A New Security and Privacy Framework for RFID In Cloud Computing

Süleyman Kardaş*,†, Serkan Çelik*,†, Muhammed Ali Bingöl*,†, and Albert Levi†

*TÜBİTAK BİLGEM UEKAE, Gebze, Kocaeli, Turkey
†Sabancı University, Faculty of Engineering and Natural Sciences, İstanbul, Turkey

***Abstract*—RFID is a leading technology that has been rapidly deployed in several daily life applications that require strong security and privacy mechanisms. However, RFID systems commonly have limited computational capacity and inefficient data management. There is a demanding urge to address these issues in the light of some mechanism which can make the technology excel. Cloud computing is one of the fastest growing segments of IT industry that provides cost effective solutions for handling and using data collected with RFID. As more and more information on companies and individuals is placed in the cloud, concerns are beginning to escalate about just how safe an environment it is. Therefore, while integrating RFID into the cloud, the security and privacy of the tag owner must be considered. Motivated by this, we first provide a new security and privacy model for RFID technology integrated to the cloud computing. In this model, we define the capabilities of the adversary and give the formal definitions. After that we propose a cloud-based RFID authentication protocol to illustrate our model. The protocol utilizes symmetric-key based cryptography. We prove that the protocol achieves destructive privacy according to our model.**

***Keywords*-Cloud Computing, RFID, Security, Privacy**

I. Introduction

Radio Frequency IDentification (RFID) technology has been around for decades. This technology has gained increasing attention as an emerging solution for automatically identifying and/or authenticating remote objects and individuals. RFID based technologies have been rapidly deployed in various daily life applications such as payment, access control, ticketing, and e-passport that require strong security and privacy mechanisms. Security and privacy are two major concerns in these applications when tags are required to provide a proof of identity. The most prominent privacy risk is the tracking of the tag owner, which permits the creation and abuse of circumstantial tag owner profiles. Therefore, an RFID system should provide confidentiality of the tag identity as well as untraceability of the tag owner even the internal state of the tag has been disclosed [13], [17], [21].

Every potential application of RFID systems may require a different approach. As an illustration, manufacturers or wholesales require a full range of compliance-tagging and verification solutions. When working to meet RFID compliance mandates, today's one foremost exigency is the need to implement a scalable solution that not only satisfies but also allows for future growth. Traditional RFID inventory management solutions are expensive for large amount of items, in the sense that they require self server maintenance and significant IT intervention.

Moreover, for some applications multiple read points may be required to track the products throughout workplace. In conventional systems multiple number of databases can be established which cause several operational problems such that synchronization of the databases, expensive system and difficult and separate management. To realize the benefits of RFID, retailers will need to upgrade their IT infrastructure in a number of areas, and their interfaces with other business will have to be closer. The verification of tagged items by RFID systems provides full traceability from sender (e.g. manufacturer) to receiver by maintaining a single database placed in a cloud computing. This provides assurance that a product has been shipped and delivered. This is where cloud computing may come in to provide flexibility to access to the database and authenticate the tagged items/persons. A cloud system can be simply thought of as a server farm that has great computational and storage capacity maintained by the some other operators. In fact, this can greatly reduce the start-up costs as well as the drain that can be put on the IT staff for the RFID system maintainer. Thanks to cloud computing, retailers will not need to upgrade their IT infrastructure.

An RFID system using cloud service as a back-end database and computational capacity is strongly relevant when there is multiple facility providers (such as library, sport center, museum etc.) which are connected to a executive enterprise. In addition, centralizing the above RFID applications and integrating them with an executive systems will require a new level of systems integration capabilities. Using a unified cloud database empowers a single authentication system to more effectively manage pricing, events, reduces inventory losses, expands service offerings, and provides entire RFID infrastructures using a single system. The cloud paradigm provides the ability to offer a single card to each user to get service from multiple applications.

Besides the usability and reachability of cloud computing, the main question is to understand and manage the public concern such as the confidentiality and privacy issues. Therefore some skeptic questions may arise. Can we provide the confidentiality and privacy of the user's data in the public cloud domain? Can we maintain an authentication mechanism by using a far distant cloud service like in our

private database?

