A Lightweight Protocol to Robust TID-Based Anti-Counterfeiting

Qi Chai and Guang Gong
Department of Electrical and Computer Engineering, University of Waterloo
Email. {q3chai, ggong}@uwaterloo.ca

March 25, 2010

Abstract

Electronic Product Code (EPC) Radio Frequency IDentification (RFID) tags support a cost-effective anti-counterfeiting mechanism through the tag-specific and globally unique *Transponder ID* (TID). However, with the existence of customizable tags, this countermeasure could easily be bypassed as long as the TID codes are leaked through either physically opening genuine tags or unauthorizedly breaching the backend database. To the best of our knowledge, *physical protection* technologies are exploited to guarantee the confidentiality of the on-tag TIDs, but *cryptographic protections* targeting the security of database-side TIDs does not receive much attention to date.

In this paper, we investigate a new encryption mechanism for the confidentiality of database-side information, yet let it be used in real time without decryption during tag identification/authentication. To this end, a lightweight on-tag cryptographic primitive *Verifiable Cipher* is constructed to achieve a new security property called *Unauthorized Access the backend DataBase* (UADB) resistance. Based on this primitive, we propose a multifunctional protocol to robust the existed TID-based anti-counterfeiting mechanism. The advantages of this protocol, in terms of security properties offered, is then presented. At last, the proof-of-concept implementation on the 4-bit, low-cost and low-power-consumption microcontroller indicates our scheme is feasible for any low-cost passive tags. Besides, the database-side implementation justifies that encryption through *Verifiable Cipher* nearly has no impact on the performance of the database.

Index Terms. Anti-counterfeiting, RFID, Protocol, Encryption.

1 Introduction

The war between counterfeiting and anti-counterfeiting is on-going. Electronic Product Code Class 1 Generation 2 (EPC C1G2) [17], as the de facto standard to support the use of Radio Frequency IDentification (RFID) in today's fast-moving, information rich, trading network, specifies that RFID tags should support a cost-effective anti-counterfeiting mechanism through the tag-specific and globally unique 64-bit unalterable binary stream, which is called Transponder ID (TID). This countermeasure is based on the fact that chips with programmable TID memory are not commercially available. However, prototypes of semi-passive tags such as [37] demonstrate that a tag impersonation device can be easily built from less than ten Euros. With the existence of such customizable tags, the TID-based anti-counterfeiting is effective if and only if the on-tag TIDs and TIDs stored in the backend database are kept confidentially from counterfeiters through their life cycles.

The *physically protection* of tag-side data are intensively investigated. The innovations such as *Physical Unclonable Function* [15, 25] and *Memory Spots* [4] prevent counterfeiters or makes counterfeiters extremely difficult to access the on-tag data. However, to the best of our knowledge, the *cryptographic protection* of database-side information does not receive much attention to date and the security properties offered by current primitives and protocols are inadequate. This deficiency motivates us to investigate a new encryption mechanism to guarantee the confidentiality of database-side data, yet let it be used in real time without decryption during tag identification/authentication.

One should notice that homomorphic encryptions [18, 29], which are heavily relies on public key cryptography such as RSA and pairing, are not applicable to this scenario since the EPC-like tags lacks the capability to executes operations like multiplication in finite field, exponentiation of large numbers. Symmetric searchable encryptions like [8] which are constructed by deterministic encryption through hash functions may not feasible as well, because the deterministic patterns underlined by hash values in the protocol made the tag to be trackable and violates the bearer's privacy, let alone the unaffordable hardware footprint of on-tag hash function [3].

In this paper, we construct the *cryptographic protection* of backend database and propose a protocol to robust the RFID identification and authentication. The proposed solution defeats the conclusion made in [28] regarding database breaches, where an adversary is considered to be capable of cloning tags perfectly if the database containing information of tags is breached.

The remainder of the paper is organized as follows. In Section 2, we introduce the threats and roles concerned in our model and propose the construction of the verifiable-cipher, as well as our protocol. The security analysis of the protocol is given in Section 3. Then we offer the implementation, the performance and the comparison of the provided security properties in Section 4. Section 5 surveys the previous works targeting RFID security and privacy. Finally, a conclusion is made in Section 6.

2 Our Scheme

2.1 Security Model

The many possible threats arises from unprotected wireless communication between RFID readers and tags and they are captured as the adversary model [24, 32, 27] in the design of all lightweight cryptographic primitives. In the model concerned in this paper, a computationally bounded adversary $\mathcal{A}$ does following:

- **Tracking:** Utilizing the linkability between T_i and the bearer, $\mathcal{A}$ could track the bearer if fixed or predictable patterns (i.e., tag's identity) exist in the protocol between T_i and R.

- **Inventorying:** $\mathcal{A}$ wishes to learn the contents of the tag T_i by eavesdropping several legitimate communication sessions between T_i and R or, manipulating a compliant reader $\hat{R}$. $\mathcal{A}$ succeeds if any on-tag contents c_i is learnt.

- **Tag-Impersonation or Skimming:**[*] $\mathcal{A}$ eavesdrops the communication between R and T_i for several sessions and $\mathcal{A}$, with $\hat{R}$, interrogates T_i for another several sessions, $\mathcal{A}$ then writes the recorded flow(s) to a fake tag $\hat{T}$ and utilizes $\hat{T}_i$ to cheat R. $\mathcal{A}$ succeeds if R believes that $\hat{T}$ is T_i during the authentication.

- **Counterfeiting or Cloning:**[†] $\mathcal{A}$ eavesdrops the communication between R and T_i for several sessions and $\mathcal{A}$, with $\hat{R}$, interrogates T_i for another several sessions. $\mathcal{A}$ then physically open the tag and reads the credentials therein. At last, $\mathcal{A}$ creates a fake tag $\hat{T}$ based on the learnt information. $\mathcal{A}$ succeeds if R believes that $\hat{T}$ is T_i during the authentication.

