Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 784618, 8 pages
http://dx.doi.org/10.1155/2013/784618

Research Article

APP: An Ultralightweight Scheme to Authenticate ONS and Protect EPC Privacy without Cryptography in EPCglobal Networks

Wei Ren,[1,2] Liangli Ma,[3] and Yi Ren[4]

[1] *School of Computer Science, China University of Geosciences, Wuhan 430074, China*
[2] *Shandong Provincial Key Laboratory of Computer Network, Jinan 250014, China*
[3] *School of Electronic Engineering, Naval University of Engineering, Wuhan 430033, China*
[4] *Department of Computer Science, National Chiao Tung University, Hsinchu 30010, Taiwan*

Correspondence should be addressed to Wei Ren; weirencs@cug.edu.cn

Received 21 January 2013; Revised 10 April 2013; Accepted 22 April 2013

Academic Editor: Danny Hughes

EPCglobal network is used to share product data between trading partners, which was proposed by EPCglobal. Object Name Service (ONS) in EPCglobal framework raises two critical security risks: the authenticity of IP addresses for Physical Markup Language (PML) servers and the privacy of Electronic Product Codes (EPCs). Existing work considers either the IP address authentication or the EPC privacy. In addition, that work mainly relies on cryptographic tools, in which key distribution is not a trivial task and also causes a large amount of computation overhead. In this paper, we make the first attempt to solve those two security risks together without relying cryptography. We propose a scheme, namely, APP (authenticate ONS and protect EPC privacy), to guarantee the authenticity of IP addresses for PML servers as well as EPC privacy and to maintain ultralightweight computation cost. Moreover, we give formal definition of the authenticity and the privacy in ONS context. The security achievements are strictly analyzed and proved. The extensive analysis results justify the applicability of the proposed scheme.

1. Introduction

EPCglobal is a typical network framework for the Internet of Things (IoT), machine to Machine (M2M), and RFID networks. It has been envisioned as a key method to recognize, locate, and trace EPC-enabled physical objects (e.g., RFIDs or sensors). Moreover, it is used to facilitate supply chain management, food trace back, logistics, and so forth.

Concretely, EPCglobal relies on Object Name Service (ONS) to map Electronic Product Code (EPC) to an IP address of a server. The server, called Physical Markup Language (PML) server in EPCglobal, provides detailed product information of the EPC. To do so, an EPC tag reader obtains EPCs from tags and submits the EPCs to ONS. Based on the received EPCs, ONS returns the IP address of the corresponding server. Generally speaking, the architecture of ONS consists of distributed server systems and can support iteratively query for scalability and flexibility.

ONS architecture raises two security concerns: one is the authenticity of returning results. If the returning results are fake, the product information will be detoured to a forged server with garbage information. The other is the privacy of EPC. If EPC is revealed by ONS server, the user's privacy may be damaged. For example, a user looks up an EPC for a bottle of medical tablets that is privacy sensitive. Unfortunately, above security risks have not been largely recognized, and rare works exist to address both risks at the same time.

Currently, a few works use some similar methods for DNS security [1], rely on Public Key Infrastructure (PKI) [2], or depend on P2P architecture [3]. Those solutions experience many difficulties: The schemes relying on cryptography usually induce extensive computation overhead. The key distribution and management issues raise many deployment hurdles. The assumption of existing PKI is unrealistic in the current situation. Some solutions such as P2P solution require the migration of underlying network architecture.

In addition, all schemes can only solve either of the aforementioned security concerns and not both. Moreover, as smartphones start to equip RFID reader function, the EPC reader will become portable. To save the power consumption of such hand-held devices, ultralightweight solutions are desired.

In this paper, we propose an ultralightweight solution to authenticate the ONS record and protect the user's privacy without cryptography. In addition, we strictly prove its security strength in terms of authenticity and privacy. Moreover, we adapt a formal and rigorous method to state, present, and analyze the security goals. That is, we formulate the definition of authenticity and privacy in EPCglobal. We formally prove the achievement of proposed scheme with respect to authenticity and privacy strength. All presentations strictly follow the formal expressions for better clarity and rigorous generality.

The contributions of the paper are listed as follows.

(1) We make the first attempt to propose an ultralightweight scheme in terms of computation overhead without cryptography to solve both aforementioned problems in one solution.

(2) We make the first attempt to strictly define authenticity and privacy in EPCglobal and provide formal proofs for the achievement of security goals.

(3) We propose a general scheme to represent all possible solutions for the problem.

The rest of the paper is organized as follows. Section 2 gives an overview on relevant prior work. In Section 3 we discuss the basic assumption and models used throughout the paper. Section 4 provides the detailed description of our proposed models and analysis. Finally, Section 5 concludes the paper.

2. Related Work

The security in ONS starts to attract more and more attention. Fabian and Günther [4] reviewed the security challenges of the EPCglobal network. Sun et al. [2] proposed a lightweight Public Key Infrastructure (LPKI) for trustworthy ONS. They proposed to use a new encryption encode or decode strategy of EPC and improved the reliability of the certificate authority by a new multiple customer relation model. Fabian [3] and Fabian and Günther [5] proposed to use structured P2P systems with distributed hash tables (DHT) to replace ONS architecture. They found that the strength of privacy protection slightly increased by using DHT compared to DNS, but strong protection still relied on secure key distribution mechanisms. Rosenkranz et al. [1] compared two mechanisms to improve the trust level of ONS, DNSSEC and DNSCurve. DNSSEC enables integrity and authenticity; DNSCurve additionally enables confidentiality and higher availability. Their security goals are different from our paper, and ONS security cannot be achieved by DNS security enhancement with optimal performance. Schapranow et al. [6] proposed to protect the privacy of querying parties. Their module can smoothly integrate into existing network infrastructures without major efforts. Kurkovsky et al. [7] proposed to use wearable tags embedded in badges or clothing for employee's tracking at the workplace. It may hurt the privacy of employee after continuous authentication. This kind of privacy problem of RFID has been discussed in many papers [8–11]. Shi et al. [12] proposed SecDS, a secure EPC discovery service system in EPCglobal network. They developed a secure and efficient search engine (SecDS) based on EPC Discovery Services (EPCDS) for EPCglobal network. Their work is independent of ours.