In RFID literature, some protocols require exhaustive search on private identity [3], [18] or asymmetric calculation [5], [21] in order to have a strong authentication mechanism. For large systems, these strong private protocols may result in the need of heavy and expensive servers that have fast computational capacity or large storage. Also some efficient authentication mechanisms may have several security flows (for security analysis of such protocols we refer to [2].)

Motivated from the innovations offered by cloud computing, the primary focus of this paper is to propose a security and privacy model for the existing RFID systems melded with the cloud computing paradigm in order to improve the scalability, boost the performance and maintain the security & privacy of whole systems. We first define the system procedures for our new model. Contrary to the previous models [1], [10], [12], [13], [20], [22], we have an additional oracle that an adversary can query the cloud system. Then, the adversary classes are described and we give our security and privacy definitions. Moreover, the readers do not store tag related information but the cloud does. Finally, in order to illustrate our model, we propose an RFID authentication protocol as case study. We prove that the proposal is destructive private according to our model.

The rest of the paper is structured as follows. In Section 2, we introduce our novel privacy model that includes system procedures, adversary oracles and adversary capabilities. We describe the security and privacy definitions with respect to the adversary classes. In Section 3, we propose a privacy preserving RFID authentication protocol which is integrated into a cloud computing service and analyze security and privacy according to our privacy model. Finally, we conclude the paper with a brief discussion in Section 4.

II. Our Privacy Model

Our privacy model borrows and extends the concepts from previous models [12], [22]. In our model, an RFID system consists of a cloud service, many tags, multiple readers where a tag and a reader carry out an authentication protocol by the help of the cloud service. Each tag stores a state, the cloud keeps a database of all tags. Namely, the cloud is the central back-end server. The readers authenticate the tags by the help of the cloud. Adversaries are allowed to interact with all tags and readers and the cloud. Our model is similar to the classical RFID models, which consider many tags, many readers and a back-end server. The main difference between our model and the classical models is that in our model the security and privacy between readers and the server are also considered. The privacy of the tag owners against the server, which is placed in a cloud, is also taken into account. Moreover, the tag related information such as tag owner's information, are stored only in the database of the cloud but not in the reader. Our model does not consider the physical characteristics of the radio links as studied by Danev et al. [9]. Regarding the security of the exchanged messages, our model only considers the content of the messages but not the physical properties.

In this section, we first present the system procedures and the oracles that an adversary can query. Then, the adversary classes are described. Finally, we define our security and privacy definitions.

A. System Procedure

Throughout the paper we modify the common model for RFID systems and use the similar definitions introduced in [8], [22]. An RFID scheme is defined with the following procedures.

- SetupCloud(1^ℓ) : This algorithm generates a public-private key pair (K_{C_P}, K_{C_S}) for cloud where ℓ is the security parameter and initializes its database $\mathcal{DB}$.
- SetupReader(1^ℓ) : This algorithm generates a public-private key pair (K_{R_P}, K_{R_S}) for reader where ℓ is the security parameter and stores its secrets in its non-volatile memory.
- SetupTag$_{K_P}(ID)$: This algorithm generates a tag secret K and the tag identifier ID. If this tag is legitimate, the pair (ID, K) is inserted into the database.
- Ident: An interaction protocol between a tag and a reader to complete the authentication transcript.

B. Adversary Oracles

Privacy is defined as a distinguish-ability game (or experiment Exp) between a challenger and an adversary. This game is defined as follows. The challenger first picks a random challenge bit b and then sets up the system with a security parameter k. Next, the adversary $\mathcal{A}$ is allowed to interact with the system by the help of following generic oracles. First of all, $\mathcal{A}$ creates a new tag of identifier $\text{ID}_\mathcal{T}$. Then, $\mathcal{A}$ interacts with following two collections of oracles.

