Cloud-based RFID Authentication

Wei Xie[1], Lei Xie[2], Chen Zhang[1], Quan Zhang[1], Chaojing Tang[1]
[1]School of Electronic Science and Engineering, National University of Defense Technology, Changsha, China
[2]State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
xiewei@nudt.edu.cn, lxie@nju.edu.cn, zhan9chen@hotmail.com, {quanzhang, cjtang}@nudt.edu.cn

***Abstract*—Along with the development of cloud computing, cloud-based RFID is receiving more and more attentions of researchers and engineers. However, there is no research in which cloud computing is applied to RFID authentication schemes. Most current works lay emphasis on functionalities, lacking considerations about security and privacy. Classical RFID authentication schemes fail to meet the special security and privacy requirements of cloud-based RFID. The basic postulates of traditional backend-sever-based RFID authentication, i.e. secure backend channel and entirely trustworthy database, are no longer natively tenable in cloud-based RFID scenarios. In this paper, a virtual private network agency is suggested to build secure backend channels and to provide readers with anonymous access to the cloud. The cloud database is structured as an encrypted hash table. The first cloud-based RFID authentication protocol preserving tag/reader privacy to database keepers is proposed. Comparing with classical schemes, the proposed scheme has advantages in deployment cost saving, pervasiveness of authentication, scalability of O(1) complexity to verify a tag, mobile reader holders' privacy preserving, and database security.**

Keywords—RFID; cloud computing; authentication

I. INTRODUCTION

RFID (Radio Frequency Identification) is a wireless technology using radio signals to identify tagged objects automatically and remotely. It has been widely used in supply chain management, inventory control, contactless credit card, and so on.

RFID authentication is a primary approach to secure an RFID system and make it privacy-friendly. Identifying a tag without authenticating it causes serious security issues. Attackers may intercept, manipulate, replay messages from the tag to pretend to hold the tagged object (like an ID smartcard).

There is an extensive literature addressed RFID authentication schemes (see e.g. [1], [2]). Most of them are backend-server-based, in which architecture a reader relays signals from tags to a backend server; and the backend server helps the reader to verify tags according to the backend database. A basic assumption of the architecture is a reliable and always accessible connection between the reader and the backend server, which limits the reader's mobility.

Server-less is another RFID architecture, which is designed to identify and verify tags using offline mobile readers [3-6]. The simple idea of offline authentication is to download an AL (Access List) from a CA (Certification Agency) into an mobile reader. So the reader is enabled to offline authenticate tags anywhere according to the AL without helps of a backend server. A common weakness in server-less protocols is privacy-revealing of reader holders. To the best of our knowledge, there is only one work, which is a server-less searching protocol, specially designed to preserve privacy of mobile reader holders [5]. There are, however, three shortages of the work. (1) Symmetric encryption, which is expensive to a passive tag's limited resource, is required. (2) No server-less authentication protocol is proposed besides the searching protocol. (3) A specially-structured AL is used, making the method inapplicable to enhance other server-less protocols.

Along with the development of cloud computing, cloud-based RFID becomes a new promising architecture [7-14]. Data storage and processing is moved from the backend server to a cloud offering pervasive RFID services. It is accessible using fixed or mobile readers over the internet whenever and wherever needed. There are three advantages of the cloud-based architecture. (1) A verifier with a cloud account is enabled to authenticate its tagged objects using any reader device whenever and wherever the pervasive and customized RFID service is accessible. (2) The pay-on-demand resource deployment greatly fits the needs of medium and small size enterprises. (3) The cloud is more robust than the backend server to serve large-scale applications due to resource sufficiency.

Present works addressed on cloud-based RFID are insufficient in three aspects. (1) Most current works are focused on functionalities, lacking of considerations about security and privacy. (2) There is no research in which cloud computing is applied to RFID authentication. (3) There is no research in which classical RFID schemes are enhanced to meet the special security and privacy requirements of cloud-based systems [15]. For instance, the backend server is truly trusted by readers in traditional RFID schemes. Secrets of tags are stored on the backend server without any encryption; and the backend server is knowledgeable about from which reader and to which tag a session is started. This is unacceptable in a cloud-based application, because the cloud provider is rarely completely-trusted by its clients who use the cloud service. In other words, reader holders and tag owners need to utilize the cloud robustness without losing access anonymity or data privacy. None of current RFID authentication protocols meets the requirements of cloud-based RFID applications.

The main contributions of this paper include: (1) Cloud computing is first applied to RFID authentication scenarios, the first cloud-based RFID authentication scheme is proposed. It inherits pay-on-demand resource deployment, great scalability and pervasively-accessibility from cloud computing, without lacking considerations to protect security and to preserve

privacy. (2) The most important part of the scheme is the first RFID authentication protocol, in which privacy of tags and readers are covered to backend database keepers. The protocol achieves mutual authentication between tags and readers, scalable with O(1) computational complexity of verifying a tag. It is resistant to eavesdropping, manipulating, replaying, and desynchronization attacks.

The remain of the paper is structured as follows: In section 2, we detail the reasons that current RFID authentication schemes are incapable to work in cloud-based scenarios, and provides primary requirements to design a cloud-based RFID authentication scheme. In section 3, the scheme is proposed. It applies a VPN (Virtual Private Network) agency, a global EHT (Encrypted Hash Table), and the first RFID authentication protocol against database keepers. In section 4, the complexity and security of the proposed scheme are analyzed and evaluated by comparing with two representative classical RFID authentication schemes. In section 5, we give our conclusions.