However, the role played by backend database, which is an essential part of any RFID system and where all credential information is stored, is missing in this model. The disclosure of data, often without the administrators' knowledge or control, has become commonplace in the modern society. Possible examples where the database is threaten are:

- A compromised subscriber of EPC service could query all information within its accessible domains.

- An insider (or the insider colludes with the attacker for a cut of the profit), who could easily bypass access control or system auditing, is a sword of Damocles of any information aggregation system [36].

- The massive nature of the data makes it less manageable and unlikely to be confidential through its life cycle.

We conclude these instances to be a new threat towards the RFID system, which is referred to as *Unauthorized Access the backend DataBase* (UADB): $\mathcal{A}$ first reads all tags' information stored in the database, then conducts the aforementioned attacks. Correspondingly, we call them *UADB-Inventorying*, *UADB-Tracking*, *UADB-Tag-Impersonation*, *UADB-Counterfeiting*, where the definition of "success" is the same as before. It is straightforward to envisage that the adversary now could defeat

[*] The object of Tag-Impersonation and Counterfeiting are similar. Tag-Impersonation aims to fool the identification and authentication procedures while Counterfeiting targets even to cheat the validation procedure where the genuinity of tags is tested. In this paper, we indiscriminate the authentication and validation, and use authentication refers to both of them.

[†] For both of Tag-Impersonation and Counterfeiting, the scenario that reader is impersonated is considered, which is viewed as an independent class of attack in some literature.

all previous protocols (i.e., [31, 7, 28, 39, 5, 9]). To eliminate this unexpected information leak, from cryptographic aspect, one may resorts to encryption of backend database. An inherent challenge is that the encrypted information cannot be used in real time for identification or authentication any more.

2.2 Roles

Without loss of generality, the system is composed of one legitimate reader R, n tags $T_1, \ldots ,T_n$ and one backend database. With compliant to [17], each tag is a tiny, passive wireless device which carries EPC code (epc_i), TID code (tid_i) and the PIN (pin_i), and supports: a pseudo random number generator $g : \{0,1\}^l \to \{0,1\}^*$; bitwise operations such as XOR, AND, OR and concatenated inner product (to be defined later).

Throughout this paper, we assume that a trusted initializer (named *manufacturer*), helps to generate the verifiable cipher for the backend database during the *initialization*; the wired communications between reader and database are protected by Transport Layer Security (TLS); the reader behaves honestly; the reader-tag communication is suspicious to aforementioned attacks, launched by an adversary $\mathcal{A}$, who has the right to access the backend database after *initialization*. We also assume that the TID code of each tag is generated uniformly at random and is globally unique. (This could easily be satisfied by inputting the non-randomized *tid* to a hash function or pseudo random sequence generator and taking the output as identity of the tag). At last, the physical protection technologies like [4, 15] are assumed to be applied to the tag which guarantees that the tag-side data is kept confidentially.

2.3 Our Lightweight Primitive – Verifiable Cipher

- $x \in \mathbb{F}_2^l$ is the concatenation of d subsymbols, i.e., $x = x[1] \parallel x[2] \parallel \cdots \parallel x[d]$ where $x[q] \in \mathbb{F}_2^{\frac{l}{d}}$, $d|l$, $q = 1, \ldots, d$.
- The inner product of two vectors in $\mathbb{F}_2^l$, say $x = (x_1, x_2, \ldots, x_l)$, $y = (y_1, y_2, \ldots, y_l)$, is $x \cdot y = \Sigma_{i=1}^l x_i y_i$.
- Let the concatenated inner product of $x \in \mathbb{F}_2^l$ and $y \in \mathbb{F}_2^{\frac{l}{d}}$ be $x \odot y = (x[1] \cdot y) \parallel (x[2] \cdot y) \parallel \cdots \parallel (x[d] \cdot y)$, where $x[q] \in \mathbb{F}_2^{\frac{l}{d}}$ is the q-th subsymbol of x for $q = 1, \ldots, d$.
- The linearity of the concatenated inner product is obvious, $y \odot (x+z) = (y \odot x) + (y \odot z)$, $x, z \in \mathbb{F}_2^l$, $y \in \mathbb{F}_2^{\frac{l}{d}}$.

In this work, the pseudo random number generator $g(x)$, $x \in \{0,1\}^l$ is realized by lightweight stream ciphers of key length l (i.e., Grain [23], WG-7 [30]). Since the key-length-varying pseudo random number generator is demanded, we define the multivariate pseudo random number generator.

Definition 1 *Multivariate Pseudo Random Number Generators are defined as*

- *Bivariate Pseudo Random Number Generator:* $g(x_1, x_2) = g(g(x_1) \oplus x_2)$*,* $x_1, x_2 \in \{0,1\}^l$*.*
- *Trivariate Pseudo Random Number Generator:* $g(x_1, x_2, x_3) = g(g(g(x_1) \oplus x_2) \oplus x_3)$*,* $x_1, x_2, x_3 \in \{0,1\}^l$*.*
- *The defined multivariate pseudo random number generators are non-associative, i.e.,* $g(x_1, x_2) \neq g(x_2, x_1)$*, which makes them resistant to Algebraic Replay Attack [14].*
- *The security of the defined multivariate pseudo random number generators are provably reducible to the security of* $g(x)$*.*

During the *Initialization*, without the present of $\mathcal{A}$, the manufacturer generated sequences $s_i \in \mathbb{F}_2^l$, $u_i \in \mathbb{F}_2^{\frac{l}{d}}$, $v_i \in \mathbb{F}_2^d$, $i = 1, \ldots, n$ uniformly at random. For $i = 1, ..., n$, the manufacture loads the i-th tag with the tuple $(epc_i, tid_i$, pin_i, u_i, $v_i)$. Next, the manufacturer uploads all n tuples $(epc_i, \{tid_i\} := tid_i \oplus s_i$, $y_i := s_i \odot u_i \oplus v_i$, u_i, $E_{pk}(pin_i || c_i))$, $i = 1, ..., n$ to the database, and destroys all intermediate data. Here $E_{pk}(pin_i || c_i)$ denotes a public-key encryption of the PIN pin_i and the tag's content c_i using the reader's public key pk.

