging applications such as animal ID, asset management, baggage handling, cargo tracking/security, contactless payment and ticketing,real-time locating systems,and supply chain management. In a supply chain,participants mainly concern the issues of anti-counterfeiting,anti-cloning,and replica-prevention of luxury products or pharmaceutics [21,15,4], healthcare [11], mobile device [10]. However, we cannot effectively track or monitor object movements in a supply chain, since an adversary can inject fake objects into the supply chain,which eventually hurts sellers and purchasers. In the supply chain management, path authentication is especially important for guaranteeing object genuineness by maintaining object pedigree and supply chain integrity.

Security and privacy are the two important issues for RFID-based supply chain systems [19,22,17,14,18,3,8,4,5]. For security property, a path authentication protocol must be able to verify if an object has taken one of the valid paths through supply chain. For privacy property, a path authentication solution should prevent adversaries from identifying, tracing, or linking tags in a supply chain. Because RFID tags are usually passive entities which have limited memory and almost no computation capability, it is thus very challenging to design a protocol which is efficient and is able to meet security and privacy requirements. Moreover,RFID-based supply chain has been a very active research area in recent years, and has attracted a lot of attention in the past years in both industry and academic, partially due to its broad deployment for automated oversight of supply chain by many large organizations, such as WalMart, Procter and Gamble, and the United States Department of Defense [16,13].

1.1 Related Work

There are already many path authentication protocols in network literature, including protocols for routing, protocols in wireless sensor network, and secure border gateway protocols [12,6,7,23,20]. However,these protocols are mostly implemented among computers or sensors with considerable computation capabilities. The minimum requirement of these protocols is participants with some computation capabilities. Thus,they are not suitable for path authentication in an RFID-based supply chain, where we assume that tags have no computation capability at all.

The first real solution for tag path authentication in an RFID-based supply chain was proposed by Blass *et.al.* [4] in NDSS'11, named "Tracker". The security and privacy of their protocol are based on an extension of polynomial signature techniques for run-time fault detection using homomorphic encryption. In their protocol, an issuer is responsible for the setup of system parameters, including public parameters for the system, and public/private key pairs for all the readers and a manager. Each party keeps his own private key secretly. Since only the

manager can verify a path and validate the path, the manager is equipped with all readerss private keys and his own private key. Meanwhile, the manager owns a valid path set which includes all possible valid paths that a tag may take. Tracker is implemented in an elliptic curve with no requirement on tag computing capability. As claimed by the authors, their solution [4] is the first one available solely based on cheap, non tamper-proof RFID tags.

1.2 Our Contribution

After an extensive study of the previous protocol [4],we are able to further improve their work with more efficient use of computing power and memory. Our improvement is twofold: one is on space memory of tags, and the other is on computational cost. Similar to [4], we use elliptic curve ElGamal-based public key encrypting [9] as the main technique to construct our protocol. However,we reduce the memory size from 6 group elements to 5 group elements, thus the memory space is reduced from 960 bits of the previous work [4] to 800 bits. This is due to the use of a different method to verify the tags path,i.e., we use another randomly re-encrypted element in the group to encrypt a tag and its valid path instead of one group element and HMAC signature [2]. With respect to the computational cost, we do not use HMAC signature in the construction of our protocol. Though the HMAC [2] operation is only a hash function, and it does not need much more computational cost, our protocol is better than the previous one [4] both in multiplication and exponentiation operations. Of the importance, our work is compatible with EPC Class 1 Gen 2 tags.

1.3 Paper Organization

The remainder of this paper is organized as follows. In Section 2, we introduce the preliminary knowledge, including components, assumptions, definitions, and security models. We then describe our efficient tag path authentication protocol in Section 3, followed by security analysis in Section 4. We compare our work with the previous one in computational cost and memory use in Section 5. Finally, we conclude this paper in Section 6

2 Preliminary and Definitions

2.1 Components

There are four entities in our tag path authentication protocol.

Tag T_i: Tags are radio transponders attached to physical objects. A tag has an initial state which is written into it by an issuer, and the state is updated each time when the tag interacts with a reader. The update action represents that the tag proceeds in a supply chain.

Reader R_i: Each reader interacts with numerous tags. It reads out the current state of a tag, and then computes a new state (re-encrypts the current state) for the tag. At last, the new state information is written to the tag. Each reader has its back-end database for computations and storage of some information.

Issuer I: There is only one issuer in the system. The issuer is responsible for the generation of system's public parameters and the public/private key pairs for all the readers and the manager. For a new tag T which is ready to enter a supply chain, the issuer I writes an initial state s_T^0 to T.

Manager M: There is only one manager in the system, which is equipped with all the private keys of readers besides his own one. The manager is the only role who can verify the validation of a path. To verify the validation of a path, he must have a set P_{valid} full of all valid paths, such as P_{valid_i} beforehand.

Similar to [4], our tag path authentication protocol includes four stages: (1) the system initialization stage; (2) the tag preparation stage; (3) the tag and reader interaction stage; and (4) the path verification stage.

2.2 Path Authentication Protocol in An RFID-Based Supply Chain

A path authentication protocol in an RFID-based supply chain typically consists of the following five algorithms:

Setup: It is run by the issuer. The algorithm outputs the system's parameters par, including an elliptic curve $\mathcal{E}$ over a finite field $\mathbb{F}_p$. $\mathcal{E}$ is with a large prime order q such that the discrete logarithm problem is intractable for $\mathcal{G} = \langle g \rangle$, where g is a generator on $\mathcal{E}(\mathbb{F}_p)$. Here, p and q are security parameters with $|p| = |q|$ =160 bit.

KeyGen: It is run by the issuer. The algorithm generates the private keys for all the readers in the system, as well as the private key of the manager.

Enc: It is run by the issuer. When a tag T is ready to enter the supply chain, the issuer computes the initial state s_T^0 of tag on T's identification ID_T using this algorithm. Finally, the issuer writes s_T^0 to the tag T.

ReEnc: It is run by the reader. When a tag T interacts with a reader, the reader reads out the current state s_T^i of T, and re-encrypts s_T^i into a new state s_T^{i+1} of T, where i represents the step the tag T proceeds in the supply chain. Finally, the reader writes s_T^{i+1} to the tag T.

Dec: It is run by the manager. When a tag T with the final state s_T^k arrives at the end point of a supply chain, the manager checks whether T has gone through a valid path specified by the issuer using this algorithm. The algorithm gets the identification of T as well as its path by decrypting s_T^k.