II. Review and Requirement

Traditional RFID authentication schemes are reviewed to show their inapplicability to cloud-based applications in this section. There are mainly two architectures of current schemes: the backend-server-based, and the server-less.

A. Reviews of Traditional RFID

The backend-server-based RFID is illustrated in Figure 1. It is composed of tags, readers, and a backend server. The readers are generally fixed. They identify and verify tags by querying the backend server. Communications between readers and tags are on frontend channels using public radio signals, thus, considered to be insecure. Communications between readers and the backend server are on backend channels which is generally on private intranet, thus, considered to be secure. Backend-server-based protocol designers only need to pay attention to protect frontend communications without worrying about the backend security. However, A major disadvantage of the backend-server-based architecture is the limited mobility of readers because of using private intranet connection to build a secure and always accessible backend channel. It makes the backend-server-based architecture inapplicable to the scenarios in which readers are required to move across cities or even countries.

The server-less RFID is illustrated in Figure 2. It is composed of tags, readers, and a CA. There are two phases: initialization and authentication. The mobile reader accesses the CA and downloads an AL through a secure connection in the initialization phase. Unlike the stationary and well protected backend server, the mobile reader is generally a

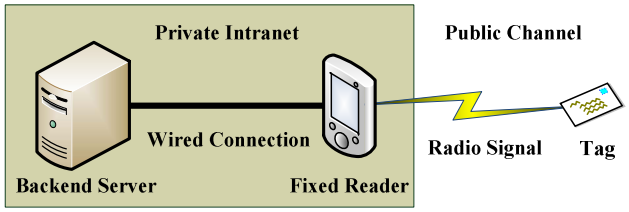

Fig. 1. The backend-server-based RFID architecture

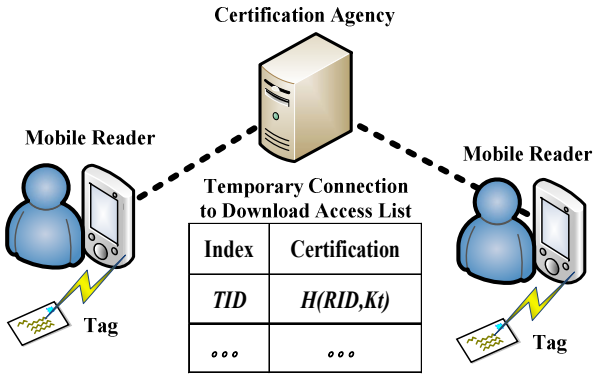

Fig. 2. The server-less RFID architecture

portable device such as a notebook computer, a smart phone, etc. It might be stolen. Then, the AL stored in it might be maliciously utilized to forge tags. The notion of RID (Reader IDentifier) is defined to prevent the forging. Each tag's authentication credential is a hash digest derived from the tag's key and the reader's RID, listed in the AL and indexed by TID (Tag's IDentifier). It makes the AL exclusively useable for the reader. Attackers are infeasible to forge a tag by extracting the tag's key from the reader-specific AL. But as a result, the tag is also infeasible to generate a valid request without the reader's RID. So, the RID is require to be transmitted from the reader to the tag. In the authentication phase, the reader challenges a tag with the RID, waiting the tag response with H(RID, Kt). The reader then searches its AL, trying to find a matched credential to verify the Kt and then identify the TID.

The server-less architecture provides readers with capabilities of large-scale moving and offline authentication. There are, however, three drawbacks in current server-less authentication protocols. (1) All of them transmit RIDs in plaintext, revealing the privacy of mobile reader holders. Attacks can identify and locate the holders by sniffing the uncovered RIDs. (2) Exhaustive searching through the AL are with a computational complexity of O(N), where N denotes the number of tags. It means unsatisfactory scalability to serve a huge number of tags. (3) Computational processes of searching and verifying is executed all by a single personal portable device without any help of a backend server, significantly reducing the performance.

B. Requirements of Cloud-based RFID

The rising cloud-based RFID is illustrated in Figure 3. It is offered as a service of cloud computing to individuals and organizations. It is composed of tags, readers, and a serving cloud. The readers can be fixed or mobile, accessing the cloud by wired or wireless connections. Backend servers are replaced with a pervasive cloud which provides readers with data storing and querying. Comparing with the traditional, the cloud-based RFID has many advantages, meanwhile, is challenged by special concerns over security and privacy. Existing RFID authentication protocols are inapplicable to cloud-based applications because of lacking two primary capabilities.