3. Problem Formulation

3.1. Network Model. There exist two major entities in ONS context: requester (denoted as $\mathscr{R}$) and ONS server (denoted as $\mathscr{S}$). The requester reads RFID tag to obtain EPC and submits it to ONS server. The ONS server subsequently returns the IP address of the server who can provide detailed product information on that EPC. The requester then consults the server with returned IP address so as to fetch the detailed product information.

Although the architecture of ONS is very similar to DNS, we observe that there still exists a major distinction between ONS and DNS: the content on the server returned by ONS for a given EPC is usually fixed and shorter than that in the server returned by DNS as it is the information for a product. The content on the server returned by DNS may be changed frequently as it is the information for a web site.

3.2. Attack Model and Trust Model. We only consider adversaries at ONS servers as the paper concentrates on the authenticity of returned records (IP addresses) and EPC privacy. The adversary is denoted as $\mathscr{A}$. We point out following possible attacks.

Definition 1 (ONS pollution attack ($\mathscr{A}_{\text{poll}}$)). ONS server returns a fake IP address upon being requested for an EPC. The PML server at the fake IP address provides forged product information. In shorthand,

$$\begin{aligned} \mathscr{R} &\longrightarrow \mathscr{S} : \{epc\}, \\ \mathscr{S} &\longrightarrow \mathscr{R} : \left\{ip_{epc} \mid \left(epc, ip_{epc}\right) \notin \mathsf{MAP}_{auth}\right\}, \end{aligned} \tag{1}$$

where epc is the requested EPC; ip_{epc} is the IP address of PML server for that epc; MAP_{auth} is an imaginary authenticated list containing correct (epc, ip_{epc}) pairs.

Definition 2 (ONS leakage attack ($\mathscr{A}_{\text{leak}}$)). ONS server reveals the pair of submitted EPC and requester's IP address to other third parties who are interested in them. In shorthand,

$$\mathscr{S} \longrightarrow * : \{epc, ip_r\}, \tag{2}$$

where $*$ means public; epc is the requested EPC; ip_r is the requester's IP address.

Definition 3 (ONS inference attack ($\mathscr{A}_{\text{infr}}$)). ONS server deduces the activities related to submitted EPCs and reveals

those activities and requester's IP address to other third parities who are interested in them. In shorthand,

$$\mathcal{S} \longrightarrow * : \{act, ip_r\}, \tag{3}$$

where $*$ means public; *act* is an activity.

ONS server is untrustworthy as we assume adversaries at ONS server are interested in the user's privacy and intend to break the authenticity. Requester must be trustworthy, as it is a prerequisite requirement for further discussion; otherwise, the discussion is meaningless and no solution exists.

3.3. Security Definition and Design Goal. Informally speaking, the authenticity is guaranteed if adversaries cannot fool the requester to believe a fake IP address. More specifically, we formally state the definitions as follows.

Definition 4 (perfect authenticity of mapping IP address ($\mathsf{Auth}_{\mathrm{prft}}$)). In shorthand, it is

$$\Pr\{\mathcal{R} \text{ believes } (epc, ip) \in \mathsf{MAP}_{auth} \mid \mathcal{R} \longrightarrow \mathcal{S} : \{epc\}, \mathcal{S} \longrightarrow \mathcal{R} : \{ip \mid (epc, ip) \notin \mathsf{MAP}_{auth}\}\} = 0, \tag{4}$$

where $\Pr\{A \mid B\}$ denotes the probability that A happens after event B happens. It is perfect authenticity to defend against $\mathcal{A}_{\mathrm{poll}}$.

Definition 5 (computational authenticity of mapping IP address ($\mathsf{Auth}_{\mathrm{cmpt}}$)). For any probabilistic polynomial turing machine (PPTM) adversary $\mathcal{A}$, given any *epc*, it is computationally infeasible to find ip such that $(epc, ip) \notin R$, but let $\mathcal{R}$ believe it is correct. In shorthand,

$$\Pr\{\mathcal{R} \text{ believes } (epc, ip) \in \mathsf{MAP}_{auth} \mid \mathcal{R} \longrightarrow \mathcal{S} : \{epc\}, \mathcal{S} \longrightarrow \mathcal{R} : \{ip \mid (epc, ip) \notin \mathsf{MAP}_{auth}\}\} < \mathsf{negl}(z), \tag{5}$$

where $\mathsf{negl}(z)$ is a negligible function with security parameter z; $\Pr\{A \mid B\}$ denotes the probability that A happens after event B happens. It is computational authenticity to defend against $\mathcal{A}_{\mathrm{poll}}$.

Definition 6. Authentication attacking experiment on scheme Π defending against adversary $\mathcal{A}$- $\mathsf{ExpAuth}_{\mathcal{A},\Pi}(z)$, is defined as follows.

(1) Scheme Π is executed with security parameter z in the presence of adversary $\mathcal{A}$.