In our protocol, we use the same assumptions as in [4]. These assumptions are summarized as follows:

- A supply chain is represented by a directed diagraph $G = (V, E)$, where V is a set of vertices, and E is a set of edges. Each vertex $v \in V$ is equivalent to one step in the supply chain, and is uniquely associated with a reader R_i[1]. Each directed edge $e \in E$, $e := \overrightarrow{v_i v_j}$, is a representation from vertex v_i to vertex v_j, where v_j is a possible next step from step v_i in the supply chain.
- A valid path P_{valid_i} is a special path which the manager M will eventually check objects for. M owns a set of P_{valid} which includes all the valid paths in a supply chain.
- A reader R_i is honest-but-curious, i.e., a reader R_i at step v_i behaves correctly when it interacts with a tag which is going through it, but it will collect information from the interaction and might derive something non-trivial from those information.

2.3 Security Statements and Adversary Models

We formalize the security model using the game-based methodology. The game is played between a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

First, we describe several oracles which any PPT adversary could query during the interaction between an adversary and a challenger in the game-based security model.

- $\mathcal{O}_{\mathsf{nextsp}}(s^i_{T_i})$: On the query of a tag T_i's next step according to the state $s^i_{T_i}$, the challenger finds the next step reader R_j for $\mathcal{A}$ according to the current state of T_i. While R_j transforms the tag T_i from its current state to a new state.
- $\mathcal{O}_{\mathsf{rd}}(\mathsf{ID}_{T_i})$: On input of a tag T_i's identity ID_{T_i} by the adversary $\mathcal{A}$, this oracle returns the current state of T_i to $\mathcal{A}$.
- $\mathcal{O}_{\mathsf{enc}}(\mathsf{ID}_{T_i})$: On input of a tag T_i's identity ID_{T_i} by an adversary $\mathcal{A}$, the challenger responds to $\mathcal{A}$ the initial state of tag T_i by running $\mathsf{Enc}(\mathsf{ID}_{T_i})$.
- $\mathcal{O}_{\mathsf{reenc}}(s^j_{T_i})$: On input a state of the tag T_i, the challenger responds to $\mathcal{A}$ T_i's new state of next step $s^{j+1}_{T_i}$ by running $\mathsf{ReEnc}(s^j_{T_i})$.
- $\mathcal{O}_{\mathsf{cp}}(T_i)$: On the query of the path that a tag T_i went through, the challenger returns 1 to the adversary $\mathcal{A}$ if tag T_i went through a valid path; Otherwise, the challenger returns 0 to $\mathcal{A}$. However, the challenger does not return the real path to $\mathcal{A}$.
- $\mathcal{O}_{\mathsf{T,P}}(P_{\mathsf{valid}_i})$: On input a specified valid path P_{valid_i} by the adversary $\mathcal{A}$, the challenger randomly selects a tag from the path P_{valid_i}, and returns it to $\mathcal{A}$.
- $\mathcal{O}_{\mathsf{T,v}}(v')$: On input a specified step v', the challenger randomly picks a tag which has gone through step v', and returns it to $\mathcal{A}$.

We consider three common security requirements for RFID applications in a supply chain: authentication, tag privacy, and (tag and path) unlinkability [4].

[1] Since the tag's step in a supply chain is represented by a reader in our protocol, in this paper, we use "step" and "reader" interchangeably.

Authentication

Definition 1 (Authentication). *Authentication implies that any* PPT *adversary cannot forge a tag's internal state with a valid path that was not actually taken by the tag in the supply chain.*

We formalize this property using the following game between a PPT adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Definition 2 (Authentication Game). *A* PPT *adversary $\mathcal{A}$ is given the system's public parameters and the public keys of all readers and manager before he interacts with the challenger $\mathcal{C}$ in the following game.*

First, the adversary selects a target step $v*$, v^* is associated with some reader, say R_j.

Phase 1: *The adversary $\mathcal{A}$ makes the following queries:*

- $\mathcal{O}_{\mathsf{enc}}(\mathsf{ID}_{T_i})$*: For the initial state query for some tag T_i, the challenger chooses a valid path P_{valid_i} for $\mathcal{A}$, and returns $s^0_{T_i} \leftarrow \mathsf{Enc}(\mathsf{ID}_{T_i})$ to $\mathcal{A}$. The challenger records the path P_{valid_i} and the corresponding T_i in a table, i.e., T_{valid}.*
- $\mathcal{O}_{\mathsf{reenc}}(s^j_{T_i})$*: $\mathcal{A}$ makes request to $\mathcal{C}$ for a new state of tag T_i, the challenger searches T_{valid} for the path of T_i and finds out the next reader of T_i, i.e., R_j. Then, the challenger updates the state $s^j_{T_i}$ to a new state $s^{j+1}_{T_i}$ by running $\mathsf{ReEnc}(s^j_{T_i})$, and returns the new state $s^{j+1}_{T_i}$ to $\mathcal{A}$.*
- $\mathcal{O}_{\mathsf{rd}}(\mathsf{ID}_{T_i})$*: The challenger reads out the current state of tag T_i and returns it to $\mathcal{A}$.*
- $\mathcal{O}_{\mathsf{nextsp}}(s^i_{T_i})$*: On input of a state of some tag T_i, the challenger searches T_{valid} for the path of T_i and finds out the next reader of T_i, i.e., R_j. Finally, the challenger returns R_j to $\mathcal{A}$.*
- $\mathcal{O}_{\mathsf{cp}}(T_i)$*: on input of a tag T_i, the challenger returns 1 to the adversary $\mathcal{A}$ if the tag T_i went through a valid path; Otherwise, the challenger returns 0 to $\mathcal{A}$. However, the challenger does not return the real path to $\mathcal{A}$.*

Challenge: *$\mathcal{A}$ selects a tag T_c, and outputs a forged state of tag T_c at step r as $s^r_{T_c}$.*

Decision: *The challenger computes $(\mathsf{ID}_{T_c}, P_{\mathsf{valid}_k}) \leftarrow \mathsf{Dec}(s^r_{T_c})$. If tag T_c did not go through the step v^* and $v^* \in P_{\mathsf{valid}_k}$, the challenger outputs 1; Otherwise, he outputs 0.*

Definition 3. *Let* adv *denote the advantage that $\mathcal{A}$ outputs a valid tag state $s^r_{T_c}$ in the above security game, and $\mathcal{C}$ outputs 1 in the **Decision** stage. We say a path authentication solution is authenticated if for all* PPT *adversary $\mathcal{A}$, $\mathsf{Pr}[\mathsf{adv}] \leq \varepsilon$ holds, where ε is negligible.*

Tag Privacy

Definition 4 (Privacy). *We say that a path authentication solution keeps the privacy property, if for any* PPT *adversary $\mathcal{A}$, he cannot tell whether a tag T_i went through some step, say, reader R, in the supply chain only based on the data stored on the tag [4].*

We formally define the security model for tag privacy in the following game.