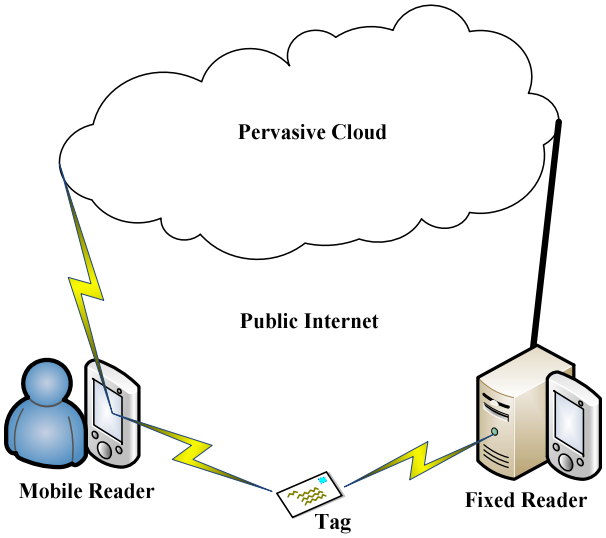

Fig. 3. The cloud-based RFID architecture

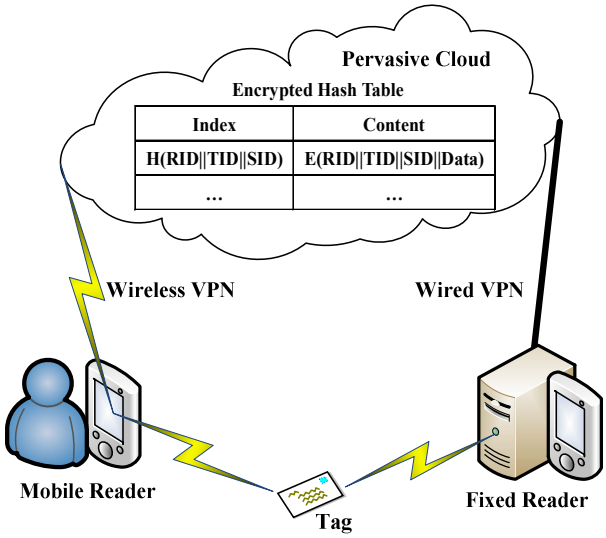

Fig. 4. The proposed cloud-based RFID authentication scheme

Firstly, cloud-based RFID authentication schemes are required to secure backend communications besides protections of the frontend security. In traditional backend-server-based schemes, readers access the backend server through wired private intranet connections, which is considered to be secure. But in the cloud-based schemes, mobile readers often access the public cloud through wireless open internet connections, which cannot be asserted as secure. There are two solutions of this issue. The first solution is to establish VPN connections between readers and the cloud in the network layer. Keeping a constant VPN username, however, harms the anonymity of readers accessing the cloud. The second solution is in the application layer to design an RFID authentication protocol protecting the backend security.

Secondly, cloud-based RFID authentication schemes are required to prevent tags/readers from privacy-revealing to the untrustworthy cloud. A traditional backend server (or the CA) is totally trusted to store secrets of readers/tags without any encryption, and able to trace readers and tags. But in a cloud-based RFID, the cloud provider is not truly trusted by the reader holders and tag owners, therefore, it is required to provide readers/tags with confidentiality of data storage and anonymity of access, which is against the cloud. The detailed requirements are as follows: (1) The database keeper is required to offer data storage service without spying upon reader's/tag's secrets. (2) The database keeper is required to offer data inquiry service without knowing reader's/tag's identities. None of current RFID authentication protocols meet any of the requirements.

III. The Proposed Scheme

The proposed cloud-based RFID authentication scheme is illustrated in Figure 4. Readers anonymously access the cloud through wired or wireless VPN connections. An encrypted hash table is utilized to prevent clients' (readers and tags) secrets from revealing to the cloud. The first RFID authentication protocol preserving readers and tags privacy against an untrusted database keeper is proposed.

A. Notations and Assumptions

Notations in this paper are listed in Table 1. Reasonable assumptions are listed as followed:

- Tags are with middleweight computing capacity of XOR, PRNG, and hashing.
- A mobile reader is embedded in a portable digital device like a PDA with computing capacity of XOR, PRNG, hashing, and symmetric encryption.

TABLE I. Notations

Notation	Description
RID, R	denotes the identity of an RFID reader
TID, T	denotes the identity of an RFID tag
SID, S	numbers authentication sessions between a reader and a tag, starts from zero, with bit length L
M	denotes the last number of sessions (i.e. the maximum SID) between a reader and a a tag.
Data	denotes any data relevant to a session about the user, the reader, and/or the tag.
Nr	denotes a random number generated by an RFID reader
Nt	denotes a random number generated by an RFID tag
PRNG()	denotes the PRNG (Pseudo Random Noise Generation) function of Nr and Nt
H()	denotes a secure one-way hash function with output length L. That is, H(): {0,1}*→{0,1}L
E()	denotes an encryption function by using a symmetric algorithm with a reader private key.
D()	denotes a decryption function by using a symmetric algorithm with a reader private key.
⊕	denotes the bitwise XOR (Exclusive OR) operation
||	denotes the concatenation operation

- The frontend communications between tags and readers are on public radio channels. Attackers are able to eavesdrop, manipulate, delete, replay frontend messages.
- The backend communications between readers and the cloud are through VPN connections. Attackers are able to intercept, block, and resend TCP/IP packets in network layer, however, infeasible to validly parse, modify, or replay the RFID authentication protocol messages in the application layer.
- The cloud provider, i.e. the database keeper, is not trusted. It may be malicious or vulnerable.

B. *VPN Agency*

There are four kinds of participants in the proposed scheme: i.e. tag owner, verifier, VPN agency, and cloud provider. The order of connections is illustrated in Figure 5.