(2) $\mathcal{R}$ sends *epc* to $\mathcal{S}$; $\mathcal{A}$ at $\mathcal{S}$ finds *ip*, where $(epc, ip) \notin \mathsf{MAP}_{auth}$, and sends *ip* to $\mathcal{R}$. $\mathcal{R}$ believes *ip* is a correct *ip*, then outputs 1, otherwise, outputs 0.

(3) If and only if $\mathcal{R}$ outputs 1, the experiment outputs 1.

Definition 7. Scheme Π guarantees perfect (computational) authenticity in the presence of any (PPTM) adversary $\mathcal{A}$ (denoted as $\mathsf{Auth}_{\Pi,\mathcal{A}} = 1$), if and only if for any (PPTM) adversary $\mathcal{A}$ and scheme Π, the probability that the output of authentication attacking experiment equals 1 satisfies

$$\Pr\left[\mathsf{ExpAuth}_{\mathcal{A},\Pi}(z) = 1\right] = 0 \,(\leq \mathsf{negl}(z)), \tag{6}$$

where $\mathsf{negl}(z)$ is a negligible function with parameter z. (In the above equation, the contents in parentheses are corresponding and present simultaneously.)

The EPC privacy defending against $\mathcal{A}_{\mathrm{leak}}$ is guaranteed if and only if adversaries cannot know the requested *epc* as ip_r is always known by adversaries at $\mathcal{S}$. In this situation, *epc* should be either encrypted or transformed.

The EPC privacy defending against $\mathcal{A}_{\mathrm{infr}}$ is guaranteed, if and only if adversaries cannot deduce requester's activities after viewing requested *epc* serials. Note that, it is impossible to hide *epc* from $\mathcal{S}$ as *epc* must be known by $\mathcal{S}$ to return corresponding *ip*. Thus, the privacy requirement is to disturb the requester's activities in *epc* serials. More specifically, we formally state the definitions for EPC privacy as follows:

Definition 8 (user activity). It is a behavior related to certain products that are attached with requested EPCs, denoted as $ACT = \{A_1, A_2, \ldots, A_m\}$, where A_i $(i = 1, \ldots, m)$ is an activity.

Definition 9 (seduce). It links an activity to a serial of EPCs, called SEQ_{epc}. Suppose $SEQ_{epc}(t) = \{EPC_{i+1}, \ldots, EPC_{i+t}\}$, where $EPC_{i+1}, \ldots, EPC_{i+t}$ are t EPCs, $0 \leq i \leq n - t$, $1 \leq t \leq n$, and $|EPC| = n$, $|SEQ_{epc}(t)| = t$. $SEQ_{epc}(t)$ can deduce that activity A_j $(j = 1, \ldots, m)$ happens; This deduction is represented as $(SEQ_{epc}(t), A_j) \in \mathsf{MAP}_{prvc}$, where MAP_{prvc} is a privacy deduction relation set mapping a serial of EPCs to an activity.

Definition 10 (perfect privacy). Simply speaking, adversaries cannot link to anyone in ACT after viewing $SEQ_{epc}(t)$. In shorthand, the perfect privacy is

$$\Pr\left\{\left(SEQ_{epc}(t), *\right) \in \mathsf{MAP}_{prvc} \mid \mathcal{R} \longrightarrow \mathcal{S} : SEQ_{epc}(t) = \left\{EPC_{i+1}, \ldots, EPC_{i+t}\right\}\right\} = 0, \tag{7}$$

where $\Pr\{A \mid B\}$ denotes the probability that A happens after event B happens.

Computational privacy can be defined similarly like computational authenticity.

Definition 11. Privacy attacking experiment on scheme Π defending against adversary $\mathcal{A}$-$\mathsf{ExpPrvc}_{\mathcal{A},\Pi}(z)$, which is defined as follows

(1) Scheme Π is executed with security parameter z in the presence of adversary $\mathcal{A}$.

(2) $\mathcal{R}$ sends $SEQ_{epc}(t)$s to $\mathcal{S}$. If $\mathcal{A}$ finds $A_j \in ACT$, such that $(SEQ_{epc}(t), A_j) \in \mathsf{MAP}_{prvc}$, $\mathcal{A}$ outputs 1, otherwise, outputs 0.

(3) If and only if $\mathcal{A}$ outputs 1, the experiment outputs 1.

Definition 12. Scheme Π guarantees perfect (computational) privacy in presence of any (PPTM) adversary $\mathscr{A}$ (denoted as $\mathsf{Prvc}_{\Pi,\mathscr{A}} = 1$), if and only if for any (PPTM) adversary $\mathscr{A}$ and scheme Π, the probability that the output of perfect (computational) privacy attacking experiment equals 1 satisfies

$$\Pr\left[\mathsf{ExpPrvc}_{\mathscr{A},\Pi}(z) = 1\right] = 0\,(\leq \mathsf{negl}(z)), \tag{8}$$

where $\mathsf{negl}(z)$ is a negligible function with parameter z. (In the above equation, the contents in parentheses are corresponding and present simultaneously.)

Therefore, the design goal is to propose a scheme Π satisfying

$$\mathsf{Prvc}_{\Pi,\mathscr{A}} = 1, \qquad \mathsf{Auth}_{\Pi,\mathscr{A}} = 1, \tag{9}$$

and especially with ultralightweight computation without cryptography.