At the beginning of the *Running* stage, $\mathcal{A}$ could freely access the backend database and may download all n tuples. After this, $\mathcal{A}$ leaves the database. Later on, the reader initializes a communication with a tag and forwards the tag's response $p := tid_i \oplus n$, $\delta = n \odot u_i \oplus v_i$ to the database, where $n \in \mathbb{F}_2^l$ is chosen uniformly at random by the tag. The database then starts to distinguish tid_i by searching over all the encrypted data. To be exact, for $j = 1, \ldots, n$, it executes $c_j := \{tid_j\} + p$, $c_j \in \mathbb{F}_2^l$ and calculates the intermediate flag $\epsilon \in \mathbb{F}_2^d$ by $\epsilon := c_j \odot u_j$. The server then tests the equality between ϵ and $y_j + \delta$. If it is true, the tag tid_i is successfully identified as tid_j; If for all n tuples, the database cannot find a match, the procedure then stops with failure. We denote this procedure as $vcsearch(p, \delta)$.

From the descriptions above, it is clear that the correctness of this lightweight primitive inherits from linearity of concatenated inner product and the security relies on the onewayness of inner product and the one-time-pad like encryption (i.e, $\{tid_i\} := tid_i \oplus s_i$, $y := s_i \odot u_i \oplus v_i$ where s_i, u_i, v_i are tag-varying, independent random sequences).

2.4 The Multifunctional Protocol

Note that the primitive itself is malleable [16]. To be specific, assume $\mathcal{A}$, through accessing the backend database, has the tuple $(epc_t.tid_i \oplus s_i, s_i \odot u_i \oplus v_i, u_i, E_{pk}(pin_i)$ for tag T_i. $\mathcal{A}$ then loads the a customizable tag $\hat{T}_i$ with $(epc_i, tid_i \oplus s_i, s_i \odot u_i \oplus v_i, u_i)$ and let it emits the response $\hat{p} := x_i \oplus s_i \oplus n, \hat{\delta} = s_i \odot u_i \oplus v_i \oplus n \odot u_i = (s_i \oplus n) \odot u_i \oplus v_i)$. Not surprisingly, the database will identify $\hat{T}_i$ as T_i in this case.

Table 1: A Multifunctional Protocol

Tag T_i		Reader R		Database
randomly pick n_T	$\xleftarrow{(identification,n_R)}$	randomly pick n_R		
$p := g(epc_i, n_T, n_R)$	$\xrightarrow{(p,n_T)}$		$\xrightarrow{(p,n_T \| n_R)}$	search for epc_j such that
				$g(epc_j, n_T, n_R)$ equals p
		decrypt to get pin_i	$\xleftarrow{E_{pk}(pin_i \| c_i)}$	if success, response
randomly pick n'_T	$\xleftarrow{(authentication,n'_R)}$	randomly pick n'_R		
$p' := tid_i \oplus n'_T$				
$e := g(n'_R, pin_i)$				
$\delta' := (n'_T \oplus e) \odot u_i \oplus v_i$	$\xrightarrow{(p',\delta')}$	$e' := g(n'_R, pin_i)$	$\xrightarrow{(p' \oplus e',\delta')}$	$vc(p' \oplus e', \delta')$
			$\xleftarrow{success}$	if success, response

We show the proper use of this primitive by presenting a multifunctional protocol in Table 1, which satisfies the needs of tag identification and authentication. In identification procedure, the reader, cooperating with backend database, tries to locate an unique identity like EPC and TID, claimed by the tag. This is equal to a query-search process in the information system. Secondly, the reader challenges the tag with the information it holds to test if the claimed identity is true, which is known as authentication. In our protocol, counterfeiting is thwarted by authentication.

The memory cost is $3l + \frac{l}{d} + d$ per tag, assume epc_i, tid_i, pin_i, u_i, v_i are to be loaded into tag's memory. For instance, $l = 64$, $d = 8$, the memory cost is 272 bits/tag. Essentially, memory is no longer a limited resource as it was. The fact that current EPC tags are almost equipped with 256-bit to 2048-bit user programmable memory underlines that our protocol is implementable for off-the-shelf EPC tags. On the other side, with the same l and d, the database-side storage is 272 bits/tag, or equivalently 3×10^7tags/GB. Here we ignore the storage cost by tags' content c_i since this value is application-specific.

2.5 Multiple Reader Scenario

The public encryption $E_{pk}(pin_i||c)$ using R's public key pk rises an nature question: if multiple readers are involved in the model, whose public key should be utilized for this encryption? We next solve this problem by introducing the concept of clustered readers.

Concerning the fact that to enable the manageability of the supply-chain (including suppliers, intermediaries, third-party service providers, and customers), the tagged products are distributed through predictable channels as well as laid out on the shelves for customers at specific stores. Stated in another way, the roaming range of each tag is limited, which can be represents as an aggregation of certain readers. (A tag out of its “territory” is very likely to be a counterfeited one, which is an application layer strategy for anti-counterfeiting.) Following this fact, the readers in our model are divided into different clusters such that readers in one of such clusters have the access to the same subset of tags which is designed by the manufacture at the very beginning. Without loss of generality, let $R_1, .., R_z$ be one cluster which accesses a subset of tag $\{T_1, ..., T_n\}$. Next, each cluster elects a head, say R_0 using reputation-based ranking algorithms, concerning the aspects like computational capability, communication bandwidth, security mechanisms equipped, online time, etc.. Existed schemes surveyed in [26] could serve for this purpose.