The tag owner and the verifier are frontend participants of the proposed scheme. Tag owners are those own tagged items. Verifiers are reader owners or holders. A tag owner and a verifier can be identical sometimes. For instance, the tag owner is also the verifier in a scenario that a person uses a PDA to identify his/her tagged personal belongings. On the other hand, in a scenario that a club identifies its members by authenticating their smart ID cards, the tag owner is the member to be identified, and the verifier is the club. The security and privacy requirement of a tag-owner/verifier is the infeasibility either to sniff out the TID/RID or to forge the tag's /reader's messages.

The VPN agency and the cloud provider are backend participants of the proposed scheme. The VPN agency provides readers and the cloud with VPN routing. The cloud provider offers a cloud service of RFID authentication to the verifier and the tag owner.

VPN routing makes the communication between the reader and the cloud as secure as in a private intranet. It means that, on one hand, the confidentiality and freshness of the backend communications are ensured. Attackers are able to intercept, block, and resend TCP/IP packets in the network layer, however, infeasible to effectively parse, modify, or replay RFID authentication protocol messages in the application layer. On the other hand, network-layer-anonymity of readers accessing the cloud is achieved. The VPN agency provides a reader with a random virtual IP address each login. A malicious cloud can not links the protocol messages form the same reader based on the packets' source address.

A potential threat caused by anonymous accessing is the difficulty to prevent malicious clients from anonymous storing massive junk data to the cloud, occupying storage space of other clients to perform a DOS attack. However, it is accountable afterwards. The victimized tag owner and verifier can file a suit at the cloud, submitting their attacked storage space address. Then, the cloud provider can find out the attacker's VPN IP address from which the attack starts, according to the accessing logs. Finally, the malicious client can be identified by the VPN agency according to the VPN login logs. The process of revealing malice is a multiparty cooperation based on other research fields not covered in this paper.

The use of VPN routing protects readers' anonymity only in network layer. The corresponding anonymity in application layer (i.e. in the RFID authentication protocol) is provided by the proposed scheme in subsection D.

C. *Encrypted Hash Table*

An EHT is utilized in the proposed scheme to prevent client's data confidentiality and access anonymity from revealing to the untrustworthy cloud provider. Its structure is illustrated in Figure 4. The index which is a hash digest H(RID||TID||SID) uniquely denotes the session with SID from the reader with RID to the tag with TID. According to the one-way characteristic of hash function, attackers as well as the cloud provider are infeasible to parse an index to crack the anonymity of the session. The record indexed by H(RID||TID||SID) is E(RID||TID||SID||Data). It is a cipher text according to a reader-defined encryption algorithm with a reader-managed key. The RID field is used to check the integrity of the cipher text after decryption by the reader. The TID field is used for mutual authentication as a shared secret between the reader and the tag. The SID field is an identifier which is a term in a reader-defined sequence. It can be enumerated to generate all indexes of previous sessions from the reader to the tag. The Data field is customized to store any application data such as location of a tag, access time, account balance in a smartcard, etc. All these data are encrypted by the reader-self to prevent privacy from revealing to the cloud.

Need of special note is, the encryption used in the EHT is not to protect transmitted messages but to protect stored data. There is no key distribution process, because both encryption and decryption algorithm are designed and implemented by a reader-self. The key management can be provided by using any mature approach needless to be detailed in this paper.