Definition 5 (Privacy Game). *A* PPT *adversary* $\mathcal{A}$ *is given the system's public parameters and the public keys of all readers and manager before he interacts with the challenger* $\mathcal{C}$ *in the following game.*

Choose: *The adversary* $\mathcal{A}$ *chooses a reader* R *(step) as his target.*

Phase 1: *In this phase, the adversary* $\mathcal{A}$ *makes the following queries to the challenger* $\mathcal{C}$*:*

- *For any queries with the form of* $\mathcal{O}_{\mathsf{enc}}(\mathsf{ID}_{T_i})$, $\mathcal{O}_{\mathsf{reenc}}(s^j_{T_i})$, $\mathcal{O}_{\mathsf{rd}}(\mathsf{ID}_{T_i})$, $\mathcal{O}_{\mathsf{nextsp}}(s^j_{T_i})$, *and* $\mathcal{O}_{\mathsf{cp}}(T_i)$*, the challenger* $\mathcal{C}$ *responds to* $\mathcal{A}$ *in the same way as in the "Authentication Game"*
- $\mathcal{O}_{\mathsf{T,v}}(v')$*: On input of a specified step* v'*, the challenger picks a tag which went through step* v' *randomly, and returns it to* $\mathcal{A}$*.*

Challenge: $\mathcal{C}$ *chooses a random bit* b *from* $\{0,1\}$*. If* $b=0$*,* $\mathcal{C}$ *selects a tag* T_c *which did not go through* R*. Otherwise,* $\mathcal{C}$ *selects a tag* T_c *which went through* R*. Then,* $\mathcal{C}$ *reads out the current state of* T_c*, i.e.,* $s^j_{T_c}$*, and sends to* $\mathcal{A}$ *an updated state* $s^{j+1}_{T_c}$ *computed using* $\mathsf{ReEnc}(s^j_{T_c})$*.*

Phase 2: $\mathcal{A}$ *continues to make the above queries to* $\mathcal{C}$ *adaptively as in* ***Phase 1****, with the restriction that* $\mathcal{A}$ *cannot make a query on* $\mathcal{O}_{\mathsf{T,v}}(R)$*.*

Decision: *Finally,* $\mathcal{A}$ *outputs a guess* $b'=1$ *if he regards* T_c *went through* R*. Otherwise, he outputs 0.*

Definition 6. *Let* adv *denote the event that* $\mathcal{A}$ *outputs a right guess in the above game. We say that a path authentication solution is privacy preserving, if for any* PPT *adversary* $\mathcal{A}$*,* $\mathsf{Pr}[\mathsf{adv}] \leq \varepsilon$ *holds, where* ε *is negligible.*

Unlinkability. In accordance with [4], unlinkability is divided into tag unlinkability and path unlinkability.

Tag unlinkability means that given some states of two arbitrary tags T_0 and T_1 in a supply chain, no PPT adversary can distinguish T_0 from T_1 with non-negligible advantage.

We define the security game for tag unlinkability as follows:

Definition 7 (Tag Unlinkability Game). *The* PPT *adversary* $\mathcal{A}$ *is given the system's public parameters and the public keys of all readers and manager before he interacts with the challenger* $\mathcal{C}$ *in the following game.*

Choose: *The adversary* $\mathcal{A}$ *chooses two random tags* T_0 *and* T_1[2]*.*

Phase 1: *The adversary* $\mathcal{A}$ *makes the following queries:*

- *For any queries with the form of* $\mathcal{O}_{\mathsf{enc}}(\mathsf{ID}_{T_i})$, $\mathcal{O}_{\mathsf{reenc}}(s^j_{T_i})$, $\mathcal{O}_{\mathsf{rd}}(\mathsf{ID}_{T_i})$, $\mathcal{O}_{\mathsf{nextsp}}(s^j_{T_i})$, *and* $\mathcal{O}_{\mathsf{cp}}(T_i)$*, the challenger* $\mathcal{C}$ *responds to* $\mathcal{A}$ *in the same way as in the "Authentication Game"*
- $\mathcal{O}_{\mathsf{T,P}}(P_{valid_j})$*: On input of a specified valid path* P_{valid_j} *by the adversary* $\mathcal{A}$*, which both* T_0 *and* T_1 *did not go through, the challenger picks a tag from the path* P_{valid_j} *randomly, and returns it to* $\mathcal{A}$*.*

[2] We suppose that both T_0 and T_1 must be in a valid path in the following simulation.

Challenge: *First, $\mathcal{C}$ chooses a random bit b from $\{0,1\}$. Second, $\mathcal{C}$ reads out the current state of T_b, i.e., $s^i_{T_b}$. Finally, $\mathcal{C}$ updates state $s^i_{T_b}$ to a new state $s^{i+1}_{T_b}$ by running $\mathcal{O}_{\mathsf{reenc}}(s^i_{T_b})$, and returns $s^{i+1}_{T_b}$ to $\mathcal{A}$.*

Phase 2: *$\mathcal{A}$ continues to make queries as in **Phase 1**.*

Decision: *$\mathcal{A}$ outputs his guess b'. If $b' = b$, $\mathcal{A}$ is said to be successful in the game; Otherwise, $\mathcal{A}$ fails in the game.*

Definition 8 (Tag Unlinkability). *Let* adv *define the event that $\mathcal{A}$ outputs a right guess in the **decision phase** in the above tag unlinkability game. We say that a path authentication solution is tag unlinkable, if for any* PPT *adversary $\mathcal{A}$, $\mathsf{Pr}[\mathsf{adv}] \leq \frac{1}{2} + \varepsilon$ holds, where ε is negligible.*

Path unlinkability. Path unlinkability means that given two tags T_i and T_j, no PPT adversary $\mathcal{A}$ can tell whether these two tags went through the same path with probability at least $\frac{1}{2} + \varepsilon$.

Next, we define the security game for path unlinkability.