Definition 1: (Adversary Oracles-I)

- CreateTag($\text{ID}_\mathcal{T}$) : It creates a free tag $\mathcal{T}$ with a unique identifier $\text{ID}_\mathcal{T}$ by using **SetupTag**$_{K_{C_P}}$. It also inserts $\mathcal{T}$ into $\mathcal{DB}$.
- Launch()$\rightarrow \pi$: It makes the reader $\mathcal{R}$ start a new $Ident$ protocol transcript π.
- SendReader$(m, \pi) \rightarrow m'$: This sends the message m to the reader $\mathcal{R}$ in the protocol transcript π and outputs the response m'.
- SendCloud$(m, \pi) \rightarrow m'$: This sends the message m to the cloud $\mathcal{C}$ in the protocol transcript π and outputs the response m'.
- SendTag$(m, vtag)_b \rightarrow m'$: on input vtag, this oracle retrieves the triple $(vtag, \mathcal{T}_i, \mathcal{T}_j)$ from the table D and sends the message m to either $\mathcal{T}_i$ (if $b = 0$) or $\mathcal{T}_j$ (if $b = 1$). It returns the reply from the tag (m'). If the above triple is not found in D, it returns $\perp$.

Table I
THE ADVERSARY CLASSES

Strong	$\Rightarrow$	Destructive	$\Rightarrow$	Weak		Active Insider
$\Downarrow$		$\Downarrow$		$\Downarrow$		$\Downarrow$
Narrow Strong	$\Rightarrow$	Narrow Destructive	$\Rightarrow$	Narrow Weak		Passive Insider

- DRAWTAG$^b(\mathcal{T}_i, \mathcal{T}_j) \rightarrow vtag$: on input a pair of tag references, this oracle generates a virtual tag reference, as a monotonic counter, $vtag$ and stores the triple $(vtag, \mathcal{T}_i, \mathcal{T}_j)$ in a table D. Depending on the value of b, $vtag$ either refers to $\mathcal{T}_i$ or $\mathcal{T}_j$. If $\mathcal{T}_i$ is already references as the left-side tag in D or $\mathcal{T}_j$ as the right-side tag, then this oracle also returns $\perp$ and adds no entry to D. Otherwise, it returns $vtag$.
- FREE$(vtag)_b$: on input $vtag$, this oracle retrieves the triple $(vtag, \mathcal{T}_i, \mathcal{T}_j)$ from the table D. If $b = 0$, it resets the tag $\mathcal{T}_i$. Otherwise, it resets the tag $\mathcal{T}_j$. Then it removes the entry $(vtag, \mathcal{T}_i, \mathcal{T}_j)$ from D. When a tag is reset, its volatile memory is erased. The non-volatile memory, which contains the state S, is preserved.
- CORRUPT$(T_i) \rightarrow S$: It returns volatile and non-volatile memory of the tag T_i.
- RESULT$(\pi) \rightarrow x$: When π completes, returns $x = 1$ if the tag is identified, returns $x = 0$ otherwise.

In our model, we also define two another oracles as follows.

Definition 2: (Adversary Oracles-II)

- CORRUPT$(\mathcal{R}_i) \rightarrow S$: It returns volatile and non-volatile memory of the reader $\mathcal{R}_i$.
- CORRUPT$(Cloud) \rightarrow S$: It returns volatile and non-volatile memory of the cloud.

Definition 3: $(Exp_{S,A}())$ By using the DRAWTAG oracle the adversary can arbitrarily select the tags to interact with. According to the challenge bit b, the system that the challenger presents to the adversary will behave as either the left tags $\mathcal{T}_i$ or the right tags $\mathcal{T}_j$. After $\mathcal{A}$ called the oracles, it outputs a guess bit g. The outcome of the game will be $g \stackrel{?}{=} b$, i.e., 0 for an incorrect and 1 for a correct guess. The adversary wins the privacy game if it can distinguish correctly the left from the right world being executed.
The advantage of the adversary $Adv_{S,\mathcal{A}}(k)$ is defined as:

$$\left| Pr\left[Exp^0_{S,A}(k) = 1\right] + Pr\left[Exp^1_{S,A}(k) = 1\right] - 1 \right|.$$

C. Privacy Classes

Contrary to previous models, our model classify the adversaries as either insider adversary or outsider adversary. The cloud is expected to be the insider adversary who runs the protocol between a legitimate reader and itself correctly, but might save the messages to distinguish the tags. Namely, the cloud is honest but curious during its protocol runs. However, for the outsider adversaries, similar to Vaudenay privacy class [22], we introduce four privacy classes of polynomial-time bounded adversaries, determined by $\mathcal{A}$'s access to RESULT or CORRUPT oracles. These classes are formally defined as follows.