D. *The Proposed Protocol*

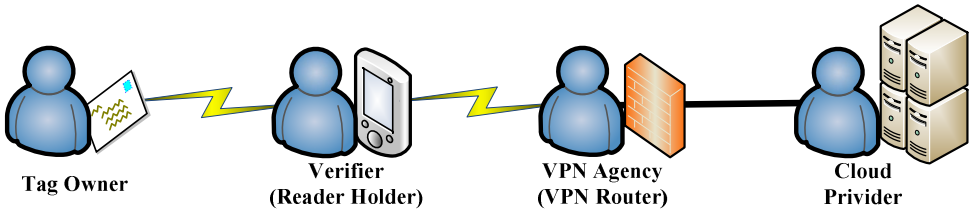

Fig. 5. The participants in the proposed scheme

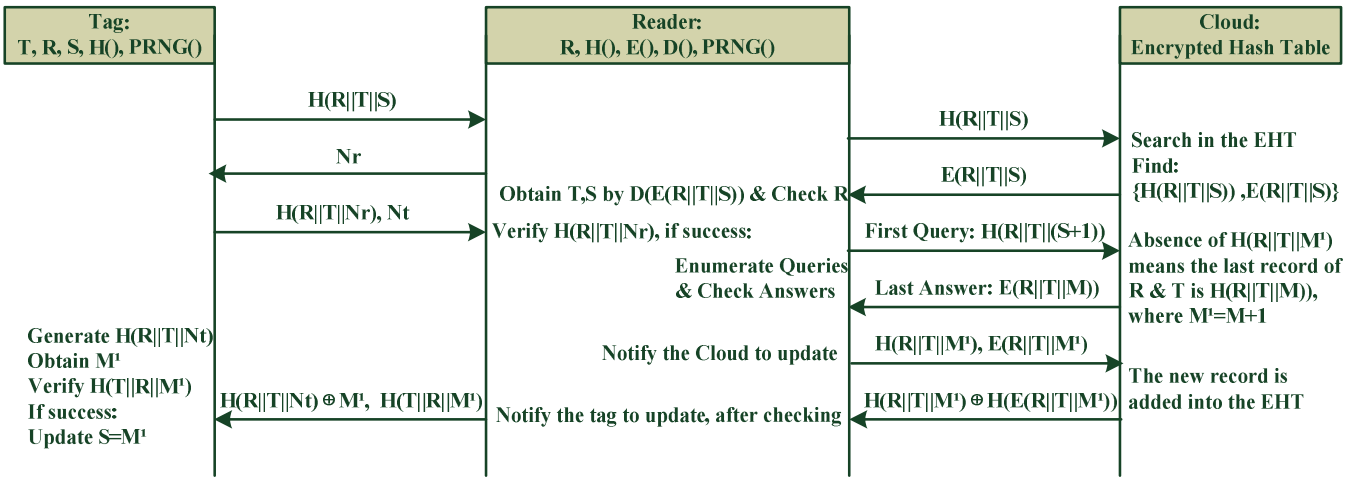

Fig. 6. The proposed cloud-based RFID authentication protocol

The proposed cloud-based RFID authentication protocol is illustrated in Figure 6. E(RID||TID||SID||Data) in the EHT is simplified to E(R||T||S), where R denotes RID, T denotes TID, S denotes SID. The Data field is omitted without any influence on the authentication process. S+1 denotes the next term after the S th term in the reader-defined sequence.

In particular, the RID has been shared by the reader with the tag in a registration phase, in which the reader write its RID and an initialized SID into the tag. The reader also adds an initialized record, i.e.{H(R||T||S), E(R||T||S)}, into the backend EHT in the registration phase. It is tenable to assume the registration is secure due to two reasons. (1) The frontend communication, i.e. the reader initializes the tag, is just once, and thus can be implemented in a closed environment. (2) The backend communication, i.e. the reader initializes the EHT, is actually through the VPN protected networks.

The 1st step of the proposed authentication scheme is for the reader to obtain T and S. The tag generates H(R||T||S) as an authentication request, sends it to the reader. The reader reads the cipher text E(R||T||S) indexed by H(R||T||S) from the cloud EHT, decrypts, and checks the integrity by verifying R, and then obtains T, S.

The 2nd step is to authenticate the tag. The reader generates a random number Nr as a challenge to the tag. The tag calculates H(R||T||Nr) as a response and a random nonce Nt as its challenge to the reader. The reader verifies the response, if valid, the 3rd step is started; otherwise, the protocol is terminated.

The 3rd step is to check the synchronization of S between the tag and the EHT. The reader tries to read the next record indexed by H(R||T||(S+1)) from the EHT, and check the integrity. If there is a valid record , it means that the tag has been desynchronized. The reader continues to try to read the S+2 th record indexed by H(R||T||(S+2)) and so on, until finding the last valid record, assuming its SID is M.

The 4th step is to update the cloud EHT. The reader writes E(R||T||M') with the index H(R||T||M') into the EHT, where M'=M+1. A message of H(R||T||M') ⊕ H(E(R||T||M')) is send back to the reader from the cloud to confirm the updating is successful.

The 5th step is to send a comprehensive response to be verified by the tag. The reader calculate H(R||T||Nt) as the response to the tag's random challenge Nt. The response is also used as an encryption key which is XORed with M' as a simple encryption. The encrypted M', i.e. H(R||T||Nt) ⊕ M', is send to the tag with H(T||R||M') which is also to be verified by the tag.

The 6th step is for the tag to authenticate the reader and repair the desynchronization. The tag calculates H(R||T||Nt) XORed with the received H(R||T||Nt) ⊕ M' to obtain M', then calculates and verifies H(T||R||M'). If success, it means the M' is not modified by attackers, then synchronization is achieved again by updating S=M' on the tag. Moreover, the validity of M' also means the validity of the reader's response H(R||T||Nt) which is XORed together with M', that is, the reader is authenticated by the tag.

IV. Comparison, Analysis, and Evaluation

The proposed scheme is analyzed and evaluated in this section, comparing with two classical RFID authentication schemes. One is Chien et al.'s backend-server-based protocol using tags of EPC C1G2 (Class 1 Generation 2) standard [16]. The other is the first RFID server-less authentication protocol proposed by Tan et al. [3]. These two protocols are very representative, attracting lasting attention till today.

A. Applicability and Complexity

Tags capability requirement. The protocol [16] is lightweight that Gen2 tags only support PRNG, CRC, and bitwise XOR operations, not support hash functions. The protocol [3], which replaces backend database with reader-specific AL, requires tags to support hashing to generate a valid response to the specific reader. So, it is middleweight as well as all later server-less protocols. The proposed scheme using an EHT also requires tags to support hashing, and therefore middleweight. However, using a hash function does not necessarily mean the order of magnitude increasing for a tag. For instance, the research [17] presented a lightweight

hash-function family, suitable for extremely constrained devices such as passive RFID tag.