Definition 9 (Path Unlinkability Game). *The* PPT *adversary $\mathcal{A}$ is given the system's public parameters and the public keys of all readers and manager before he interacts with the challenger $\mathcal{C}$ in the following game.*

Choose: *$\mathcal{A}$ chooses a random tag T, $\mathcal{C}$ then gives the path P_{valid_t} that T went through to $\mathcal{A}$.*

Phase 1: *The adversary $\mathcal{A}$ makes the following queries:*

- *$\mathcal{O}_{\mathsf{T,P}}(P_{valid_j})$: On input of a specified valid path P_{valid_j} by the adversary $\mathcal{A}$, the challenger picks a tag from the path P_{valid_j} randomly, and returns it to $\mathcal{A}$.*
- *For any queries with the form of $\mathcal{O}_{\mathsf{enc}}(\mathsf{ID}_{T_i})$, $\mathcal{O}_{\mathsf{reenc}}(s^j_{T_i})$, $\mathcal{O}_{\mathsf{rd}}(\mathsf{ID}_{T_i})$, $\mathcal{O}_{\mathsf{nextsp}}(s^j_{T_i})$, and $\mathcal{O}_{\mathsf{cp}}(T_i)$, the challenger $\mathcal{C}$ responds to $\mathcal{A}$ in the same as in the "Authentication Game"*

Challenge: *First, $\mathcal{C}$ chooses a random bit b from $\{0,1\}$. If $b = 0$, $\mathcal{C}$ randomly chooses a tag T_c which does not go through P_{valid_t}; Otherwise, if $b = 1$, $\mathcal{C}$ randomly chooses a tag T_c which goes through P_{valid_t}. Second, $\mathcal{C}$ reads out the current state of T_c, i.e., $s^i_{T_c}$. Finally, $\mathcal{C}$ updates state $s^i_{T_c}$ into a new state $s^{i+1}_{T_c}$ by running $\mathcal{O}_{\mathsf{reenc}}(s^i_{T_c})$, and sends $s^{i+1}_{T_c}$ to $\mathcal{A}$ as the target.*

Phase 2: *$\mathcal{A}$ continues to make queries as in **Phase 1** with the restriction that $\mathcal{A}$ cannot make a query on $\mathcal{O}_{\mathsf{T,P}}(P_{valid_t})$.*

Decision: *$\mathcal{A}$ outputs his guess b' which indicates whether T_c goes through P_{valid_t}, where $b' = 1$ means that $\mathcal{A}$ guesses T_c goes through P_{valid_t}, and$b' = 0$ means that $\mathcal{A}$ guesses T_c does not go through path P_{valid_t}.*

Definition 10 (Path Unlinkability). *Suppose that* adv *defines the event that $\mathcal{A}$ outputs a right guess in the **decision phase** in the above path unlinkability game. We say that a path authentication solution is path unlinkable, if for any* PPT *adversary $\mathcal{A}$, $\mathsf{Pr}[\mathsf{adv}] \leq \frac{1}{2} + \varepsilon$ holds, where ε is negligible.*

3 The Proposed Tag Path Authentication Protocol

In this section, we first recall the Tracker protocol [DBLP:conf/ndss/BlassEM11,21] in Section 3.1. Second, we propose our track and trace protocol for RFID-based supply chain in Section 3.2. Finally, we give a concise comparison on these two protocols to show that our protocol is more efficient in computational cost and tag's memory space overhead.

3.1 Description of Tracker Protocol

Typically, a tracker protocol consists of the four phases: (1) an initial setup phase; (2)new tags' preparation for entering the supply chain; (3) the interaction between a tag and a reader in the supply chain; and (4) the manager's verification on a path. These four phases are described as follows [4]:

Initialization This phase is done by the issuer I:

1. Select a homomorphic mapping $\mathcal{M}_\Phi : \mathbb{F}_q \to \mathcal{E}$ to map a mark $\phi(\mathcal{P})$ to a point in the elliptic curve such that $\forall m_1, m_2 \in \mathbb{F}_q, \mathcal{M}_\phi(m_1 + m_2) = \mathcal{M}_\phi(m_1) + \mathcal{M}_\phi(m_2)$, and a mapping of mark $\phi(\mathcal{P}) \in \mathbb{F}_q$ to a point as $\mathcal{M}_\phi(\phi(\mathcal{P})) = \phi(\mathcal{P}) \cdot P \in \mathcal{E}$.
2. Set up an elliptic curve ElGamal cryptosystem **??** and generate the secret key sk and public key $pk = (P, Y = sk \cdot P)$, such that the order of P is a large prime q, $|q| = 160$ bit.
3. Select x_0, a generator of the finite field $\mathbb{F}_q$, and $a_0 \leftarrow_R \mathbb{F}_q$.
4. Generate a random bit string k_0, $|k_0| = 160$ bit. The initial step v_0, representing the issuer in the supply chain, is associated with (a_0, k_0).
5. Generate η random numbers $a_i \in \mathbb{F}_q, 1 \leq i \leq \eta$, and η random bit string k_i, $|k_i| = 160$ bit. I sends to each reader R_i, representing step v_i, the tuple (i, a_i, k_i) using a secure channel.
6. I provides M with secret key sk, generator x_0, and tuple (i, a_i, k_i). Therewith, M is equipped with all the keys and informed which reader R_i at step v_i knows which (a_i, k_i)
7. The manager M knows all the valid paths in a set S_{valid}, he computes all the $|S_{\mathsf{valid}}|$ valid path marks $\phi(P_{\mathsf{valid}})$.
8. Finally, M computes and stores pairs$(M_\phi(\phi(P_{\mathsf{valid}})$, steps), where steps is the sequence of steps $\overrightarrow{v_0 v_{P_{\mathsf{valid},1}} \dots v_{P_{\mathsf{valid},\ell}}}$ of P_{valid_i}. That is, M knows for each mapping the sequence of steps.

Preparation

- Draw a random identification $ID \in \mathbb{F}_q$ and two random numbers $r_\phi, r_{ID} \in \mathbb{F}_q$.
- Compute

$$c^0_{ID} = E(ID) = (U_{ID}, V_{ID}) = (r_{ID} \cdot P, \mathcal{M}(ID) + r_{ID} \cdot Y)$$

$$c^0_\phi = E(\phi(v_0)) = (U^0_\phi, V^0_\phi) = (r_\phi \cdot P, a_0 \dot{P} + r_\phi \cdot Y)$$

- Let HMAC be a secure HMAC algorithm, $\mathsf{HMAC}_k(m) : \mathbb{F}_q \times \mathbb{F}_q \to \mathbb{F}_q$. I computes signature $\sigma^0(v_0, ID) := \mathsf{HMAC}_{k_0}(ID)$.
- Finally, the issuer I writes state $s_T^0 = (c_{ID}^0), c_\phi^0, \sigma_0)$ into T. Now, T is ready to enter the supply chain.