4. Proposed Schemes

4.1. Basic Schemes. Before we propose our advanced scheme, we review some basic schemes to illustrate our motivations.

(1) Protect Authenticity via Digital Signature. The straightforward method to protect authenticity of EPC is relying on the digital signature. Suppose there exists Trusted Third Party (TTP). TTP signs the signatures for each pair of (epc, ip) with its private key SK_{ttp}. The public key of TTP PK_{ttp} is predeployed at $\mathscr{R}$. The authenticity of EPC can be achieved by following method:

$$\begin{gathered}\mathscr{R} \longrightarrow \mathscr{S} : \{epc\},\\ \mathscr{S} \longrightarrow \mathscr{R} : \left\{ip, Cert = Sign\left((epc, ip), SK_{ttp}\right)\right\},\\ \mathscr{R} : Vrfy\left((epc, ip), Cert, PK_{ttp}\right) \stackrel{?}{=} 1,\end{gathered} \tag{10}$$

where $Cert$ is a certificate or a signature from TTP for (epc, ip); $Sign(\cdot,\cdot)$ is a digital signature function; SK_{ttp} is the secret key of TTP; $Vefy(\cdot,\cdot,\cdot)$ is a signature verification function; PK_{ttp} is the public key of TTP.

This method requires TTP to sign a large number of signatures previously and deploy them to $\mathscr{S}$. It is not scalable and flexible when the number of EPCs is large.

(2) Protect Authenticity via PKI Online. If there exists PKI, the certificate for public key can be fetched, and the signature of TTP can be generated on-line. The authenticity of EPC can be achieved by the following method:

$$\begin{gathered}\mathscr{R} \longrightarrow \mathscr{S} : \{epc\},\\ \mathscr{S} \longrightarrow \mathrm{TTP} : \{(epc, ip)\},\\ TTP \longrightarrow \mathscr{S} : \left\{Cert = Sign\left((epc, ip), SK_{ttp}\right)\right\},\\ \mathscr{S} \longrightarrow \mathscr{R} : \{ip, Cert\},\\ \mathscr{R} : Vrfy\left((epc, ip), Cert, PK_{ttp}\right) \stackrel{?}{=} 1.\end{gathered} \tag{11}$$

This method requires that TTP exists and signs signatures on-line. It may be scalable when the number of EPCs is large, but more delay and communication overhead are induced.

(3) Protect Privacy via TTP's Encryption and Online Decryption. For protecting the privacy of EPC, the straightforward method is via encryption. The database on pairs of (epc, ip) at $\mathscr{S}$ is encrypted by TTP's public key PK_{ttp}. That is, $\mathscr{S}$ possesses a list of $\langle E_{epc}, E_{epcip}\rangle$, where $E_{epc} = Encr(epc, PK_{ttp})$; $E_{epcip} = Encr((epc, ip), PK_{ttp})$; $Encr(\cdot,\cdot)$ is a public key encryption function. The EPC privacy can be achieved by the following method:

$$\begin{gathered}\mathscr{R} \longrightarrow \mathscr{S} : \left\{E_{epc} = Encr\left(epc, PK_{ttp}\right)\right\},\\ \mathscr{S} \longrightarrow \mathscr{R} : \left\{E_{epcip}\right\},\\ \mathscr{R} \longrightarrow TTP : \left\{E_{epcip}\right\},\\ TTP \longrightarrow \mathscr{R} : \{(epc, ip)\}.\end{gathered} \tag{12}$$

This method requires TTP to encrypt a large number of epc and (epc, ip) previously, deploy them to $\mathscr{S}$, and decrypt E_{epcip} on-line. It is not scalable and flexible when the number of EPCs is large. Besides, the EPC privacy protection can only defend against adversaries at $\mathscr{S}$ but not on the links.

(4) Protect Authenticity via P2P Redundancy. If there does not exist PKI or TTP, the authenticity of EPC has to rely on redundancy information that can be provided from P2P network. The authenticity of EPC can be achieved by the following method:

$$\begin{gathered}\mathscr{R} \longrightarrow \mathscr{S}_i : \{epc\},\\ \mathscr{S}_i \longrightarrow \mathscr{R} : \{ip_i, epc\},\\ \mathscr{R} \longrightarrow \mathscr{S}_j : \{epc\},\\ \mathscr{S}_j \longrightarrow \mathscr{R} : \{ip_j, epc\},\\ \mathscr{R} : ip_i \stackrel{?}{=} ip_j,\end{gathered} \tag{13}$$

where $\mathscr{S}_i$, $\mathscr{S}_j$ are any two $\mathscr{S}$s in P2P network; $\mathscr{S}_i$ and $\mathscr{S}_j$ should not be colluded. The privacy protection cannot be achieved in this method.

With the above warmup, we next propose an advanced scheme to achieve the design goal. We list major notations used in the remainder of the paper in Table 1.

4.2. Advanced Scheme: APP. We propose an advanced scheme APP (authenticate ONS and protect EPC privacy)—an ultralightweight scheme for both authenticity and privacy—as follows.

At $\mathscr{R}$ the Following Happens

Step 1. $\mathscr{R}$ has been predeployed by an authenticated set of (epc, ip) pairs. Indeed, the set is a table with two fields—epc and ip—denoted as $EPCIP_k$. We have

$$\forall (epc, ip) \in EPCIP_k, (epc, ip) \in \mathsf{MAP}_{auth}, \left|EPCIP_k\right| = k. \tag{14}$$

Table 1: Notation.

$\mathscr{A}$	Adversary
$\mathscr{R}$	Requester
$\mathscr{S}$	ONS server
$EPCIP_k$	A set of EPC and IP pairs with set size k
EPC_k	A set of EPCs with the set size k
$EPCIP_a$	A set of EPC and IP pairs with set size a
EPC_a	A set of EPCs with the set size a
EPC_{b-a-1}	A set of EPCs with the set size $b-a-1$
$\overline{epc}$	Requested EPC
$\overline{ip}$	IP address corresponding to $\overline{epc}$
EPC_b	A set of EPCs with the set size b
IP_b	A set of IP addresses with the set size b corresponding to EPC_b

All EPCs in $EPCIP_k$ forms a set denoted as EPC_k. That is,

$$EPC_k = \Omega_{epc}\left(EPCIP_k\right), \tag{15}$$

where Ω is projection operation for a field epc in the table $EPCIP_k$.