Readers capability requirement. All operations to authenticate tags are executed by the backend sever in the protocol [16]. The readers are only required to relay messages between tags and the backend server, and support PRNG to defend against replay attacks. Whereas in the protocol [3], all operations to authenticate tags are executed by reader-selves. The readers are required to support hash functions beside PRNG. In the proposed protocol, storing and searching operations are executed by the cloud, meanwhile, other computing operations such as hashing, encryption, decryption, etc, are executed by readers. The readers are required to support symmetric encryption and decryption due to the use of EHT which keeps client's privacy from revealing to the cloud. However, it is not a high requirement for readers. Even mobile readers, which are generally embedded in portable digital device, have adequate capability to support symmetric algorithm like AES, let alone fixed readers with much more hardware resource.

Scalability, i.e. computational complexity to verify a tag. The scalability of an RFID authentication protocol is generally evaluated according to the complexity that a verifier (reader or backend server) identifies a tag. Both the protocols [3] and [16] depend on a brute-force search through the database or the AL to find a matched TID. It makes the computational complexity to verify a tag become O(N), where N denotes the number of tags. It means these protocols are not well scalable in a large-scale application with a huge number of tags. In the proposed protocol, H(R|T|S) as an index is generated by a tag , then sent to a reader. The reader can therefore read the matched record from the EHT by only one time of accurate indexing, instead of launching a brute-force searching through all TIDs, which is generally essential in a O(N) scheme. Thus, the complexity of the proposed scheme is only O(1) which means better scalability than most current RFID authentication protocols.

Offline authentication is to authenticate tags with an offline reader without connecting to a backend database. The protocol [3] is specially designed for offline authentication, meanwhile, the protocol [16] based on a database in a backend server and the proposed protocol based on the EHT in the cloud, cannot work in offline scenarios. The developments of pervasive computing and mobile networking, however, make offline scenarios less and less.

Pervasive (ubiquitous) authentication is to authenticate tags by whatever reader device wherever and whenever, provided that the user logins in the RFID system with a constant username. The protocol [16], like most backend-server-based protocols, depends on private intranet connections to the database. Lacking of large-scale mobility makes the backend-server-based protocols unsuitable to the requirement of pervasive authentication. The protocol [3] replaces the backend database with an AL downloaded into a specific reader. So it is device-related. That is, a user cannot use other reader device to identify his/her tags if the original reader storing the AL is missing. It indicates that the server-less RFID authentication is not ubiquitous too. The proposed protocol utilizes cloud computing in which SaaS (Software as a service) is a kind of user-oriented web service. The cloud-based RFID authentication is unrelated to the used device (reader), user's location, or logging time. It is only related to the login user's identity (RID), therefore, is ubiquitous.

Deployment cost. The RFID system based on a backend server (like in the protocol [16]) has great potential in upgrading hardware to serve large-scale applications. However, the updating deployment is inflexible that resource for the peak load is wasted at idle. The server-less protocol [3] is not suitable for large-scale applications because the system performance depends on a single reader limited in updating hardware. The proposed cloud-based protocol views RFID authentication as a pay-on-demand service. It is applicable to large-scale applications without over-preparing extravagant resource for peak load, especially meeting the needs of medium and small size enterprises.

B. Security and Privacy

Mutual authentication means that a tag authenticates a reader while the reader authenticates the tag. It is useful for access control of tags. The server-less protocol [3] only achieves unilateral authentication. The protocol [16] as well as the proposed protocol achieves mutual authentication which is essential for a protocol based on state-update. Attackers are feasible to successfully launch desynchronization attacks by pretend to be an authorized reader and sending fake state-update messages without mutual authentication.

Resistance against desynchronization. It is fatal to the protocols based on state-update. The protocol [3] has native immunity against desynchronization because it is not state-update based. The protocol [16] has been proved to be vulnerable to desynchronization attacks in [18]. It is untenable to claim that the proposed protocol is immune against desynchronization without formal verification. However, it is tenable to view it as resistant to desynchronization because there are six precautionary and reparative designs as follows. (1) A reader notifies a tag to update its state (i.e. the last SID) after confirming the successful state-update of the EHT. (2) The reader notifies the cloud to update the EHT through a VPN connection, preventing the state-update messages from being manipulated by attackers. (3) The use of PRNG makes it infeasible to replay a previous state-update message to the tag. (4) The tag verifies the state-update message as well as the reader's identity before actual execution of the state-update. (5) Although attackers are able to keep the tag from synchronizing by blocking all state-update messages, the tag is still able to be authenticated by readers. The desynchronization only enables the attackers to trace the tag before repaired. (6) The reader checks the status of synchronization and tries to repair it if needed in each session. To the best of our knowledge, the proposed protocol is invulnerable to all existing types of desynchronization attacks.

TABLE II. COMPARISONS

RFID authentication schemes	Backend server -based [16]	Server-less [3]	The proposed Cloud-based
Tag operation types	PRNG/CRC	PRNG/Hash	PRNG/Hash
Reader operation types	PRNG	PRNG/Hash	PRNG/Hash/AES
Pay-on-demand deployment	No	No	Yes
Offline Authentication	No	Yes	No
Pervasive Authentication	No	No	Yes
Mutual Authentication	Yes	No	Yes
Verification complexity	O(N)	O(N)	O(1)
Desynchronization	Vulnerable	Immune	Resistant
Tag owners privacy	Revealed	Preserved	Preserved
Reader holders privacy	Undefined	Revealed	Preserved
Database Encryption	Not At All	Partial	Entire

Tag privacy requirement is to keep attackers from sniffing tag's secrets or confirming two sessions are related to a same tag. Tag owner's identity and location privacy are probably revealed if the requirement is not met. The research [18] pointed out that the protocol [16] reveals tag keys for the incorrectly use of CRC as a one-way hash function. In the protocol [3] and the proposed cloud-based protocol, hashing TID with a random number provides communications with confidentiality and freshness to preserve tag privacy.