Interaction

- Assume that a tag T arrives at step v_i and reader R_i in the supply chain $P = \overrightarrow{v_0 v_1 \dots v_{i-1}}$. R_i reads out T's current state $s_T^{i-1} = (c_{ID}^{i-1}), c_\phi^{i-1}, \sigma^{i-1})$.
- Given the ciphertext $c_\phi^{i-1} = (U_\phi^{i-1}, V_\phi^{i-1}), x_0$ and a_i, R_i computes $c_\phi^i = (U_\phi^i, V_\phi^i)$, wehre

$$U_\phi^i = x_0 \cdot U_\phi^{i-1} = (x_0 r_\phi^{i-1}) \cdot P$$

$$\begin{aligned} V_\phi^i &= x_0 \dot{V}_\phi^{i-1} + a_i \cdot P \\ &= (a_0 x_0^i + \sum_{j=1}^{i} a_j x_0^{i-j}) \cdot P + (x_0 r_\phi^{i-j}) \cdot Y \end{aligned}$$

- Using $\sigma^{i-1}(ID)$, R_i computes $\sigma^i(ID) = \mathsf{HMAC}_{k_i}(\sigma^{i-1}(ID))$.
- R_i re-encryps $c_{ID}^{i-1}), c_\phi^i$. It picks randomly two numbers r'_{ID} and $r'_\phi \in \mathbb{F}_q$, and outputs two new ciphertext as:

$$c_{ID}^i = (U_{ID}^i, V_{ID}^i) = (r'_{ID} \cdot P + U_{ID}^{i-1}, r'_{ID} \cdot Y + V_{ID}^{i-1})$$

$$c_\phi^{'i} = (U_\phi^{'i}, V_\phi^{'i}) = (r'_\phi \cdot P + U_\phi^i, r'_\phi \cdot Y + V_\phi^i)$$

Veriication

- M reads out tag T's state $s_T^\ell = (c_{ID}^\ell, c_\phi^\ell, \sigma^\ell(ID))$.
- M decrypts c_{ID}^ℓ to get the plaintext $ID = D_{sk}(c_{ID}^\ell) \in \mathbb{F}_q$.
- M checks for cloning, by looking up ID in M's database DB_{clone}. If $ID \in DB_{clone}$, then M outputs $\varnothing$ and rejects T.
- Otherwise, M decrypts c_ϕ^ℓ and gets $\pi = D_{sk}(c_\phi^\ell) = \phi(\mathcal{P}) \cdot P$. Then, M matches the result with his list of valid mapping $\mathcal{M}_\phi(\phi(P_{\mathsf{valid}_i}))$. If there is no match existed, M outputs $\varnothing$ and rejects T.
- M checks the signature: check if the following equation holds using the secret keys $(k_0, k_1, \dots, k_\ell)$,

$$\sigma^\ell(ID) = \mathsf{HMAC}_{k_\ell}(\mathsf{HMAC}_{k_{\ell-1}}(\dots(\mathsf{HMAC}_{k_0}(ID)))).$$

- If the above equation holds, M outputs P_{valid}, add ID to DB_{clone}. Otherwise, M outputs $\varnothing$ and rejects T.

3.2 Our Protocol

In this part, we propose an efficient tag track and trace protocol for RFID-based supply chains. Our protocol shares the assumptions of the first tag track and trace protocol [[4],21]. Compared with [4], our protocol is better both in computation and memory cost. Our protocol consists of the following five algorithms:

Setup: It outputs the system's public parameters par, including an elliptic curve $\mathcal{E}$ over a finite field $\mathbb{F}_p$. $\mathcal{E}(\mathbb{F}_p)$ is of a large prime order q such that the discrete logarithm problem is intractable for $\mathcal{G} = \langle g \rangle$, wehre g is a generator on $\mathcal{E}(\mathbb{F}_p)$. Here, p and q are security parameters with $|p| = |q| = 160$ bit. Meanwhile, a cryptographic collision-resistent hash function $\mathsf{H} : \{0,1\}^* \to \mathcal{G}$ is output by this algorithm.

KeyGen: It generates the public/private key pairs for all the readers in the system as well as the manager. For a reader R_i, it picks a random element x_i from $\mathbb{F}_q$, and sets $sk_i = x_i$, $pk_i = g^{x_i}$. For the only manager in the system, the algorithm selects two random x_{m_1}, x_{m_2} from $\mathbb{F}_q$, and sets $sk_m = (x_{m_1}, x_{m_2})$, $pk_m = (pk_{m_1}, pk_{m_2}) = (g^{x_{m_1}}, g^{x_{m_2}})$.

The above two algorithms are run by the issuer during the system initialization stage.

Enc: When a tag T is ready to enter a supply chain, the issuer I computes the initial state s_T^0 of T on T's identification ID_T using this algorithm.

- Select a valid path P_{valid_i} for tag T, and a random number $r \in \mathbb{F}_q$.
- Compute $s_T^0 = (s_{1_T}^0, s_{2_T}^0, s_{3_T}^0, s_{4_T}^0)$, where $s_{1_T}^0 = g^r$, $s_{2_T}^0 = pk_{m_1}^r \cdot \mathsf{ID}_T$, $s_{3_T}^0 = pk_{m_2}^r \cdot \mathsf{H}(\mathsf{ID}_T, P_{\mathsf{valid}_i})$, ,$s_{4_T}^0 = (pk_1 \ldots pk_\ell)^r$. Here, $pk_1 \ldots pk_\ell$ are the respective public keys of all the readers in the system.
- Finally, the issuer I writes state s_T^0 into the tag T with the identification is ID_T. Now, the tag T is qualified to enter the supply chain.

This algorithm is run by the issuer during the tag preparation stage.

ReEnc: When a tag T interacts with a reader, the reader reads out the current state s_T^i of T, then, re-encrypts T's state s_T^i into a new state s_T^{i+1}. The re-encryption algorithm is described as follows:

- Aessume that a tag T arrives at step v_{i+1} and reader R_{i+1} with public key pk_{i+1} reads out T's current state s_T^i for some $i, 1 \le i \le \ell$.
- Given the state information s_T^i, we divide it into the following two cases:
  1. For $i = 0$, which means the tag is read by the first reader in the supply chain. In this case, we parse s_T^0 into $(s_{1_T}^0, s_{2_T}^0, s_{3_T}^0, s_{4_T}^0)$. Then, R_{i+1} selects a random number $r_{i+1} \in \mathbb{F}_q$, and computes $s_T^{i+1} = (s_{1,1_T}^{i+1}, s_{1,2_T}^{i+1}, s_{2_T}^{i+1}, s_{3_T}^{i+1}, s_{4_T}^{i+1})$, where

$$s_{1,1_T}^{i+1} = s_{1_T}^0 \cdot g^{r_1} = g^{r+r_1}$$

$$s_{1,2_T}^{i+1} = (s_{1_T}^0)^{r_1} = g^{r \cdot r_1}$$

$$s_{2_T}^{i+1} = s_{2_T}^0 \cdot pk_{m_1}^{r_1} = pk_{m_1}^{r+r_1} \cdot \mathsf{ID}_T$$

$$s_{3_T}^{i+1} = s_{3_T}^0 \cdot pk_{m_2}^{r_1} = pk_{m_2}^{r+r_1} \cdot \mathsf{H}(\mathsf{ID}_T, P_{\mathsf{valid}_i})$$

$$s_{4_T}^{i+1} = \frac{s_{4_T}^0}{(s_{1_T}^0)^{sk_{i+1}}} = (pk_1 \ldots pk_i \cdot pk_{i+2} \ldots pk_\ell)^{r \cdot r_1}$$