Definition 4: (Adversary Classes) An adversary $\mathcal{A}$ is a p.p.t. algorithm which has arbitrary number of accesses to either the oracles described in Definition 1 or the oracles described in Definition 2.

- **Insider** $\mathcal{A}$ cannot access to any oracles except CORRUPT(Cloud) oracle described in Definition 2.
- **Weak** $\mathcal{A}$ uses only the oracles given in Definition 1 except CORRUPT(T_i) oracle.
- **Destructive** $\mathcal{A}$ uses only the oracles given in Definition 1 but cannot use any oracle on a tag after using CORRUPT(T_i).
- **Strong** $\mathcal{A}$ uses only the oracles given in Definition 1 without any restrictions.
- **Narrow** $\mathcal{A}$ has no access to RESULT oracle.
- **Wide**: $\mathcal{A}$ has access to RESULT oracle.

Remark 1: In a real-life system, **Insider** adversary makes sense when the RFID system owner would like to outsource his/her services to a cloud. In this case, the cloud owner is able to access all the data stored in the cloud and can analyze any interactions with his/her cloud services. Therefore, the system owner may want his/her system to be secure against this attack.

According to the capability of the attacker **Insider** adversary could be two types: passive and active.

Definition 5: (Passive Insider Adversary) A passive **Insider** adversary is one who follows the protocol and does not modify any data but is curious to get some information and may keep all the data and its intermediate computations. In case the adversary is the cloud owner then one may call the cloud owner as *semi-honest* party.

Definition 6: (Active Insider Adversary) An active **Insider** adversary is one who covers the passive adversary and can actively modify the local data or internal computations. In case the adversary is the cloud owner then one may call the cloud owner as *malicious* party.

We also define X^+ and X^* privacy notion variants, where X refers to the basic privacy notion. $^+$ refers to the notion that arises when the adversary has also access to CORRUPT$(\mathcal{R})$ oracle. But * refers to the notion that arises when the capabilities of the adversary are further restricted with respect to CORRUPT oracle. The restricted CORRUPT oracle will only return the non-volatile state of the corrupted party (tag, reader or the cloud) but not the volatile memory state. With this restriction, we exclude trivial privacy attacks on