Step 2. Select (epc, ip) pairs from $EPCIP_k$. The number of pairs is a, which form a testing set $EPCIP_a$. That is,

$$EPCIP_a \subset EPCIP_k, \left|EPCIP_a\right| = a. \tag{16}$$

Similarly, all EPCs in $EPCIP_a$ forms a set, denoted as EPC_a. That is,

$$EPC_a = \Omega_{epc}\left(EPCIP_a\right), \tag{17}$$

where Π is projection operation for a field epc in the table $EPCIP_a$; $|EPC_a| = a$.

Step 3. Suppose the requested EPC is $\overline{epc}$. $\mathscr{R}$ randomly generates $b-a-1$ distinct EPCs that are not in the set EPC_a and do not equal $\overline{epc}$. It is called a set EPC_{b-a-1}. It mixes three sets, namely, EPC_a, EPC_{b-a-1}, and $\{\overline{epc}\}$. The union set is EPC_b. That is,

$$\begin{gathered} EPC_b = EPC_a \cup EPC_{b-a-1} \cup \left\{\overline{epc}\right\} = \left\{epc_1, \ldots, epc_b\right\}, \\ \left|EPC_b\right| = b. \end{gathered} \tag{18}$$

Step 4. $\mathscr{R}$ sends the mixed set EPC_b to $\mathscr{S}$:

$$\mathscr{R} \longrightarrow \mathscr{S} : \left\{epc_1, \ldots, epc_b\right\}; \tag{19}$$

At $\mathscr{S}$ the Following Happens

Step 5. $\mathscr{S}$ searches its database and returns corresponding IP addresses to $\mathscr{R}$:

$$\mathscr{S} \longrightarrow \mathscr{R} : \left\{ip_1, \ldots, ip_b\right\}; \tag{20}$$

At $\mathscr{R}$ the Following Happens

Step 6. $\mathscr{R}$ checks the correctness of IP addresses, namely $IP_a = \{ip_{i1}, \ldots, ip_{ia}\} \subset IP_b$. That is, check whether

$$\left(epc_{ij}, ip_{ij}\right) \in EPCIP_a, \quad \left(j = 1, \ldots, a\right) \tag{21}$$

are satisfied.

Step 7. If all IP addresses in IP_a are correct, $\mathscr{R}$ believes the returning result of $\overline{ip}$. That is, it records IP address for $\overline{epc}$, denoted as $\overline{ip}$.

4.2.1. Extension. (1) The above can be conducted by $\mathscr{R}$ for more rounds. If in all rounds $\overline{ip}$ believes the returning results, the final result will be believed. That is, suppose round number is r; the $\overline{epc}$ is mixed in the final round. In first $r-1$ rounds, the $\overline{epc}$ is a dummy. Only if $\mathscr{R}$ believes $\mathscr{S}$ in first i, $1 \le i \le r-1$ rounds, $\mathscr{R}$ continues the next round (namely, $i+1$ round).

(2) $EPCIP_k$ may be updated by adding item $(\overline{epc}, \overline{ip})$. That is,

$$EPCIP_k \Longleftarrow EPCIP_k \cup \left\{\left(\overline{epc}, \overline{ip}\right)\right\}. \tag{22}$$

The updating can further be extended to batch v ($1 \le v \le b-a$) items that are randomly selected from EPC_{b-a-1}.

(3) The verification for IP address can be extended to the verification of EPC information. In case the IP address corresponding to certain EPC is changed, the verification can be migrated to EPC information. The table $EPCIP_k$ can be extended to table $EPCINFO_k$ accordingly as the information for designated EPC is usually constant. $EPCINFO_k$ is used for authenticating returned IP address. Most processes in above seven steps maintain unchanged, except that the fields in table are changed, and the verification will be delayed upon requesting PML server.

(4) The parameters a, b in scheme APP can be extended to adaptive tuning according to the observation on the trustworthiness of ONS server. If accumulative trustworthiness is over a threshold value, the security parameter a, b can be changed to smaller ones for better performance (with respect to communication overhead).

4.2.2. Discussion. (1) As an EPC is short (no more than 96 bits), it does not obviously damage communication performance when submitting multiple EPCs. Similarly, an IP address is short (no more than 128 bits), and it does not obviously damage communication performance when returning multiple IP addresses.

(2) The above discussion is independent to buffered ONS architecture. If buffered ONS is available, $\mathscr{R}$ does not need to explicitly request $\mathscr{S}$, instead of requesting the buffered ONS. It thus can defend against poisonous ONS buffers. Indeed, buffered ONS records can be looked upon as an imaginary ONS server.

(3) It is better to let authenticated set EPC_a be different in the requests for a given ONS server $\mathscr{S}$. We let $k \gg a$. It does not induce much overhead as the storage of EPC_k

Require: $EPCIP_k, \overline{epc}, \mathcal{S}$
Ensure: EPC_b
$EPCIP_a \Leftarrow RandomSelect(EPCIP_k)$
$EPC_a \Leftarrow TableProject(EPCIP_k, epc)$
$\overline{epc} \Leftarrow ReadTag()$
$EPC_{b-a-1} \Leftarrow RandomGenerateEPC()$
$EPC_b \Leftarrow EPC_a \cup \{\overline{epc}\} \cup EPC_{b-a-1}$
$Send(EPC_b, \mathcal{S})$

Algorithm 1: APP-1 algorithm.