2. For $i \geq 1$, we parse s_T^i into $(s_{1,1_T}^i, s_{1,2_T}^i, s_{2_T}^i, s_{3_T}^i, s_{4_T}^i)$. Then, R_{i+1} selects a random number $r_{i+1} \in \mathbb{F}_q$, and computes $s_T^{i+1} = (s_{1,1_T}^{i+1}, s_{1,2_T}^{i+1}, s_{2_T}^{i+1}, s_{3_T}^{i+1}, s_{4_T}^{i+1})$, where

$$s_{1,1_T}^{i+1} = s_{1_T}^i \cdot g^{r_{i+1}} = g^{r+r_1+\ldots+r_{i+1}}$$

$$s_{1,2_T}^{i+1} = (s_{1_T}^i)^{r_{i+1}} = g^{r \cdot r_1 \ldots r_{i+1}}$$

$$s_{2_T}^{i+1} = s_{2_T}^i \cdot pk_{m_1}^{r_{i+1}} = pk_{m_1}^{r+r_1+\ldots+r_{i+1}} \cdot \mathsf{ID}_T$$

$$s_{3_T}^{i+1} = s_{3_T}^i \cdot pk_{m_2}^{r_{i+1}} = pk_{m_2}^{r+r_1+\ldots+r_{i+1}} \cdot \mathsf{H}(\mathsf{ID}_T, P_{\mathsf{valid}_i})$$

$$s_{4_T}^{i+1} = \frac{s_{4_T}^i}{(s_{1,2_T}^i)^{sk_{i+1}}} = (pk_{i+2} \ldots pk_\ell)^{r \cdot r_1 \ldots r_{i+1}}$$[3]

This algorithm is run by the respecting reader when the tag and reader interacts during the interaction stage.

Dec: When a tag T with the final state s_T^k arrives at the manager, the manager checks whether T has gone through a valid path specified by the issuer using this algorithm. The algorithm gets the identification of T as well as its path by decrypting s_T^k.

- M reads out T's state $(s_T^k = s_{1,1_T}^k, s_{1,2_T}^k, s_{2_T}^k, s_{3_T}^k, s_{4_T}^k)$.
- M decrypts $s_{2_T}^k$ to get the plaintext $\mathsf{ID}_T = \frac{s_{2_T}^k}{(s_{1,1_T}^k)^{sk_{m_1}}}$.
- For a possible valid path, suppose that $pk_i \ldots, pk_j$ are public keys of those readers who are not in that valid path. M checks whether the following equation holds or not:

$$s_{4_T}^k \stackrel{?}{=} (s_{1,2_T}^k)^{x_i+\ldots x_j}$$

  The manager can find out all the readers who took part in the interaction with the tag T.
- The manager further verifies the path by testing whether $\mathsf{H}(\mathsf{ID}_T, P_{\mathsf{valid}_i}) = s_{3_T}^k / (s_{1,1_T}^k)^{sk_{m_2}}$, where k represents the last step that the tag T has gone through.

This algorithm is run by the unique manager during the path verification stage.

[3] For simplicity, we use R_i (whose public key is pk_i) to represents the corresponding step i in the supply chain in our protocol, where $1 \leq i \leq \ell$.

3.3 Comparison

From the above description of the two track and trace protocols for RFID-based supply chains, we can find that in our newly proposed protocol, we encrypt a tag and its path as a whole message under the manager's public key. Each time a reader interacts with a tag, it erases itself from one element of the ciphertext, and randomly re-encrypts all the elements of the ciphertexts. While, Tracker [4],21] uses an HMAC signature to further verify the tag and its path. There is no need to use an HMAC signature to further verify the tag and its valid path in our protocol, since the manager could verify the tag and path by decrypting the state of the tag. Such improvement reduces the memory space of tags from 960 bits to 800 bits, since an HMAC signature needs 160 bits.

4 Security Analysis on The Proposed Protocol

We give a security analysis on security, privacy, and unlinkability, respectively in this section. For all the following security proof, we use the same system's parameters.

4.1 Authentication

Theorem 1. *Any forged state of tag T_i output by a* PPT *adversary $\mathcal{A}$, which $\mathcal{A}$ has claimed that the tag T_i has gone through some step but in fact the tag does not go through it in a supply chain, can be detected by the challenger.*

Proof. Given any PPT adversary $\mathcal{A}$ attacking our tag path authentication protocol on the security property of authentication, the challenger can always detect whether the tag with that state went through the target step, say v^*.

$\mathcal{C}$ runs Setup to generate the system's public parameters, such as an elliptic curve $\mathcal{E}$ over a finite field $\mathbb{F}_p$, a large prime order q of $\mathcal{E}$ such that the discrete logarithm problem is intractable for $\mathcal{G} = \langle g \rangle$, where g is a generator on $\mathcal{E}(\mathbb{F}_p)$, and p and q are security parameters with $|p| = |q| = 160$ bit. Meanwhile, a cryptographic collision-resistent hash function $\mathsf{H} : \{0,1\}^* \to \mathcal{G}$ is output by $\mathcal{C}$. $\mathcal{C}$ also generates the public/private key pairs for all the readers and the manager in the system. Finally, $\mathcal{C}$ gives these public parameters and public keys to $\mathcal{A}$. Next, we describe how $\mathcal{A}$ and $\mathcal{C}$ interact during the security game.

First, the adversary $\mathcal{A}$ chooses a random step v^* as his target, v^* is associated with some reader R_j.

Phase 1: In this phase, $\mathcal{A}$ can adaptively make queries including $\mathcal{O}_{\mathsf{enc}}(\mathsf{ID}_{T_i})$, $\mathcal{O}_{\mathsf{reenc}}(s^j_{T_i})$, $\mathcal{O}_{\mathsf{rd}}(\mathsf{ID}_{T_i})$, $\mathcal{O}_{\mathsf{nextsp}}(s^j_{T_i})$, and $\mathcal{O}_{\mathsf{cp}}(T_i)$, $\mathcal{C}$ responds to $\mathcal{A}$ as described in the authentication security game.

Challenge: In this phase, $\mathcal{A}$ outputs a forged state of arbitrary tag T_c. We denote the state by $s^r_{T_c} = (s^r_{1,1_{T_c}}, s^r_{1,2_{T_c}}, s^r_{2_{T_c}}, s^r_{3_{T_c}}, s^r_{4_{T_c}})$.

