Diffusion-Confusion based Light-weight Security for Item-RFID Tag-Reader Communication

DIVYAN M. KONIDALA[1], KWANGJO KIM[2], MADE HARTA DWIJAKSARA[2], DAEYOUNG KIM[2]

[1] Dept. of Information and Communications Engineering
[2] Dept. of Computer Science
Korea Advanced Institute of Science and Technology (KAIST)
REPUBLIC OF KOREA
{divyan, kkj, made.harta, kimd}@kaist.ac.kr

Abstract

In this paper we propose a challenge-response protocol called: *DCSTaR*, which takes a novel approach to solve security issues that are specific to low-cost item-RFID tags. *DCSTaR* protocol is built upon light-weight primitives such as 16 bit: Random Number Generator, Exclusive-OR, and Cyclic Redundancy Check and utilizing these primitives it also provides a simple Diffusion-Confusion cipher to encrypt the challenges that are sent from the tag to the RFID reader, thus obscuring sensitive data from eavesdropping malicious readers. *DCSTaR* protocol also provides an efficient way for consumers to verify whether tagged items are genuine or fake and to protect consumers' privacy while carrying tagged items.

Keywords: RFID, Tag-Reader communication security, Light-weight cryptography, Customer privacy, Diffusion-Confusion cipher, EPCglobal Class-1 Gen-2

1 Introduction

1.1 Item RFID: Technology and Standards

Radio Frequency IDentification (RFID) technology [17] offers businesses an automated supply chain management system [36]. With RFID technology, manufacturers attach Passive-RFID item-tags to their products (items). Passive item-tags are low-cost electronic labels that are resource constrained (up to 512 bytes of memory, 3K gates). These tags contain tiny computer chips with very small antennas and are powered-up by a Radio-Frequency (RF) signal from an RFID reader. The tiny chip contains a unique Electronic Product Code (EPC) that identifies the item to which it is attached to, and the antenna automatically transmits this EPC number (without requiring line-of-sight scanning) to readers within the RF range (up to 10m).

Further information associated with the item/EPC number (*e.g.*, item description, manufacturing date, packaging, shipments, item arrival and departure details, *etc.*) is captured and stored on a network of servers and databases, called EPC-Information Services (EPC-IS) [1]. The unique EPC number is like a universal resource locator (URL) directing the reader to the right EPC-IS on the EPC Network from where the reader can download and upload data about the item it scanned. Therefore, RFID and EPC-IS assist geographically distributed supply-chain stakeholders (*e.g.*, manufacturers, distributors, retailers, *etc.*) with instantaneous item identification, and "real-time" updating, querying, accessing and sharing of item information such as, shipping and receiving, track and trace, theft detection, precise item recall *etc.* As a result, very soon we can expect to see RFID tagged consumer items at many retailers.

The standards like the ISO 18000: Part 1-4, 6 and 7 describe the use of RFID for item management. We also have the EPCglobal Inc. [1], leading the development of industry-driven standards for the EPC to support RFID in supply chain management. The ISO 18000 Part 6C is in fact the EPCglobal's standard: "Class-1 Generation-2 (C1G2) UHF (Ultra High Frequency) RFID Protocol for Communications at 860MHz - 960MHz" [10]. This standard is for low-cost, passive-backscatter, 'interrogator talks first', RFID system operating in the 860 MHz - 960 MHz frequency range. It specifies the Physical interactions (the signaling layer of the communication link) between readers and tags, and reader-tag operating procedures and commands.

In the proceeding sections, we "exemplify" the C1G2 protocol only to understandably describe the motivation, design and working aspects of our proposed light-weight security protocol and certainly not to imply that our protocol is only suitable for C1G2 tags, instead it can be applied to other types of item-passive tags.

1.2 Security Aspects of C1G2 (ISO 18000:6C) Protocol

As per the EPCglobal C1G2 UHF RFID Protocol standard [10], a tag's chip has four memory banks: *Reserved*, *EPC*, *TID*, and *User*. The *EPC* memory bank is used to store the EPC number, *TID* memory bank for tag's unique manufacturer identity number, and *User* memory bank for additional user data. The manufacturer of the items stores a 32 bit *Access Password* ($A[31:0]$) and a 32 bit *Kill Password* ($K[31:0]$) into the tags' *Reserved* memory bank. The reserved memory bank is permanently locked by the manufacturer; therefore the *Access* and *Kill Passwords* can neither be read nor modified by any reader.

The tag has the capability to verify these two passwords, therefore if a reader sends the right *Access Password*, the tag enters the *Secured State*, where the reader is allowed to carry out mandatory commands such as *Read*, *Write*, and *Lock* on the tag. On the other hand if a reader sends the right *Kill Password*, the tag enters the *Killed State*, where it is permanently disabled. The C1G2 standard does not provide details on how to securely communicate the *Access* and *Kill Passwords* to the readers along the supply chain.