We now ready to change our protocol to solve the aforementioned problem. During the initialization, the manufacture encrypts PIN and content of each tag by R_0's key, say $E_{k_{R_0}}(pin_j||c_j)$. During the running stage and when reader $R_i, i > 0$ is online, it first negotiates with R_0 a fresh session key k_i using Internet Key Exchange (IKE) protocol, and R_i interrogates T_j using the proposed protocol. One slightly different is, once the records is located, the database responses to the head R_0 with $(R_i, E_{k_{R_0}}(pin_j||c_j))$. The head next does an encryption $E_{\frac{k_i}{k_{R_0}}}(E_{k_{R_0}}(pin_j||c_j)))$ and sends the result to R_i. Now, R_i receives and decrypts it using the session key k_i and gets pin_j and c_j. Note that the encryption employed here satisfies the following property

$$E_{k_1}(E_{k_2}(x)) = E_{k_1 \cdot k_2}(x)$$

This is instantiated by PH-SAEP+ Encryption [35]: a Pohlig-Hellman encryption with SAEP+ padding [1].

Definition 2 *PH-SAEP+ Encryption. PH-SAEP+ is a triple $\{KEYGEN, E_{k_1}, D_{k_2}\}$ such that*

- $KEYGEN(1^n) = (p, k_1, k_2)$ *where p is a prime that is publicly known and k_1 is a secret key and k_2 is its multiplicative inverse.*

- $E_{k_1}(x) = (((x||h_2(x||r)) \oplus h_1(r))||r)^{k_1}$ *mod p, where r is chosen at random, $h_1 : \{0,1\}^{|r|} \Rightarrow \{0,1\}^{|x|+s}$ and $h_2 : \{0,1\}^{|x|+|r|} \Rightarrow \{0,1\}^s$ are two hash functions, with security parameter s.*

- $D_{k_2}(E_{k_1}(x)) = (E_{k_1}(x))^{k_2}$ *mod* $p = c||r$. *Extract r and produce $x||\delta = c \oplus h(r)$, and verify whether $h_2(x||r)$ equals to δ or not. If yes, set the output to be x.*

- *The security of PH-SAEP+ relies on the fact (or assumption) that discrete logarithm problem is hard, namely ,given P, $c = x^{k_1}$ mod p, the probability that there exists an probabilistic polynomial time algorithm which could compute $k_1 = log_x c$ mod p is negligible. The detail proof of CCA security of PH-SAEP+ is in [35].*

3 Security analysis

We now consider how our protocol performs in the presence of adversary $\mathcal{A}$. Apparently if the protocol is resistant to *UADB-Inventorying*, it naturally resists to *Inventorying*. For reasons of space and clarity, our analysis focuses on UADB-enabled attacks. The analysis towards conventional attacks is offered, if and only if the resistance to UADB-enabled attacks is not provided.

3.1 Tracking

Provided $\mathcal{A}$ has a replica of the all n tuples, $\mathcal{A}$ could identify the tag T_i by search for epc_j such that $g(epc_j, n_T, n_R)$ equals p, which is exactly what the database does during the identification phase of the protocol. This security property is not achievable because $\mathcal{A}$ is equivalent to the backend database in this trivial case.

Without UADB, it is straight forward to see that, in our protocol, the output of T_i all masked by random numbers, which are contributed by both T_i and R. If the freshness property (to be defined later) is satisfied, then the probability of the adversary guessing epc_i is negligible. For example, even the reader's challenge n_R is fixed, the tag's response $g(epc_i, n_T, n_R)$ is always different due to the randomness offered by n_T. At $\mathcal{A}$'s point of view, there is no linkability between the tag and bearer, thus our protocol is resistant to *Tracking*.

3.2 Inventorying

For the similar reason as of UADB-Tracking, UADB-Inventorying is not achievable. For the inventorying attack, recall the fact in the protocol, the identity is masked by session-independent random numbers. In addition, the tuples stored in backend database are encrypted (i.e., $E_{pk}(pin_i||c_i)$) and do not leak information about the tag's contents even been downloaded by A. Thus our protocol is resistant to inventorying.

3.3 Tag Impersonation or Tag Replay or Skimming

To impersonate T_i, $\mathcal{A}$ makes use of epc_i, $\{tid_i\}$, y_i, u_i and all information learnt by eavesdropping. $\mathcal{A}$ is now able to create the half-fledged $\hat{T}_i$ which could cheat the identification procedure. However, since $\mathcal{A}$ cannot decrypt $E_{pk}(pin_i||c_i)$, $\hat{T}_i$ produces an illegitimate response like $\hat{\delta}' = (n'_T \oplus \hat{e}) \odot u_i \oplus v_i$, $\hat{e} \neq e$. Besides, this reader, acted like a man in the middle of tag-database communication, will xor p' with e, which makes $(p' \oplus e, \hat{\delta}')$ no longer a pair. As a result, the probability of success when running $vcsearch(p' \oplus e, \hat{\delta}')$ on the database is negligible. In all, our protocol is resist to *UADB-Tag-Impersonation.*

3.4 Counterfeiting or Cloning

In this scenario, $\mathcal{A}$ may consider advanced and complicated attacks in this case, such as algebraic attack for RFID protocols [14] which is essentially solving a small system of equations.

Based on the learnt information, $\mathcal{A}$ constructs a small system of equations like (1) where u_i is known and tid_i, v_i, e_j, $n'_{T,j}$ are unknown. (e_j, $n'_{T,j}$ represents the value of e, n'_T in the j-th session). Note that (1) is intractable since in each new session, when two more such equations are imported to this system of equations, two unknowns, say e_j, $n'_{T,j}$, are also involved. This system of equations is always non-full rank and unsolvable.

$$\begin{cases} tid_i \oplus n'_{T,j} & j = 1, ..., m \\ (n'_{T,j} \oplus e_j) \odot u_i \oplus v_i & j = 1, ..., m \end{cases} \tag{1}$$

Formally, our protocol has the following two necessary properties defined in [14] to be immune of the generalized algebraic attack, which are

- **Freshness**: given nonce r_2, secret key s, the adversary's advantage in correctly guessing $g(r_1, r_2, s)$ for an unknown, randomly chosen nonce r_1 must be negligible. The pseudo random number generator g in our protocol, which is realized by lightweight stream ciphers, satisfies this requirement. Suppose pin_i is a secret whilst $n'_{R,i}$ is sensed by $\mathcal{A}$, $e_i := g(n'_{R,i}, pin_j)$ is still unpredictable for $\mathcal{A}$.