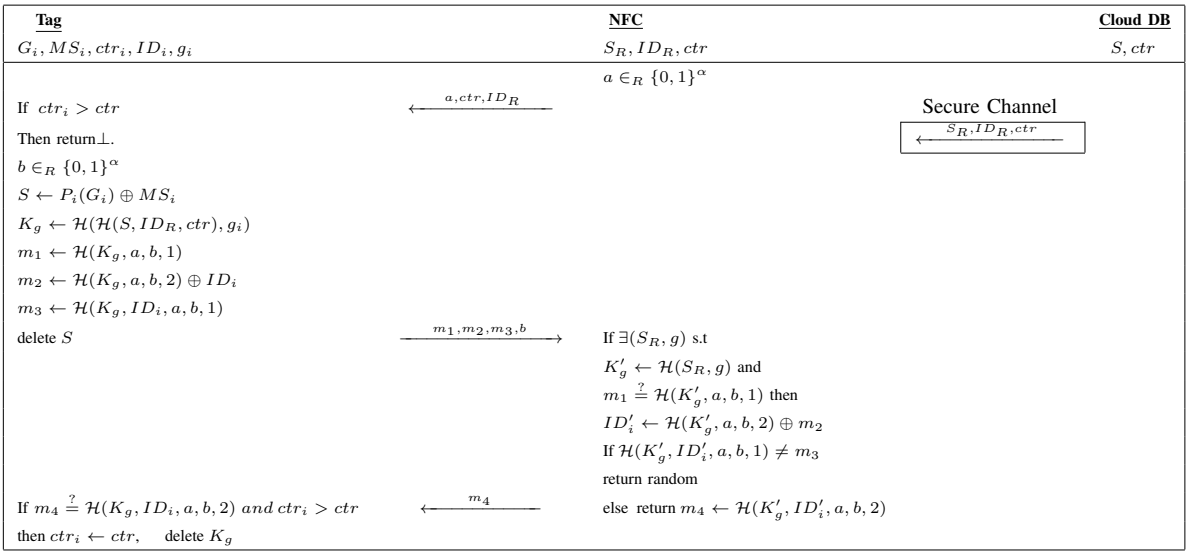

Figure 1. A Destructive Private Authentication Protocol^{+*}.

multi-pass protocols in which the tags are required to store some information in volatile memory during the session of the protocols.

D. Notion of Security and Privacy

Definition 7: (Correctness) An RFID scheme is correct if the identification of a legitimate tag only fails with negligible probability with respect to system's security parameter.

Definition 8: (Tag Authentication) An RFID system achieves tag authentication if for every strong adversary and for every tag in the system, the probability of attacker's impersonating any tag is at most negligible. The adversary may interact with the tag they want to impersonate. The adversary can corrupt all tags but not the impersonated tag.

Definition 9: (Privacy [12]). A privacy preserving protocol, modeled by an RFID system $\mathcal{S}$, is said to computationally provide privacy notion X, provided that for all polynomially bounded adversaries $\mathcal{A}$, it holds that $Adv^X_{\mathcal{S},\mathcal{A}}(k) \leq \epsilon$, for negligible ϵ.

III. Case Study

A. Preliminaries and Notations

The protocol is based on low-cost symmetric primitives such as physically unclonable functions and hash functions. The function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^\alpha$ is a cryptographic hash function, which is treated as random oracle. Namely, the function $\mathcal{H}$ responds to every query with a truly random response chosen uniformly from $\{0,1\}^\alpha$. The function always gives the same response for a given input word.

Moreover, we also use physically unclonable functions (PUF) that are defined as a disordered physical structure implementing a unique function that maps challenges to responses. The responses depend on the nano-scale structural disorder of the PUF, which is assumed to be unclonable or not even reproducible by the PUF's manufacturer. There several types of PUF functions such as optical, coating, delay, SRAM, and etc. [15], [19]. However, in this protocol, we utilize the coating PUF function which is modeled by [16]. The PUF function, which is used in the protocol, is defined as $P : \{0,1\}^\alpha \rightarrow \{0,1\}^\alpha$ where α is the security parameter. The further properties of the PUF function are presented in [14].

B. The Proposal

Let $\mathcal{I}$ be a trusted issuer who sets up the system parameters and the secrets of each party. $\mathcal{I}$ first selects a random master secret $S \in_R \{0,1\}^\alpha$ and creates an counter ctr, which is initially set to zero. The cloud stores the master secret S and the counter ctr. Integration of a reader into system is very simple by just sending a triple $(ID_R, S_R = \mathcal{H}(S, ID_R, ctr), ctr)$ to the reader via a secure channel. $\mathcal{I}$ defines a group size (say l) and creates a counter g which specifies the order of the group a tag belongs to. During the registration of a tag $\mathcal{T}_i$, $\mathcal{I}$ first selects a random unique $ID_i \in_R \{0,1\}^\alpha$, and a random challenge G_i and computes the masked master secret $MS_i \leftarrow S \oplus P_i(G_i)$ and specifies the order of the tag g_i and set its counter $ctr_i \leftarrow 0$. $\mathcal{T}$ stores the values $(MS_i, ID_i, G_i, g_i, ctr_i)$.