Reader privacy requirement is to keep attackers from sniffing reader's secrets or confirming two sessions are started by a same reader. The mobile reader holder's identity and location privacy are probably revealed if the requirement is not met. The backend-server-based protocol [16] defines no notion of reader identity, all readers are identical and indistinguishable, so that it is meaningless of reader privacy. The server-less protocol [3] simply directly transmits RID in plaintext without any consideration of reader privacy. In the proposed protocol, hashing RID with a random number provides communications with confidentiality and freshness to preserve reader privacy. Moreover, the use of VPN agency provides readers with anonymous connections to the cloud.

Database security is a crucial but widely neglected issue in current RFID researches. The issue is about how to keep secrets of tags/readers secure even if the database keeper (e.g. backend server, CA, or Cloud) is malicious or compromised. The database security in the protocol [16] entirely depends on the assumed-trustworthy and assumed-invulnerable backend server. Although an AL is partly hashed in the server-less protocol [3], the TID field is stored in plaintext. Once the reader stored the AL is stolen, attackers are enabled to trace all tags listed in the reader's AL. In the proposed cloud-based protocol, any data in the EHT is either encrypted or hashed, and the decryption is not executed by the cloud but by the reader-selves. It is infeasible to crack any information of tags without the corresponding reader's key, even if the cloud is malicious or totally compromised by attackers.

The resistances to common attacks (such as eavesdropping, manipulating, replaying, etc.) are essential for an RFID authentication protocol. In the proposed scheme, the backend communications between readers and the cloud are through VPN networks with confidentiality and freshness. Attackers are indeed able to intercept, block, and resend TCP/IP packets in network layer. However, it is harmless, because it is infeasible to validly parse, modify, or replay the protocol messages in the application layer due to the VPN properties. The frontend communications between tags and readers are composed of random challenges and hashed digests (responses). The use of challenge-response technology provides the frontend communications with freshness, defending against replaying attacks. The hash digests are derived from secret RID, TID, SID, with confidentiality, defending against eavesdropping and manipulating. The EHT is composed of hashed and encrypted data, infeasible to be eavesdropped or manipulated. It is indeed potential to invalidly modify an anonymous record in the EHT if the cloud is vulnerable. But it is infeasible to launch this type of DOS attack to a specified reader or tag. And most of these malicious modifications can be repaired by later valid operations.

C. Evaluation

Comparisons between the proposed and the classical schemes are listed in Table 2.

According to the above comparisons, we evaluate the proposed cloud-based RFID authentication scheme as follows. The proposed scheme is middleweight, requiring tags to support PRNG and a hash function. Mutual authentication between readers and tags is achieved without revealing privacy. It is resistant against eavesdropping, manipulating, replaying, and desynchronization attacks.

Comparing to the classical RFID authentication schemes, the proposed scheme's advantages lie in: (1) the pay-on-demand resource deployment greatly meets the requirements of medium and small size enterprises. (2) The cloud-based RFID authentication is offered as a pervasive and customized service unrelated to device, location, or time, provided that the user logins in as a constant username . (3) The computational complexity to verify a tag is O(1), which means the proposed protocol is well scalable to large-scale applications. (4) The proposed scheme preserves mobile reader holders' protocol. (5) The database, which is crucial for RFID security, is encrypted and well protected in the proposed scheme.

However, the proposed scheme has higher requirements for application conditions than the traditional schemes. (1) Readers are required to support symmetric encryption algorithm like AES. The requirement is higher than that for readers in

traditional schemes, but not computational expensive to a common portable digital device. (2) Offline authentication is not achieved in the proposed scheme; however, the internet is pervasive along with the development of mobile networks today.

V. CONCLUSIONS

There are more and more researchers paying attentions to the Cloud-based RFID. However, present works are insufficient in three aspects. (1) Most of them are focused on functionalities, lacking of considerations about security and privacy. (2) There is no research in which cloud computing is applied to RFID authentication schemes. (3) There is no research in which classical RFID schemes are enhanced to meet the special security and privacy requirements of cloud-based applications. Moreover, current RFID authentication schemes are inapplicable to the cloud-based scenarios. It is because that, on one hand, most current schemes are backend-server-based. Wired (for security) and always accessible connections to a intranet are assumed between readers and a backend server. This assumption is not natively tenable in public cloud-based RFID applications. On the other hand, readers and the backend database are belong to the same stakeholder, therefore totally trust each other, or even as an identical participant in the traditional protocols. On the contrary, reader owners and the cloud provider are generally different stakeholders as independent participants in cloud-based RFID applications. Thus, privacy of tags and readers are required to be preserved against not only attackers but also the cloud provider.