Require: $EPC_b, \mathcal{R}$
Ensure: $\overline{ip}$
$Receive(EPC_b, \mathcal{R})$
$Send(IP_b, \mathcal{R})$

Algorithm 2: APP-2 algorithm.

is lightweight even though k is large. That is, the length of one record in EPC_k is no more than $96 + 128 = 224$ bits; thus, the total length for EPC_k with k records is $224 * k$ bits.

Algorithms proposed for APP scheme are as in Algorithms 1–3.

Analysis

Proposition 13. *The authenticity strength of APP with one round is* $1 - (a!(b-a)!/b!)$.

Proof. If and only if adversaries correctly answer the testing set EPC_a in EPC_b, requesters will accept the returning results. As $|EPC_a| = a$ and $|EPC_a| = b$, the probability that adversaries correctly guess the location of EPC_a in EPC_b is thus $(a!(b-a)!)/b!$. That is the probability that adversaries can cheat requesters to believe a fake returning IP address. Thus, the authenticity strength of APP with one round is $1 - (a!(b-a)!)/b!$. □

Proposition 14. *The authenticity strength of APP with r rounds is* $1 - (a!(b-a)!/b!)^r$.

Proof. The probability that adversaries can cheat requesters in all r rounds is the probability of a successful guess in all r times, which is $(a!(b-a)!/b!)^r$. Thus, the authenticity strength is $1 - (a!(b-a)!/b!)^r$. □

Proposition 15. *The privacy strength of APP with one round is* $1/b$.

Proof. If and only if adversaries correctly guesses the location of $\overline{epc}$ in EPC_b, the privacy will be broken. As EPC_{b-a-1} is randomly generated, the linkages between serial EPCs are blurred. That is, $(SEQ_{epc}(t), A_j) \in \mathsf{MAP}_{prvc}$. As $|EPC_b| = b$, and the privacy strength of one round of APP is $1/b$. □

Proposition 16. *The privacy strength of APP with r rounds is* $1/(b * r)$.

Require: $IP_b, \mathcal{S}$
Ensure: $\overline{ip}$
$Receive(IP_b, \mathcal{S})$
$IP_a \Leftarrow GetIP(IP_b)$
$CORRECT \Leftarrow Check(IP_a)$
if $(CORRECT)$ **then**
$Accept\left(\left\{\left(\overline{epc}, \overline{ip}\right)\right\}\right)$ //go ahead
else
$Abandon(\overline{ip})$
end if

Algorithm 3: APP-3 algorithm.

Proof. Straightforward.

Claim 1. Scheme APP is ultralightweight. □

Proof. The computation overhead for authenticity protection is merely the verification of string comparison; no cryptographic computation is induced. Besides, no computation overhead for privacy protection is induced. The induced communications are $b - 1$ times. As the length of EPCs is no more than 96 bits, and IP address is no more than 128 bits, the total induced extra communication overhead is less than $224 * (b - 1)$ bits. □

If the elements in testing set are recurrent, the security will be damaged. k is also a security parameter influencing the authenticity and privacy strength. For simplicity and security, let $k \gg a$. Otherwise, the following analysis proofs the influence of parameter k in scheme APP.

Proposition 17. *The probability that* EPC_a *is recurrent in two subsequent rounds of selection of* EPC_a *at the requestors is* $1/C(a,k) = (a!(k-a)!)/k!$, *where* $C(a,k)$ *is the combination counts for selecting a elements from k elements.*

Proof. View the selection of EPC_a as an event with probability $1/C(a,k)$. Suppose EPC_a is the set of the first round selection; the recurrence of EPC_a in the next round is thus $1/C(a,k)$. □

Proposition 18. *The probability that x items in* EPC_a *are recurrent in two subsequent rounds of* EPC_a *selection is* $1/(C(x,a) * C(a-x, k-a))$.

Proof. Suppose EPC_a is the set of the first round selection; the recurrence of x items in EPC_a in the next round is $1/(C(x,a) * C(a-x, k-a))$. □

Proposition 19. *The probability that x items in* EPC_a *are recurrent in any y rounds of* EPC_a *selection is* $1/(C(x,a) * C(a-x, k-ya))$.

Proof. Suppose EPC_a is the set of the first round selection; the probability that the recurrence of x items in EPC_a in any y rounds is $1/(C(x,a) * C(a-x, k-ya))$. □

Lemma 20. *Any required strength for authenticity and privacy can be achieved by APP scheme via selecting a proper security parameter (i.e., APP is sufficient for the authenticity and privacy).*

Proof. Suppose the authentication and privacy strength requirements are (α, β). The decision of security parameters for one round of scheme APP is as follows.

(1) Select b such that $1/b < \beta$.

(2) Select a such that $(a!(b-a)!)/b! < \alpha$.

(3) Select $k \gg a$.

□

Proposition 21. *APP scheme can guarantee the authenticity and privacy (i.e.,* $\mathsf{Prvc}_{APP,\mathscr{A}} = 1$, $\mathsf{Auth}_{APP,\mathscr{A}} = 1$*).*

Proof. According to the Lemma, a security parameter (denoted as z in the definition of authenticity and privacy) can be selected for scheme APP to guarantee the required strength for authenticity and privacy. □

4.3. A General Scheme. We finally propose a general scheme to unify all possible schemes to protect authenticity and privacy in ONS context to defend against adversaries in ONS server and channels. The attacks such as ONS pollution attack, ONS leakage attack, and ONS deduction attack can be mitigated.

(1) $epc' \Leftarrow Hide\ (prik1_r, epc)$: it is a computation function at $\mathscr{R}$. $\mathscr{R}$ transforms requested epc to another form epc' by using $prik1_r$. $prik1_r$ is the key privately possessed by $\mathscr{R}$.