The protocol steps are depicted in Figure 2. When a reader (e.g. NFC) $\mathcal{R}$ is connected to the cloud, the cloud sends a triple $S_R \leftarrow H(S, ID_R, ctr)$, $ID_R \in_R \{0,1\}^\alpha$ and ctr to the reader via secure channel. When a tag $\mathcal{T}$ comes in the range of the reader, the reader first chooses a random number $a \in_R \{0,1\}^\alpha$ and sends the triple (a, ID_R, ctr) to $\mathcal{T}$. Then, $\mathcal{T}$ first checks whether ctr is greater or equal to its counter ctr_i. If $ctr < ctr_i$, $\mathcal{T}$ aborts the protocol. $\mathcal{T}$ also chooses another random number $b \in_R \{0,\}^\alpha$, evaluates the

PUF P_i with G_i and XOR it with MS_i to recover master key $S \leftarrow P_i(G_i) \oplus MS_i$. Then, $\mathcal{T}$ computes the session secret $K_g \leftarrow \mathcal{H}(\mathcal{H}(S, ID_R, ctr), g_i)$. Then, $\mathcal{T}$ computes $m_1 \leftarrow \mathcal{H}(K_g, a, b, 1)$, $m_2 \leftarrow \mathcal{H}(K_g, a, b, 2) \oplus ID_i$,$m_3 \leftarrow \mathcal{H}(K_g, ID_i, a, b, 1)$ and sends (b, m_1, m_2, m_3) to the reader. $\mathcal{T}$ deletes S from memory. After that, for all possible value of g, $\mathcal{R}$ computes $m_1' \leftarrow \mathcal{H}(\mathcal{H}(S_R, g), a, b, 1)$ to find a match $m_1' \stackrel{?}{=} m_1$. If a match is found, then $\mathcal{R}$ derives $ID_i' \leftarrow \mathcal{H}(\mathcal{H}(S_R, g), a, b, 2) \oplus m_2$. $\mathcal{T}$ also checks whether the integrity of ID_i' is protected by simple checking the equality of $m_3 \stackrel{?}{=} \mathcal{H}(K_g', ID_i', a, b, 1)$. Now, If every steps are on the right line, $\mathcal{R}$ authenticates the $\mathcal{T}$. $\mathcal{R}$ finally calculates $m_4 \leftarrow \mathcal{H}(K_g', ID_i', a, b, 2)$ and sends it to $\mathcal{T}$. $\mathcal{T}$ checks whether both conditions are hold $ctr > ctr_i$ and $\mathcal{H}(K_g, ID_i, a, b, 2) \stackrel{?}{=} m_4$. If these conditions are hold, then $\mathcal{T}$ updates its counter $ctr_i \leftarrow ctr$. Finally, $\mathcal{T}$ deletes K_g from the memory.

After the reader authenticating the tag, the reader will run a Private Information Retrieval (PIR) protocol with the cloud in order to get the tag related information such as tag owner's photo, birth-date and etc. PIR protocols allow a user to get a data item from a database while hiding the identity of the item being retrieved. In the protocol, the reader simply use ID_i for its query but the cloud will not be aware of it. PIR is out of our scope, so for further details we refer to [4], [6], [7], [11].

Remark 2: Note that whenever a strong adversary tries to apply a physical attack on a target tag, she cannot reach either the valid secret K_g or the valid master secret S. In order to achieve a micro-probing attack on the tag, she should first make a hole on the coating by using Focused Ion Beam. In this case, the structure of the PUF most probably gets a damage that the response of the PUF would be very high level noisy and the PUF control will detect such level of noise and destroys the PUF. The response will not be valid and the master secret S and the session key K_g will not be computed correctly.

C. The Security and Privacy Analysis

In this section, we provide the security and privacy analysis of the protocol depicted at Figure 1.

Remark 3: Throughout this section, one can assume that there is one reader and many tags in the system. There is no loss in the generality with this assumption. To see that, for fixed a and b values, different N_R values produce different K_g values. However, all these K_g values have same randomness (they are indifferent) in the view of the adversary. Thus, the adversary cannot distinguish whether only one or more readers are used in the system. Hence, one NFC is enough for the analysis. Moreover, we use a slightly enhanced version of CREATETAG oracle in the proof of the privacy by adding extra parameter to the function which specifies the group of the tag.

Theorem 1: The proposed protocol satisfies tag authentication against destructive adversary.