- **ARR, Algebraic Replay Resistance** Let $O_s(x)$ be an oracle which upon input x randomly chooses y and returns y and $g(x, y, s)$. If s is unknown, then given access to a polynomial number of queries $O_s(x_1)$, ..., $O_s(x_L)$ to the oracle, it is infeasible to compute $g(r_1, r_2, s)$ for a given $r_1 \notin \{x_1, ..., x_L\}$ and any r_2. The ARR property guarantees that there is no efficient algorithm to compute a response r_2, $g(r_1, r_2, s)$ to the challenge r_1 even after having observed previous challenge-response pairs. In our protocol, the non-associative property and onewayness of the multivariate pseudo random number generators exhibit ARR.

Through the analysis, our protocol is resist to *UADB-Counterfeiting*. Taking into account the very lightweight computation here, this protocol could be used to robust TID-based anti-counterfeiting in EPC C1G2 standard.

3.5 Other Security Properties

Denial of Service (DoS) Resiliency: DoS attack is generally considered when secret values updating and synchronization between tags and database are introduced in protocol design. Our protocol is secure against DoS attack because no synchronization is needed. From another point of view, the complexity of *vcsearch* is linear to the number of tuples stored, which makes the query-intensive DoS attack infeasible.

Forward/Backward secrecy: An attacker cannot relate any previous/future readings of a tag (or set of tags) to a further/previous captured tag (or set). In order to achieve forward and backward secrecy, a stronger pseudo random number generator should be used [40], which might be too luxury for current low-cost passive tags, thus beyond the scope of this paper.

As a summary, the comparison of security features provided by previous protocols and our protocol is shown in Table 2. See Section 4 for the protocols cited.

Table 2: Comparison of Security Properties

	(a, a') [a]	(b, b') [b]	(c, c') [c]	(d, d') [d]
EPC-Gen2 [17]	×, ×	×, ×	×, ×	×, ×
C-D [7]	✓, ×	×, ×	×, ×	×, ×
Q-Y-Y [10]	✓, ×	×, ×	×, ×	×, ×
Gen2^{+} [38]	×, ×	✓, ×	×, ×	×, ×
HB^{+} [28]	✓, ×	✓, ×	✓, ×	×, ×
HB^{++} [2]	✓, ×	✓, ×	✓, ×	×, ×
YA-TRAP [39]	✓, ×	×, ×	✓, ×	×, ×
O-TRAP [5]	✓, ×	×, ×	✓, ×	×, ×
RIPP-FS [9]	✓, ×	×, ×	✓, ×	×, ×
Proposed Protocol	✓, ×	✓, ×	✓, ✓	✓, ✓

[a] (Inventorying, UADB-Inventorying);
[b] (Tracking, UADB-Tracking);
[c] (Tag-Impersonation, UADB-Tag-Impersonation);
[d] (Counterfeiting, UADB-Counterfeiting);

4 Implementation and Performance

In order to show the feasibility of the proposed protocols for both RFID tags and backend database, we implement and analyze both tag-side and database-side executions of our solutions.

4.1 Tag-side Simulation

The facts that development kit of EPC C1G2 devices are not commercially available and EPC tags, based on Application Specific Integrated Circuit (ASIC), are not user-programmable, drive us to implement tag-side primitives on a 4-bit microcontroller Platform ATAM893-D, a member of ATMEL MARC4 family. This microcontroller is famous for the high data throughput and extremely low current consumption (i.e., it consumes 0.1μA in deep-sleep mode, 0.6μA in sleep mode, 70μA in power-down mode and less than 1mA in active mode.). Besides, ATAM893-D has 4-bit single-chip microcontroller inherits a RISC core and contains EEPROM, RAM, parallel input/output ports, two 8-bit programmable multifunction counters/timer and an on-chip clock generation with integrated RC, 32kHz and 8MHz crystal oscillators. The development tool for programming on MARC4 microcontrollers is the MARC4 Starter Kit containing: an E-Lab ICP V24 Portable programmer, an MARC4 Programming board, a ready-to-run application board TEMIC T4xCx92; an integrated qForth compiler; an integrated simple MARC4 core simulator; an integrated Help Function with qForth dictionary and an ATMEL-wm ICP programmer software.

Figure 1: ATMEL MARC4 Starter Kit

There are three operations executed on the tag, namely l-bitwise XOR, CIP and PRNG. l-bitwise XOR is implemented by the iterative use of an arithmetic operation *DTOGGLE* in qForth, which XORs a byte at a given address (on top of the *Expression Stack*). Next, given the fixed u_i for each tag, the CIP is implemented using an lookup table of $2^{\frac{l}{d}}$ bits, which is similar as the implementation of S-box in the block ciphers. The on-tag pusedo random number generator is implemented by a novel lightweight cipher WG-7 [30], which has good verified randomness properties such as long period, 1-order resiliency, ideal two-level autocorrelation. The design of WG-7 as well as its cryptanalysis is presented in [30].

For each of the three operations, we test the consumed cycles by using different parameters (l, d). The result is given in Table 3, from which one could conclude that the CIP is almost as lightweight as XOR and the increase in l and d slight affects the cycle costs. Clocking the microcontroller at ZMHz, one can compute the time cost in microsecond (μs) through $\frac{cycle}{Z}$. Choosing $Z = 8$, our scheme satisfies the timing constrain of tags' response specified in [17] operations. Compared with [19, 34], the memory consumption, measured by lines of codes, is relatively small. Note that the PRNG g contributes around 80% to the memory usage. Future improvement in the design of lightweight PRNG decrease the memory

Table 3: Performance of tag-side computation on a 4-bit microcontroller.

(l, d)	l-bitwise XOR	CIP	PRNG $g(.)$		ROM	Stack Depth
	[cycle]	[cycle]	init.	output	[lines of code]	[EXP/RET]
$(32, 8)$	107	183	10866	3252	1146	7/4
$(96, 12)$	307	311	10866	9761	1387	7/4
$(96, 16)$	307	616	10866	9761	1197	7/4
$(128, 16)$	408	616	10866	12998	1389	7/4

usage in our protocol. In all, this implementation underlines that it is feasible to run our primitives and protocol on the same weak EPC tags.