Purpose: the privacy of epc is protected via epc'. That is, on viewing of epc', the epc can be correctly guessed with a low probability (namely, a predefined threshold value). Or it is computationally infeasible to compute epc from epc'.

(2) $Request\ (epc', tag_r = f(epc', k2_r), \mathscr{R}, \mathscr{S})$: it is a communication function at $\mathscr{R}$. $\mathscr{R}$ sends epc' and tag_r to $\mathscr{S}$. $k2_r$ is a private key or a secrete key shared between $\mathscr{R}$ and $\mathscr{S}$. That is, $k2_r = \{prik2_r \parallel k2_{rs}\}$. f is a (trapdoor) one-way function with second preimage resistance.

Purpose: for adversaries in communication channels, the authenticity of epc' is protected. That is, it is computationally infeasible to find another $epc'' \neq epc'$, such that $tag_r = f(epc'', *)$, where $*$ is any string. For authenticated channels, tag_r can be omitted. For adversaries in communication channels and at $\mathscr{S}$, the privacy of epc is protected (already explained in the first step).

(3) $\{0 \parallel 1\} \Leftarrow Verify\ (\mathscr{S}, epc', tag_r, k2_s)$: it is a computation function at $\mathscr{S}$. $\mathscr{S}$ verifies the authenticity of epc' via tag_r. The verification needs $k2_s$ that is a public key corresponding to $prik2_r$ or a secrete key sharing with $\mathscr{R}$. That is, $k2_s = \{pubk2_r \parallel k2_{rs}\}$. If and only if the result is 1, $\mathscr{S}$ continues; otherwise, $\mathscr{S}$ terminates.

Purpose: $\mathscr{S}$ authenticates received epc'. For authenticated channels, this step can be omitted.

(4) $ip' \Leftarrow Find\ (\mathscr{S}, epc')$: it is a computation function at $\mathscr{S}$. $\mathscr{S}$ finds the corresponding ip' of epc'. epc' and ip' cannot be unknown with respect to $\mathscr{S}$.

Purpose: $\mathscr{S}$ searches ip' according to epc'.

(5) $Response\ (ip', tag_s = f(epc', k3_s), \mathscr{S}, \mathscr{R})$: It is a communication function at $\mathscr{S}$. $\mathscr{S}$ returns ip' and tag_s to $\mathscr{R}$. $k3_s$ is a private key or a secrete key shared with $\mathscr{R}$. That is, $k3_s = \{prik1_s \parallel k3_{rs}\}$. f is a (trapdoor) one-way function with second preimage resistance.

Purpose: For adversaries in communication channels and at $\mathscr{R}$, the privacy of ip is protected via ip'. That is, on viewing of ip', the ip can be correctly guessed with a low probability (namely, a predefined threshold value). Or it is computationally infeasible to compute ip from ip'.

For adversaries in communication channels, the authenticity of ip' is protected. That is, it is computationally infeasible to find another $ip'' \neq ip'$, such that $tag_s = f(ip'', *)$, where $*$ is any string.

For authenticated channels, tag_s can be omitted.

(6) $\{0 \parallel 1\} \Leftarrow Verify\ (\mathscr{R}, ip', tag_s, k3_r)$: it is a computation function at $\mathscr{R}$. $\mathscr{R}$ verifies the authenticity of ip' via tag_s. The verification needs $k3_r$ that is a public key corresponding to $prik1_s$ or a secrete key sharing with $\mathscr{R}$. That is, $k3_r = \{pubk1_s \parallel k3_{rs}\}$. If and only if result is Y, $\mathscr{S}$ continues; otherwise, $\mathscr{S}$ terminates.

Purpose: $\mathscr{R}$ authenticates received ip'. For authenticated channels, this step can be omitted.

(7) $ip \Leftarrow Recover\ (prik3_r, ip')$: it is a computation function at $\mathscr{R}$. ip is the corresponding IP address for epc. Only $\mathscr{R}$ can recover ip from ip' as only $\mathscr{R}$ possesses $prik3_r$. $prik3_r$ could be equal to $prik1_r$.

Purpose: $\mathscr{R}$ obtains final inquired result ip corresponding to epc.

Next, to simplify the discussion and concentrate on adversaries only at $\mathscr{S}$, we propose a simplified general scheme to unify all possible schemes to protect authenticity and privacy in ONS context to defend against only adversaries in ONS server. The attacks such as ONS pollution attack, ONS leakage attack, and ONS deduction attack can be mitigated.

(1) $epc' \Leftarrow Hide\ (prik1_r, epc)$. It is a computation function at $\mathscr{R}$. $\mathscr{R}$ hides requested epc into another form epc' by using $prik1_r$. $prik1_r$ is the key privately possessed by $\mathscr{R}$.

(2) $Request\ (epc', \mathscr{R}, \mathscr{S})$. it is a communication function at $\mathscr{R}$. $\mathscr{R}$ sends epc' to $\mathscr{S}$.

(3) $ip' \Leftarrow Find\ (\mathscr{S}, epc')$: it is a computation function at $\mathscr{S}$. $\mathscr{S}$ finds the corresponding ip' of epc'.

(4) $Response\ (ip', \mathscr{S}, \mathscr{R})$: it is a communication function at $\mathscr{S}$. $\mathscr{S}$ returns ip' to $\mathscr{R}$.

(5) $ip \Leftarrow Recover\ (prik2_r, ip')$: it is a computation function at $\mathscr{R}$. ip is the corresponding IP address for epc. Only $\mathscr{R}$ can recover ip from ip' as only $\mathscr{R}$ possesses $prik2_r$. $prik2_r$ could be equal to $prik1_r$.