Proof: The proof is pretty trivial. Note that the adversary cannot get the values of either K_g or S regardless of how many tags she is allowed to use or corrupt. Moreover, by definition 8. the adversary is not allowed to corrupt the target tag. It is a so low probability that the adversary get the ID of the target tag. Even if this event is realized, the adversary's producing correct m_3 value is at most negligible since reader sends the challenge values a randomly. Thus, the system satisfies tag authentication. ■

Theorem 2: The proposed protocol satisfies destructive privacy.

Proof: The only way for adversary to destroy the privacy is to choose right tags from the same group and left tags from different groups and to expect having the same response to a specified challenge value. First of all, the adversary creates two tags by calling $T_1 =$CREATETAG$(ID_1, 0)$ and $T_2 =$CREATETAG$(ID_1, 1)$ oracles. Then she applies $vtag_1 =$DRAWTAG(T_1, T_2) and uses SENDTAG$(a, ctr, ID_R, vtag_1)$ for l times and stores the answers ${m^i}_{1_1}, {m^i}_{2_1}, {m^i}_{3_1}, {b^i}_1$ where $i \in \{1, \ldots, l\}$. Similarly, the adversary creates another two tags by calling $T_3 =$CREATETAG$(ID_3, 0)$ and $T_4 =$CREATETAG$(ID_4, 2)$ oracles. Then she applies $vtag_2 =$DRAWTAG(T_3, T_4) and uses SENDTAG$(a, ctr, ID_R, vtag_2)$ for k times and stores the answer of the ${m^j}_{1_2}, {m^j}_{2_2}, {m^j}_{3_2}, {b^j}_2$ where $j \in \{1, \ldots, k\}$. If ${b^{i_0}}_1 = {b^{j_0}}_2$ for some i_0 and j_0 but ${m^{i_0}}_{1_1} \neq {m^{j_0}}_{2_1}$ then the answer is the right tags. Otherwise the answer is the left tags. The probability of having wrong result after these observations is negligible. Note that the adversary does not need to create more tags as described above since having more protocol runs with these two tag groups has the same effect of creating new tags and having protocol rounds for the adversary. Therefore, with given parameters the success probability of the adversary is

$$1 - \prod_{i=0}^{k-1} (1 - \frac{l}{2^\alpha - i}).$$

Let $P = \prod_{i=0}^{k-1} (1 - \frac{l}{2^\alpha - i})$, then

$$ln(P) = \sum_{i=0}^{k-1} ln(1 - \frac{l}{2^\alpha - i}) \approx -\sum_{i=0}^{k-1} \frac{l}{2^\alpha - i} > \frac{(k-1)l}{2^\alpha}.$$

So,

$$1 - P < 1 - e^{\frac{(k-1)l}{2^\alpha}}.$$

Note that, the probability above is negligible as k, l are polynomially bounded in α. Thus, the proposed protocol satisfies destructive privacy. ■

Theorem 3: The proposed protocol is resistant against passive insider adversary according to Definition 5.

The correctness of the last theorem is obvious as the cloud does not even know whether NFC has a protocol transaction with any tag at a specified time. In this protocol, the role of the cloud is just initialize the reader for ctr and ID_R values.

IV. Conclusion and Discussion

In this paper, we provide a new security and privacy model for RFID technology, which is integrated into cloud service to leverage the availability and scalability of the system. In this model, we first define the capabilities of the adversary and then give the definitions of the security and privacy. After that we give an example of RFID authentication protocol. Using our privacy model we analyze the sample protocol and proved that the proposal is destructive private.

References

[1] G. Avoine. Adversarial model for radio frequency identification. Technical report, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC, 2005.

[2] G. Avoine, M. A. Bingöl, X. Carpent, and S. B. Yalcin. Privacy-Friendly Authentication in RFID Systems: On Sublinear Protocols Based on Symmetric-Key Cryptography. *IEEE Transactions on Mobile Computing*, 12(10):2037–2049, 2013.

[3] G. Avoine, E. Dysli, and P. Oechslin. Reducing time complexity in rfid systems. In *Proceedings of the 12th international conference on Selected Areas in Cryptography*, SAC'05, pages 291–306, Berlin, Heidelberg, 2006. Springer-Verlag.