4.2 Server-side Simulation

We next evaluate the performance of backend database-side. To this end, we first generate and store n (which ranges from $5,000$ to $5,000,000$) virtual tuples like $(epc_i, \{tid_i\}, y_i, u_i, E_{pk}(pin_i||c_i))$ in the memory of IBM Thinkpad T43 (P4 1.8, 2G memory, 80G harddisk). When the reader forwards tag's response $(p' \oplus e', \delta')$ to the database, one thread called *vcsearch* starts to search over the entire data set. Compared with the search over plaintext which is in fact consist of n l-bit comparison operations, *vcsearch* has n inner product operation, n XOR operation and n d-bit comparison. However, these bit-wise operations are very lightweight, which do not consume much of the CPU cycles. Besides, to let time cost be tested in the worst case, the tuple at the bottom of the set of virtual tuples is been searched.

The results are given in Figure 2, where the consumed time is proportional to the size of tuples as one would expect. When $l = 32$, the throughput is approximately 1GBps per thread. In the real database applications, volumes of optimization technologies are employed and hundreds of parallelized threads are involved for the search process. Hence, the single thread throughput exhibits that our primitive is feasible for the databases9.

4.3 False Distinguishing

Due to the compression nature of concatenated inner product, say $\{0,1\}^l \mapsto \{0,1\}^d$, the lost entropy $l - d$ results in false distinguishing in *vcsearch*. In other words, Tag tid_i will be mistakenly identified as tid_j where $i \neq j$ with probability $\frac{1}{2^d}$, More formally, there is a probability $\frac{1}{2^d}$ such that

$$((tid_j + s_j) + (tid_i + n_R + n_T)) \odot u_i = (n_R + n_T) \odot u_i + s_j \odot u_i$$

A straightforward solution is to increase d, which results the decrease of lost entropy. However,

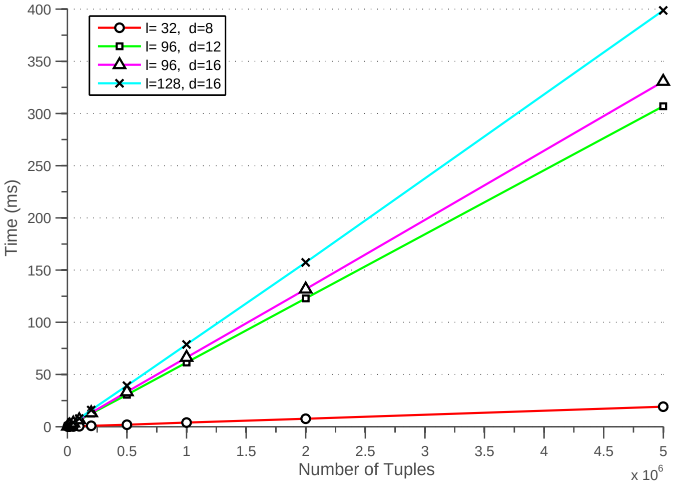

Figure 2: Database-side Performance

extra storage is demanded in the backend database. According to the experiential results, $d = 12, 16$ gives an balanced trade-off of these two factors.

5 Literature review

To the best of our knowledge, there is no literature regarding the cryptographic protection of backend database in RFID system targeting the anti-counterfeiting problem. In this section we first survey the schemes regarding *physical protection* for the anti-counterfeiting. Besides, typical works regarding identification and authentication protocols are reviewed as well, since some of them are also resistant to attacks like tag-impersonation or tag-clone.

Physical-Protection The most straightforward way of anti-counterfeiting is to build low-cost hardware chips such that even if an counterfeiter has physical access to the tag (and therefore the circuit schematics of this chip), it is close to impossible to produce an identical one. *Physically Unclonable Functions* (PUFs) [15] follows this idea, which prevents tags from been cloned. Several physical characteristics are exploited for the construction of PUFs, such as random delay property of wires and transistors, speckle patterns of optical medium for laser light, capacitance of a coating layer covering an IC, the acoustic refections etc.. A related-but-different primitive, *Physical Layer Identification* [11], refers to a verification system is built to test whether a device's fingerprint matches its claimed identity. Note that here the fingerprint are extracted from other physical characteristics such as modulation-shape and spectral features of the signals emitted by the tags. However, very recently, attacks are reported in [13] towards this approach.

The innovative primitive Memory Spots [4], which has small size (2 mm^2), very fast on-air data rate (10 Mbps), relatively large memory sizes (between 32 KB and 512 KB), provides another aspect of anti-counterfeiting. The data could be kept secretly in this tamper-proof-like low-cost chip, which makes it a perfect match of our scheme.

HB-family Jules and Weis extend Hopper and Blum's work [22] and propose HB$^+$ protocol [28] which has security reducible to *Learning Parity with Noise* problem. Alice and Bob share two secret vectors $\mathbf{x}$ and $\mathbf{y}$. In each round, a blinding-factor vector $\mathbf{a}$ is randomly generated by Alice and sent to Bob. Bob randomly selects $\mathbf{b}$ as response. Alice then generates a noise bit v which takes "1" with probability η, computes and transmits $z = (\mathbf{a} \cdot \mathbf{x}^T) \oplus (\mathbf{b} \cdot \mathbf{y}^T) \oplus v$. Bob independently computes $z' = (\mathbf{a} \cdot \mathbf{x}^T) \oplus (\mathbf{b} \cdot \mathbf{y}^T)$, and validates Alices response if $z = z'$. After n rounds, the authentication succeeds if and only if there is no more than $\lceil \eta n \rceil$ responses do not match challenges. However, Gilbert, Robshaw, and Sibert [20] have shown that there exists a man-in-the-middle active attack (GRS attack) for the second pass of every round of HB$^+$. Since the discovery of the GRS attack, a variety of protocols built upon HB$^+$, such as HB^{++} [2], HB* [12], etc., have been designed, intending to thwart the GRS attack. However, Gilbert, Robshaw, and Sibert [21] showed that these variants are vulnerable to GRS-like attacks as well and HB$^+$ remains the most attractive member of the family.