Decision: In this phase, the challenger computes ID_{T_c} and $\mathsf{H}(\mathsf{ID}_{T_c}, P_{\mathsf{valid}_i})$ by running $\mathsf{Dec}(s^r_{T_c})$. If P_{valid_i} contains v^* (i.e., R_j), but R_j's public key pk_j remained in $s^r_{4_{T_c}}$, then, $\mathcal{C}$ outputs 1 (it means that tag T_c has not gone through the step v^*, but $v^* \in P_{valid_i}$.) Otherwise, $\mathcal{C}$ outputs 0.

Analysis. Suppose that the adversary claims that the forged state of tag T_c did not go through the step v^*, but in fact v^* appears in P_{valid_i}, the challenger can detect it easily. Since if the tag T_c did not go through the step v^*, without loss of generality, we use reader R_j, whose public key is pk_j to represent the step v^*, then pk_j must appear in $s^i_{4_{T_c}}$. If $\mathcal{C}$ has judged that pk_j appears in $s^r_{4_{T_c}}$, then $\mathcal{C}$ can easily draw the conclusion that the state is not a valid state of tag T_c if P_{valid_i} contains R_j, where P_{valid_i} can be computed by the challenger using the manager's private key.

This indicates that the adversary cannot forge a valid state of any tag that he claimed having gone through a valid path, but in fact the tag did not go through it.

4.2 Privacy

Theorem 2. *If there exists a* PPT *adversary* $\mathcal{A}$ *that could tell whether a tag went through some step* v *in the supply chain with non-negligible advantage* ε*, then, there exists another* PPT *algorithm* $\mathcal{B}$ *that can solve the discrete logarithm problem with the same advantage.*

Proof. In the beginning of the game, $\mathcal{C}$ generates the system's public parameters and public/private key pairs for all the readers and the manager as those in security proof for authentication. We omit it here for brevity. Finally, $\mathcal{C}$ gives these public parameters and public keys to the adversary $\mathcal{A}$.

Choose: The adversary $\mathcal{A}$ chooses a reader R as his target step.

Phase 1: In this phase, $\mathcal{A}$ can adaptively make queries including $\mathcal{O}_{\mathsf{enc}}(\mathsf{ID}_{T_i})$, $\mathcal{O}_{\mathsf{reenc}}(s^j_{T_i})$, $\mathcal{O}_{\mathsf{rd}}(\mathsf{ID}_{T_i})$, $\mathcal{O}_{\mathsf{nextsp}}(s^j_{T_i})$, $\mathcal{O}_{\mathsf{cp}}(T_i)$, and $\mathcal{O}_{\mathsf{T,v}}(v)$ $\mathcal{C}$ responds to $\mathcal{A}$ as described in the privacy game.

Challenge: $\mathcal{C}$ chooses a random bit b from $\{0,1\}$. If $b = 0$, $\mathcal{C}$ selects a tag T_c which did not go through R. Otherwise, $\mathcal{C}$ selects a tag T_c which went through R. Then, $\mathcal{C}$ reads out the current state of T_c, i.e., $s^j_{T_c}$, and sends to $\mathcal{A}$ an updated state $s^{j+1}_{T_c}$ computed using $\mathsf{ReEnc}(s^j_{T_c})$.

Phase 2: In this phase, $\mathcal{A}$ can continue to make queries as those in **Phase 1** adaptively with the restriction that $\mathcal{A}$ cannot make a query on $\mathcal{O}_{\mathsf{T,v}}(R)$.

Decision: The adversary $\mathcal{A}$ outputs his guess bit b'. If $b' = 1$, means he guesses T_c went through R. If $b' = 0$, means he guesses T_c did not got through R.

Analysis. The tag path authentication protocol is a typical ElGamal-based public key encryption scheme [23], and randomly re-encrypted each time when a reader interacts with the tag. ElGamal encryption scheme [23] itself is based on the discrete logarithm problem. So, if the adversary can identify a tag and its path which are encrypted by the above encryption scheme, it means that he can

decrypt the state of the tag. With the help of this adversary, we can construct another adversary who can directly solve the discrete logarithm problem.

4.3 Unlinkability

Unlinkability includes tag unlinkability and path unlinkability. Tag unlinkability means that given some states of two arbitrary tags T_0 and T_1 in a supply chain, no PPT adversary $\mathcal{A}$ can distinguish T_0 from T_1 with non-negligible advantage. Path unlinkability means that given two tags T_0 and T_1 in a supply chain, no PPT adversary $\mathcal{A}$ can tell if these two tags went through a same path with non-negligible advantage. Intuitively, tag unlinkability implies path unlinkability. Since, if there is a PPT adversary who can tell whether two tags go through an identical path, we can construct another adversary, and with the help of the previous adversary, the later can distinguish these two tags from some states that were read out from some reader in the path. However, the reverse does not hold. So, in this part, we only give a proof sketch of tag unlinkability.

Theorem 3. *If there exists a* PPT *adversary* $\mathcal{A}$ *that can break the tag unlinkability of our protocol, then there must exist a* PPT *adversary* $\mathcal{B}$ *who can break the* $\mathsf{IND}-\mathsf{CPA}$ *security of ElGamal encryption scheme.*

Proof. Recall the definition of $\mathsf{IND}-\mathsf{CPA}$ security: we say a scheme is $\mathsf{IND}-\mathsf{CPA}$ secure if a PPT adversary $\mathcal{A}$ is given a ciphertext of randomly chosen two messages m_0 and m_1 with identical length in the message space after he has accessed to private-key extraction oracle several times with the restriction that $\mathcal{A}$ is not allowed to make the private key query on the target entity. $\mathcal{A}$ cannot distinguish whether the target ciphertext from the challenger is of m_0 or m_1. It is well known that ElGamal public key encryption scheme is $\mathsf{IND}-\mathsf{CPA}$ secure, whose security is based on the discrete logarithm problem. If the adversary can distinguish two tags' states (which means that the adversary can distinguish two messages of ElGamal-based's ciphertexts),it breaks the $\mathsf{IND}-\mathsf{CPA}$ security of ElGamal scheme. So, there is no such adversary who can break the tag path unlinkability of our tag path authentication protocol.

5 Performance Comparison

Analysis in this part shows that our protocol is more efficient than the previous work both in the computational cost and memory space of tags.