Figure 1 illustrates the processes in general scheme.

Proposition 22. *The APP scheme is an illustration of the simplified general scheme.*

Proof (straightforward). We list the elements in scheme APP corresponding to the elements in the simplified general scheme as follows: $epc = \overline{epc}$, $epc' = EPC_b$, $ip = \overline{ip}$, $ip' = IP_b$, and $prik1_r = prik2_r = NULL$. □

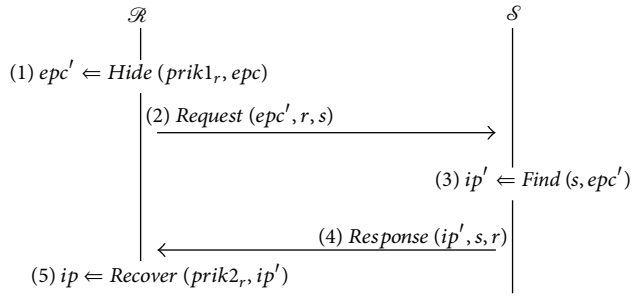

FIGURE 1: A general scheme to protect authenticity and privacy defending against adversaries at $\mathscr{S}$.

Proposition 23. *APP is the necessary condition for authenticity and privacy protection without any cryptographic computation and TTP.*

Proof (sketch). As there does not exist TTP, the authenticity and privacy have to be achieved by $\mathscr{R}$ and $\mathscr{S}$ themselves. As there do not exist cryptographic operations, $priv1_r = prik2_r = NULL$. As $\mathscr{S}$ must know epc to return ip, adversaries at $\mathscr{S}$ can reveal ip. As ip cannot be encrypted, the privacy can be achieved only by requiring multiple EPCs. As there does not exist TTP to judge the authenticity of returning IP_b, an authenticated set EPC_a is required as a self-judgement criteria. That is, $EPC_a \subset EPC_k$ is required to be possessed by $\mathscr{R}$. □

5. Conclusions

In this paper, we proposed an ultralightweight scheme to authenticate requested IP address of EPC and to protect the user's privacy in EPCglobal network without relying on any cryptographic computation or TTP. We also proposed relevant algorithms and a general scheme that can unify all possible schemes. Moreover, the security of the scheme in terms of authenticity and privacy was strictly proved, and the performance was extensively analyzed. Both justified the applicability of the proposed scheme.

Acknowledgments

Wei Ren's research was financially supported by the National Natural Science Foundation of China (61170217), the Open Research Fund from the Shandong Provincial Key Laboratory of Computer Network (SDKLCN-2011-01), Fundamental Research Funds for the Central Universities (CUG110109), and Wuhan Planning Project of Science and Technology (2013010501010144). Yi Ren's research was sponsored in part by the Aim for the Top University Project of the National Chiao Tung University and the Ministry of Education, Taiwan.

References

[1] D. Rosenkranz, M. Dreyer, P. Schmitz, J. Schoenborn, P. Sakal, and H. Pohl, "Comparison of dnssec and dnscurve securing the object name service (ons) of the epc architecture framework," in *Proceedings of the European Workshop on Smart Objects: Systems, Technologies and Applications (RFID Sys Tech '10)*, pp. 1–6, June 2010.

[2] J. Sun, H. Zhao, H. Xiao, and G. Hu, "Lightweight public key infrastructure and service relation model for designing a trustworthy ONS," in *Proceedings of the 8th IEEE/ACIS International Conference on Computer and Information Science (ICIS '09)*, pp. 295–300, June 2009.

[3] B. Fabian, "Implementing secure P2P-ons," in *Proceedings of IEEE International Conference on Communications (ICC '09)*, pp. 1–5, June 2009.

[4] B. Fabian and O. Günther, "Security challenges of the epcglobal network," *Communications of the ACM*, vol. 52, no. 7, pp. 121–125, 2009.

[5] B. Fabian and O. Günther, "Distributed ons and its impact on privacy," in *Proceedings of IEEE International Conference on Communications (ICC '07)*, pp. 1223–1228, June 2007.

[6] M. P. Schapranow, A. Zeier, F. Leupold, and T. Schubotz, "Securing EPCglobal object name service—privacy enhancements for anti-counterfeiting," in *Proceedings of the 2nd International Conference on Intelligent Systems, Modelling and Simulation (ISMS '11)*, pp. 332–337, January 2011.

[7] S. Kurkovsky, E. Syta, and B. Casano, "Continuous RFID-enabled authentication: privacy implications," *IEEE Technology and Society Magazine*, vol. 30, no. 3, pp. 34–41, 2011.

[8] M. Langheinrich, "A survey of RFID privacy approaches," *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 413–421, 2009.

[9] C. Ma, Y. Li, R. H. Deng, and T. Li, "RFID privacy: relation between two notions, minimal condition, and efficient construction," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 54–65, November 2009.

[10] A. Juels and S. A. Weis, "Defining strong privacy for RFID," *ACM Transactions on Information and System Security*, vol. 13, no. 1, article 7, 2009.

[11] Y. Li, R. Deng, J. Lai, and C. Ma, "On two RFID privacy notions and their relations," *ACM Transactions on Information and System Security*, vol. 14, no. 4, article 30, 2008.

[12] J. Shi, D. Sim, Y. Li, and R. Deng, "Secds: a secure epc discovery service system in epcglobal network," in *Proceedings of the 2nd ACM conference on Data and Application Security and Privacy (CODASPY '12)*, pp. 267–274, New York, NY, USA, 2012.