YA-TRAP-family YA-TRAP [39], designed to achieve untraceable privacy even when the tag is compromised, works as follows: a tag pre-shares a time interval $[T_t, T_{max}]$ (where T_t implies the last time it was interrogated) and a secret key K_i with the reader. Reader challenges the tag by sending current times T_r, if T_r is within interval $(T_t, T_{max}]$, tag responses a keyed hash value of T_r which can be verified by the reader, and updates T_t with T_r; otherwise, the tag outputs a random number that an adversary is unable to distinguish. However, YA-TRAP is vulnerable to both database-side Denial-of-Service (DoS) attack and tag-side DoS attack. For the former, an adversary can incapacitate a tag by sending a wildly inaccurate "current times". To solve this, O-TRAP [5], a hash-chain-like scheme, is introduced with a resynchronization mechanisms. However, the resynchronization causes $O(l \times n)$ search burdens to the database, where n is the number of tags and l is the steps required to ensure synchronization across the hash chain (the adversary could make l a huge number). Hence, it is not practical. Aiming at the same goal, RIPP-FS [9] claims to offer more security properties than its predecessors. However, a complicated tracking is later described in [33]. Besides, the highly dependency on the secure hash functions makes protocols in YA-TRAP-family less appealing for passive tags.

EPC-C1G2-family EPC-C1G2 tag is designed to strike a balance between cost and functionality, with little attention paid to security. To address this problem, particular protocols like C-D protocol [7], Q-Y-Y protocol [10] and Gen2$^+$ [38] are proposed exclusively for this standard.

C-D protocol and Q-Y-Y protocol are supposed to provide mutual authentication. However, two design flaws make them vulnerable to tag impersonation and tracking. First of all, the reader contributes the randomness to the protocol only and secondly, *Cyclic Redundancy Check* (CRC) is inappropriately treated as one way function. In fact, CRC inherits strong linearity aspects and is designed to support strong error detection particularly with respect to burst errors, not security. For any CRC and for any a, b, c and $d \in \mathbb{F}_2^l$, $CRC(a \parallel b) + CRC(c \parallel d) = CRC((a + c) \parallel (b + d))$ is always hold.

In Gen2$^+$ [38], instead of transmitting identity, PIN etc, the tag responses two 7-bit random numbers, say a, b, serving as index of a random l-word string k (named keypool) which is shared with the database, as well as creating a *check* c by XORing the two lsb of a-th word and b-th word. Database first removes tags's tuples which do not satisfy c and next computes ck' which is the majority vote of the CRC for the interval $[a, b]$ of the remaining tags. The tag then makes use of its local k to compute $ck := CRC(k[a : b])$ and compares ck with ck'. The tag does not response if the Hamming distance between ck and ck' is greater than some threshold; Otherwise, sends the locally stored EPC. This protocol is clearly subject to replay attacks because only the tag contributes to the randomness of protocol flows. Besides, it is possible for adversary to build up gradually sufficient information about the CRC of k and then recover the tag's keypool.

6 Conclusion

Inspired by how to achieve robustness in the wide-spread TID-based anti-counterfeiting mechanism, we first establish the UADB attack to show cryptographic protection of database-side data is extremely important to keep the confidentiality TID codes in the backend database. We next proposed a novel lightweight cryptographic primitive called verifiable cipher, which encrypts the data and, at the same time, keeps the encrypted data functional. With this new lightweight cryptographic primitive, we introduced a multifunctional RFID protocol and analysis its security properties in terms of UADB-enabled attacks. Then we give out performance details of our protocol implemented on both 4-bit microcontroller (to simulate the passive tag) and the IBM T43 laptop (to simulate the database server). The results justify that the time cost and storage cost is reasonable and our protocol is thus implementable to any low-cost passive tags.

References

[1] D. Boneh. Simplified OAEP for the RSA and Rabin functions. *Lecture Notes in Computer Science*, 2139, pp. 275-291, 2001.

[2] J. Bringer, H. Chabanne, E. Dottax, and S. D. Securite, HB^{++}: a lightweight authentication protocol secure against some attacks, *2nd International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing, SecPerU'06*, pp. 28-33, 2006.

[3] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. B.Robshaw and Y. Seurin, Hash functions and RFID tags: mind the gap, *Workshop on Cryptographic Hardware and Embedded Systems, CHES'08*, LNCS 5154 , pp. 283-299, 2008.

[4] H. Balinsky, E. McDonnell, L. Chen and K. Harrison, Anti-counterfeiting using memory spots, *Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks*, LNCS 5746, pp. 52-67, 2009.

[5] M. Burmester, T. Van Le and B. de Medeiros, Provably secure ubiquitous systems: universally composable RFID authentication protocols, *Conference on Security and Privacy for Emerging Areas in Communication Networks*, pp. 1-9, 2006.

[6] Y. Chang and M. Mitzenmacher, Privacy preserving keyword searches on remote encrypted data, *Lecture Notes in Computer Science* vol. 3531, pp. 442-455, 2005.

[7] C. Chen and Y. Deng, Conformation of EPC C1G2 standards RFID system with mutual authentication and privacy protection, *Engineering Applications of Artificial Intelligence*, vol. 22, no. 8, pp. 1284-1291, 2009.

[8] R. Curtmola, J. Garay, S. Kamara and R. Ostrovsky, Searchable symmetric encryption: improved definitions and efficient constructions, *Proceedings of the 13th ACM conference on Computer and Communications Security, CCS'06*, pp. 7988, 2006.

[9] M. Conti, R. D. Pietro, L. V. Mancini and A. Spognardi, RIPP-FS: An RFID identification, privacy preserving protocol with forward secrecy, *International Conference on Pervasive Computing and Communications*, pp. 229-234, 2007.