Table 1 shows that our protocol only needs 4 multiplication and 6 exponentiation operations in re-encryption, while Tracker [4] needs 3 multiplication and 8 exponentiation operations in re-encryption. During the verification, we need 2 multiplication and 2 exponentiation operations, while Tracker [8] needs 3 multiplication and 5 exponentiation operations. Meanwhile, Tracker [4] needs 4 HMAC signatures in the running of the protocol, we even do not need any signature. But we need ℓ-|path| addition operations, while Tracker does not need. Here, |path| is the length of the valid path. However, the addition operation is

Table 1. Comparison on Computational Costs

	Tracker [4]		Ours	
	Re-encryption	Verification	Re-encryption	Verification
Multiplication	3	3	4	2
Exponentiation	8	5	6	2
Addition	none	none	none	$\ell - \|path\|$
HMAC	2	2	none	

negligible compared with multiplication and exponentiation operation. So, we can draw the conclusion that our tag path authentication protocol is much more efficient than [4] in computational cost. This enhances the system's efficiency which makes it more practical in real implementation.

Table 2. Comparison on Memory Size

	Tracker [4]	Our Protocol
Tag Memories	960 bits	800 bits

From Table 2, we observe that our protocol needs less tag memory than Tracker [4] (i.e., 5 elliptic group elements (800 bits) vs. 6 elliptic group elements (960 bits)). Since tags with less memory are less cheap and widely-accepted, researchers tend to design secure tag path authentication protocol with less tag memories in RFID-based supply chain management.

6 Conclusion

In this paper, we present a more efficient path authentication protocol in an RFID-based supply chain. Our solution is a significant improvement over the previous work [4] both in computational cost and memory requirement on tags. Our protocol is compatible with EPC Class 1 Gen 2 tags and is provably secure under authentication, privacy, and (tag and path) unlinkability.

References

1. ABISearch: Rfid annual market overview (2010), `http://www.abisearch.com`
2. Bellare, M.: New proofs for nmac and hmac: Security without collision resistance. Journal of Cryptology 2006(1), 602–619 (2006)
3. Berbain, C., Billet, O., Etrog, J., Gilbert, H.: An efficient forward private rfid protocol. In: ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, Usa, November. pp. 43–53 (2009)
4. Blass, E., Elkhiyaoui, K., Molva, R.: Tracker: Security and privacy for rfid-based supply chains. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, US-

A, 6th February - 9th February 2011. The Internet Society (2011), http://www.isoc.org/isoc/conferences/ndss/11/pdf/9_1.pdf
5. Cai, S., Deng, R.H., Li, Y., Zhao, Y.: A new framework for privacy of RFID path authentication. In: Bao, F., Samarati, P., Zhou, J. (eds.) Applied Cryptography and Network Security - 10th International Conference, ACNS 2012, Singapore, June 26-29, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7341, pp. 473–488. Springer (2012), http://dx.doi.org/10.1007/978-3-642-31284-7_28
6. Čapkun, S., Buttyn, L., Hubaux, J.P.: Sector: Secure tracking of node encounters in multi-hop wireless networks. Sasn pp. 21–32 (2003)
7. Deng, J., Han, R., Mishra, S.: Security support for in-network processing in wireless sensor networks. In: ACM Workshop on Security of Ad Hoc and Sensor Networks, Sasn 2003, Fairfax, Virginia, Usa. pp. 83–93 (2003)
8. Deng, R.H., Li, Y., Yung, M., Zhao, Y.: A new framework for RFID privacy. In: Gritzalis, D., Preneel, B., Theoharidou, M. (eds.) Computer Security - ESORICS 2010, 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6345, pp. 1–18. Springer (2010), http://dx.doi.org/10.1007/978-3-642-15497-3_1
9. Elgamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory 31(4), 469–472 (1985)
10. Fan, K., Ge, N., Gong, Y., Li, H., Su, R., Yang, Y.: ULRAS: ultra-lightweight RFID authentication scheme for mobile device. In: Xu, K., Zhu, H. (eds.) Wireless Algorithms, Systems, and Applications - 10th International Conference, WASA 2015, Qufu, China, August 10-12, 2015, Proceedings. Lecture Notes in Computer Science, vol. 9204, pp. 114–122. Springer (2015), http://dx.doi.org/10.1007/978-3-319-21837-3_12
11. Farash, M.S., Nawaz, O., Mahmood, K., Chaudhry, S.A., Khan, M.K.: A provably secure RFID authentication protocol based on elliptic curve for healthcare environments. J. Medical Systems 40(7), 165:1–165:7 (2016), http://dx.doi.org/10.1007/s10916-016-0521-6
12. Hu, Y.C., Perrig, A., Johnson, D.B.: Efficient security mechanisms for routing protocols. Proc Ndss pp. 57–73 (2010)
13. Juels, A.: Rfid security and privacy: a research survey. IEEE Journal on Selected Areas in Communications 24(2), 381–394 (2006)
14. Juels, A., Weis, S.A.: Defining strong privacy for RFID. ACM Trans. Inf. Syst. Secur. 13(1) (2009), http://doi.acm.org/10.1145/1609956.1609963
15. Koh, B.R., Schuster, E.W., Chackrabarti, I.: A.: Securing the pharmaceutical supply chain. Auto-ID Labs, MIT, White Paper pp. 23–28 (2012)
16. Li, Y., Deng, R., Bertino, E.: Rfid. security and privacy. pp. 381 – 394 (1996)
17. Li, Y., Ding, X.: Protecting RFID communications in supply chains. In: Bao, F., Miller, S. (eds.) Proceedings of the 2007 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2007, Singapore, March 20-22, 2007. pp. 234–241. ACM (2007), http://doi.acm.org/10.1145/1229285.1229318
18. Ma, C., Li, Y., Deng, R.H., Li, T.: Rfid privacy: relation between two notions, minimal condition, and efficient construction. In: ACM Conference on Computer and Communications Security. pp. 54–65 (2009)
19. Sarma, S.E., Weis, S.A., Engels, D.W.: RFID Systems and Security and Privacy Implications. Springer Berlin Heidelberg (2002)
20. Sivaranjani, A., Prasad, D.V.: Optimizing bgp performance and a novel routing table structure for fast routing access on multicores. In: International Conference on Communications and Signal Processing (2014)

21. Staake, T., Thiesse, F., Fleisch, E.: Extending the EPC network: the potential of RFID in anti-counterfeiting. In: Haddad, H., Liebrock, L.M., Omicini, A., Wainwright, R.L. (eds.) Proceedings of the 2005 ACM Symposium on Applied Computing (SAC), Santa Fe, New Mexico, USA, March 13-17, 2005. pp. 1607–1612. ACM (2005), http://doi.acm.org/10.1145/1066677.1067041
22. Vaudenay, S.: On Privacy Models for RFID. Springer Berlin Heidelberg (2007)
23. Zhao, M., Smith, S.W., Nicol, D.M.: Aggregated path authentication for efficient bgp security. In: ACM Conference on Computer and Communications Security. pp. 128 – 138 (2010)