[4] L. Ballard, S. Kamara, and F. Monrose. Achieving efficient conjunctive keyword searches over encrypted data. In *Proceedings of the 7th international conference on Information and Communications Security*, ICICS'05, pages 414–426, Berlin, Heidelberg, 2005. Springer-Verlag.

[5] M. A. Bingöl, F. Birinci, S. Kardaş, and M. S. Kiraz. Anonymous RFID Authentication for Cloud Services. *International Journal of Information Security Science*, 1(2):32–42, June 2012.

[6] D. Boneh, G. Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In C. Cachin and J. Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer Berlin Heidelberg, 2004.

[7] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *Proceedings of the 4th conference on Theory of cryptography*, TCC'07, pages 535–554, Berlin, Heidelberg, 2007. Springer-Verlag.

[8] S. Canard, I. Coisel, J. Etrog, and M. Girault. Privacy-Preserving RFID Systems: Model and Constructions. Cryptology ePrint Archive, Report 2010/405, 2010.

[9] B. Danev, T. S. Heydt-Benjamin, and S. Čapkun. Physical-layer identification of RFID devices. In *Proceedings of the 18th conference on USENIX security symposium*, SSYM'09, pages 199–214, Berkeley, CA, USA, 2009. USENIX Association.

[10] R. H. Deng, Y. Li, M. Yung, and Y. Zhao. A new framework for rfid privacy. In *Proceedings of the 15th European conference on Research in computer security*, ESORICS'10, pages 1–18, Berlin, Heidelberg, 2010. Springer-Verlag.

[11] E.-J. Goh. Secure indexes. Cryptology ePrint Archive, Report 2003/216, 2003. http://eprint.iacr.org/2003/216/.

[12] J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel. A new rfid privacy model. In *Proceedings of the 16th European conference on Research in computer security*, ESORICS'11, pages 568–587, Berlin, Heidelberg, 2011. Springer-Verlag.

[13] A. Juels and S. A. Weis. Defining strong privacy for rfid. *ACM Trans. Inf. Syst. Secur.*, 13:7:1–7:23, November 2009.

[14] S. Kardaş, S. Çelik, M. Yildiz, and A. Levi. PUF-enhanced offline RFID security and privacy. *Journal of Network and Computer Applications*, 11(12):1–11, 2012.

[15] S. Kardaş, M. S. Kiraz, M. A. Bingöl, and H. Demirci. A Novel RFID Distance Bounding Protocol Based on Physically Unclonable Functions. In A. Juels and C. Paar, editors, *RFID. Security and Privacy*, volume 7055 of *Lecture Notes in Computer Science*, pages 78–93. Springer Berlin / Heidelberg, 2012.

[16] S. Maubach, T. Kevenaar, and P. Tuyls. Information-theoretic analysis of coating pufs, 2006.

[17] A.-R. Sadeghi, I. Visconti, and C. Wachsmann. PUF-Enhanced RFID Security and Privacy. In *Secure Component and System Identification – SECSI'10*, Cologne, Germany, April 2010.

[18] B. Song and C. J. Mitchell. Rfid authentication protocol for low-cost tags. In *Proceedings of the first ACM conference on Wireless network security*, WiSec '08, pages 140–147, New York, NY, USA, 2008. ACM.

[19] G. E. Suh and S. Devadas. Physical unclonable functions for device authentication and secret key generation. In *DAC '07: Proceedings of the 44th annual Design Automation Conference*, pages 9–14, New York, NY, USA, 2007. ACM.

[20] T. Van Deursen, S. Mauw, and S. Radomirović. Untraceability of rfid protocols. In *Proceedings of the 2nd IFIP WG 11.2 international conference on Information security theory and practices: smart devices, convergence and next generation networks*, WISTP'08, pages 1–15, Berlin, Heidelberg, 2008. Springer-Verlag.

[21] S. Vaudenay. Rfid privacy based on public-key cryptography. In *ICISC 2006. LNCS*, pages 1–6. Springer, 2006.

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