[10] Q. Cai, Y. Zhan and Y. Wang, A minimalist mutual authentication protocol for RFID system & BAN logic analysis, *Proceedings of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management, CCCM'08*, vol. 2, pp. 449-453, 2008.

[11] B. Danev, T. S. Heydt-Benjamin and S. Capkun, Physical-layer identification of RFID devices, *Proceedings of the 18th USENIX Security Symposium, USENIX'09*, 2009.

[12] D. N. Duc and K. Kim, Securing HB+ against GRS man-in-the-middle attack, *Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information Security*, pp. 23-26, 2007.

[13] B. Danev, H. Luecken, S. Capkun and K. El Defrawy, Attacks on Physical-layer Identification, *Proceedings of the ACM WiSec, WiSec'10*, 2010(to appear).

[14] T. Deursen and S. Radomirović, Algebraic attacks on RFID protocols, *Proceedings of the 3rd IFIP WG 11.2 International Workshop on Information Security Theory and Practice. Smart Devices, Pervasive Systems, and Ubiquitous Networks*, pp. 51-65, 2009.

[15] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola and V. Khandelwal, Design and implementation of PUF-based "unclonable" RFID ICs for anti-counterfeiting and security applications, *IEEE International Conference on RFID, RFID'08*, pp. 58-64, 2008.

[16] D. Dolev and M. Naor, Non-Malleable Cryptography, *Proceedings of the 23th annual ACM Symposium on Theory Of Computing, STOC'91*, pp. 542-552, 1991.

[17] EPC Global, Class 1 Generation 2 UHF air interface protocol standard v1.2, *http://www.epcglobalinc.org*, 2008.

[18] C. Fontaine and F. Galand, A survey of homomorphic encryption for nonspecialists, *EURASIP Journal on Information Security*, pp. 1-15, vol.2007, 2007.

[19] X. Fan, H. Hu, G. Gong, E. M. Smith and D. Engels, Lightweight implementation of Humming bird cryptographic algorithm on 4-bit microcontrollers, *Proceedings of the 1st International Workshop on RFID Security and Cryptography 2009*, pp. 838-844,2009.

[20] H. Gilbert, M. Robshaw and H. Sibert, An active attack against HB^+ – a provably secure lightweight authentication protocol, *IET Electronic Letters*, vol. 41, no. 21, pp. 1169-1170, 2005.

[21] H. Gilbert, M. J. Robshaw and Y. Seurin, Good variants of HB^+ are hard to find, *Financial Cryptography and Data Security, FC'08*, LNCS 5143, pp. 156-170, 2008.

[22] N. J. Hopper and M. Blum, Secure Human Identification Protocols. *Advances in Cryptology, ASIACRYPT'01*, LNCS 2248, pp. 5266, 2001.

[23] M. Hell, T. Johansson and W. Meier, Grain: a stream cipher for constrained environments, *International Journal of Wireless and Mobile Computing*, vol. 2, no. 1, pp. 86-93, 2007.

[24] A. Juels, RFID security and privacy: a research survey, *IEEE Journal on Selected Areas in Communications*, vol. 24, No. 2, pp. 381-394, 2006.

[25] H. C. JulienBringer and T. Icart, On physical obfuscation of cryptographic algorithms, *Progress in Cryptology, Indocrypt 2009*, pp. 88-103, 2009.

[26] A. Jøsang, R. Ismail and C. Boyd, A survey of trust and reputation systems for online service provision, *Decision Support Systems*, vol. 43, no. 2, pp. 618-644, 2007.

[27] A. Juels, Strengthening EPC tags against cloning, *Proceedings of the 4th ACM workshop on Wireless Security, WiSec'05*, pp. 67-76, 2005.

[28] A. Juels and S. A. Weis, Authenticating Pervasive Devices With Human Protocols, *Advances in Cryptology*, LNCS 3621, pp. 293-308, 2005.

[29] H. Kevin, The theory and applications of homomorphic cryptography, *Master Thesis, University of Waterloo*, 2008.

[30] Y. Y. Luo, Q. Chai, G. Gong and X. J. Lai, WG-7, a lightweight stream cipher with good cryptographic properties, *Preprint*, February, 2010.

[31] D. Molnar, A. Soppera, and D. Wagner, A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags, *Selected Areas in Crypto.*, LNCS 3897, pp. 276-290, 2006.

[32] P. Peris-Lopez, Lightweight cryptography in radio frequency identification (RFID) systems, *PhD Thesis, Universidad Carlos III De Madrid*, 2008.

[33] K. Ouafi and R. C. W. Phan, Privacy of recent RFID authentication protocols, *Information Security Practice and Experience*, LNCS 4991, pp. 263-277, 2008.

[34] M. Vogt, A. Poschmann, and C. Paar, Cryptography is feasible on 4-bit microcontrollers - a proof of concept, *2009 IEEE International Conference on RFID*, pp. 241-248, 2009.

[35] M. Raykova, B. Vo, S. Bellovin and T. Malkin, Secure anonymous database search, *Proceedings of the 2009 ACM workshop on Cloud computing security, CCS'09*, pp. 115-126, 2009.

[36] S. J. Stolfo, S. M. Bellovin, S. Hershkop, A. Keromytis, S. Sinclair and S. W. Smith, Insider attack and cyber security, *Advances in Information Security* vol. 39, 2008.

[37] J. A. K. SAP and M. Lehtonen, Anti-counterfeiting prototype report, *Technical Report of BRIDGE Project*, 2009.

[38] H. M. Sun and W. C. Ting, A Gen2-based RFID authentication protocol for security and privacy, *IEEE Transactions on Mobile Computing*, vol. 8, no. 8, pp. 1052-1062, 2009.

[39] G. Tsudik, YA-TRAP: yet another trivial RFID authentication protocol, *International Conference on Pervasive Computing and Communications, Percom'06*, pp. 640-643, 2006.

[40] J. Wu and D. R. Stinson, How to ensure forward and backward untraceability of RFID identification schemes by using a robust PRBG, *Cryptology ePrint Archive*, Report 2008/201, 2008.