The proposed cloud-based scheme adds the participant of VPN agency between readers and the cloud. It provides readers with secure backend channels against attackers and anonymous accesses to the cloud. The cloud database is constructed as an EHT. It prevents private user data from leaking to a malicious cloud provider, or to attackers when the cloud server is compromised. We propose the first RFID authentication protocol which preserves tags and readers privacy against the database keeper. According to comparisons with two classical schemes, i.e. the backend-server-based protocol in [16], and the server-less protocol in [3], the proposed cloud-based scheme has advantages in aspects as follows. (1) The resource deployment is pay-on-demand. (2) The cloud-based service is pervasive and customized. (3) The proposed scheme is well scalable with O(1) complexity to verify a tag. (4) The proposed scheme preserves mobile reader holders' protocol. (5) The database, which is crucial for RFID security, is encrypted and well protected in the proposed scheme.

The future works include: (1) designs of cloud-based authentication protocols which is lightweight in accordance with the C1G2 standard and not based on state-update; (2) designs of cloud-based ownership transfer protocols.

REFERENCES

[1] I. Syamsuddin, et al., "A survey of RFID authentication protocols based on Hash-chain method," in 3rd International Conference on Convergence and Hybrid Information Technology, ICCIT 2008, November 11, 2008 - November 13, 2008, Busan, Korea, Republic of, 2008, pp. 559-564.

[2] R. K. Pateriya and S. Sharma, "The evolution of RFID security and privacy: A research survey," in 2011 International Conference on Communication Systems and Network Technologies, CSNT 2011, June 3, 2011 - June 5, 2011, Katra, Jammu, India, 2011, pp. 115-119.

[3] C. C. Tan, et al., "Secure and serverless RFID authentication and search protocols," Ieee Transactions on Wireless Communications, vol. 7, pp. 1400-1407, Apr 2008.

[4] I.-C. Lin, et al., "Lightweight and Server less RFID Authentication and Search Protocol," in Second International Conference on Computer and Electrical Engineering Proceedings, vol 2, pp. 95-99, 2009.

[5] J. Y. Chun, et al., "RFID tag search protocol preserving privacy of mobile reader holders," IEICE Electronics Express, vol. 8, pp. 50-56, 2011.

[6] C.-F. Lee, et al., "Server-less RFID authentication and searching protocol with enhanced security," International Journal of Communication Systems, vol. 25, pp. 376-385, Mar 2012.

[7] W. Zhao, et al., "SaaS mode based region RFID public service platform," in 3rd International Conference on Convergence and Hybrid Information Technology, ICCIT 2008, November 11, 2008 - November 13, 2008, Busan, Korea, Republic of, 2008, pp. 1147-1154.

[8] T. A. Bapat, et al., "Information-gradient based decentralized data management over RFID tag clouds," in 2009 10th International Conference on Mobile Data Management: Systems, Services and Middleware, MDM 2009, May 18, 2009 - May 20, 2009, Taipei, Taiwan, 2009, pp. 72-81.

[9] J. Muller, et al., "RFID middleware as a service - Enabling small and medium-sized enterprises to participate in the EPC network," in 2009 IEEE 16th International Conference on Industrial Engineering and Engineering Management, IE and EM 2009, October 21, 2009 - October 23, 2009, Beijing, China, 2009, pp. 2040-2043.

[10] C. Dabas and J. P. Gupta, "A cloud computing architecture framework for scalable RFID," in International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, March 17, 2010 - March 19, 010, Kowloon, Hong kong, 2010, pp. 441-444.

[11] Z.-W. Yuan and Q. Li, "Research on data processing of RFID middleware based on cloud computing," in 5th International Conference on Rough Set and Knowledge Technology, RSKT 2010, October 15, 2010 - October 17, 2010, Beijing, China, 2010, pp. 663-671.

[12] A. Chattopadhyay, et al., "Web based RFID asset management solution established on cloud services," in 2011 2nd IEEE RFID Technologies and Applications Conference, RFID-TA 2011, Collocated with the 2011 IEEE MTT-S International Microwave Workshop Series on Millimeter Wave Integration Technologies, IMWS 2011, September 15, 2011 - September 16, 2011, Sitges, Spain, 2011, pp. 292-299.

[13] L. Chu and S.-J. Wu, "An integrated building fire evacuation system with RFID and cloud computing," in 7th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP 2011, October 14, 2011 - October 16, 2011, Dalian, China, 2011, pp. 17-20.

[14] D. Guinard, et al., "Cloud computing, REST and mashups to simplify RFID application development and deployment," in 2nd International Workshop on the Web of Things, WoT 2011, in Conjunction with the 9th International Conference on Pervasive Computing, June 12, 2011 - June 12, 2011, San Francisco, CA, United states, 2011.

[15] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, vol. 34, pp. 1-11, 2011.

[16] H.-Y. Chien and C.-H. Chen, "Mutual authentication protocol for RFID conforming to EPC Class 1 Generation 2 standards," Computer Standards and Interfaces, vol. 29, pp. 254-259, 2007.

[17] J. Guo, et al., "The PHOTON family of lightweight hash functions," in 31st Annual International Cryptology Conference, CRYPTO 2011, August 14, 2011 - August 18, 2011, Santa Barbara, CA, United states, 2011, pp. 222-239.

[18] T.-C. Yeh, et al., "Securing RFID systems conforming to EPC class 1 generation 2 standard," Expert Systems with Applications, vol. 37, pp. 7678-7683, 2010.