According to the C1G2 standard, tags can generate 16 bit random or pseudo-random numbers ($RN16$) and execute XOR ($\oplus$), and cyclic-redundancy check (CRC) operations. Initially the reader identifies the tag via a *Query* command to obtain its EPC number. Later, the reader and tag implement an *Access Command* procedure; which causes the tag to transition from the *Open* to the *Secured State*, where the reader and tag can communicate indefinitely.

The *Access Command* procedure is fairly easy to understand by studying the multi-step procedure shown in Figure 1. Prior to issuing the *Access Command*, the reader first requests a random number from the tag via the Req_RN command. Later, the tag sends two 16 bit random challenges $RN16\_1$ and $RN16\_2$. The reader responds with ($A[31:16] \oplus RN16\_1$) and ($A[15:0] \oplus RN16\_2$). In here the $RN16$ is used has an XOR-pad to obscure $A[31:0]$, this is known as Cover-Coding Access Password. The tag verifies these responses in order to authenticate the reader. To ensure the validity and integrity of received data both tags and readers shall compute and send a 16 bit Cyclic-Redundancy Check (CRC) value along with their data.

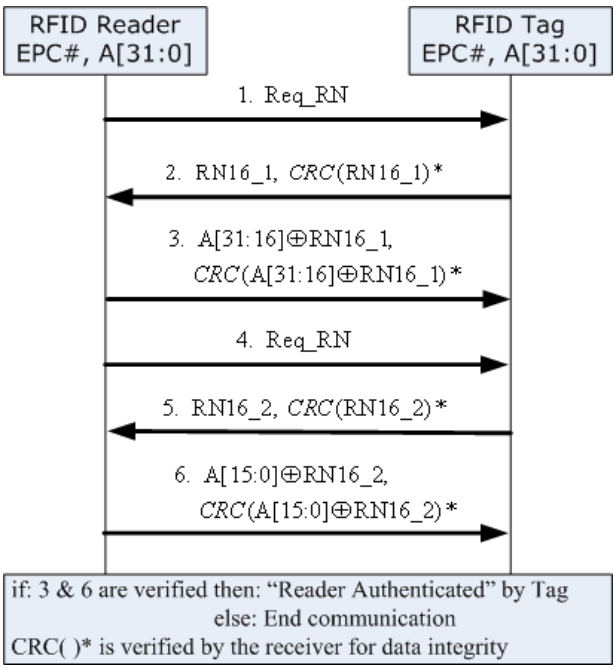

Figure 1: EPCglobal C1G2 (ISO 18000:6C) Protocol: Access Command Procedure

1.3 Security Threats

In the *Access Command* procedure (Figure 1) the tag sends its un-obscured challenges $RN16\_1$ and $RN16\_2$ (steps 2 and 5) in the open. Therefore by eavesdropping on any one of the communication sessions between the tag and the reader any adversary can capture $RN16\_1$ and $RN16\_2$, and reverse the $\oplus$ operation in the reader's responses - steps 3 and 6 to expose $A[31:0]$. Because of this flaw, even though both the ISO and EPCglobal standards provide (weak) reader authentication and tag memory locking features, they suffer from the following security threats, for which we propose appropriate security requirements.

1.3.1 Man-in-the-Middle Attack

To accommodate quick and speedy scanning of goods in large bulks, EPCglobal C1G2 UHF RFID tags exhibit outstanding far-field performance. Readers can query and communicate with these tags over a range of 10 meters. Therefore, we can anticipate Man-in-the-Middle attacks from powerful malicious readers. This attack can be mounted to eavesdrop on the communication channel between the tag and the reader, capture the tag's EPC number, impersonate either as a tag or a reader, manipulate

their communicating data, and disclose/expose the *Access Password*.

1.3.2 Cloned Fake Tags

The intrinsic functionality of a tag is to respond to any querying reader with its EPC number. Therefore a malicious reader can easily scan and copy the data (EPC number and exposed *Access Password*) on a genuine tag and embed the same data onto a fake tag. This fake tag can be attached to a counterfeit item. Even though a particular tag gives out a genuine EPC number, it must still be authenticated by the reader.

1.3.3 Malicious Readers

An exposed *Access Password* can be utilized by a malicious reader to corrupt the genuine tag. Therefore a tag must also be able to authenticate its reader. Also, only authorized readers must be allowed to access the EPC-IS.

1.3.4 Insider Attack

All the hundreds of readers in the supply chain cannot be trusted with *Access* and *Kill Passwords*. Any disgruntled employee can compromise authorized readers in a system and can easily obtain these *Passwords*. Especially the *Access Password* for a tag remains the same for the rest of the item's life cycle. Therefore, an exposed *Access Password* at any of the stockholders end would easily lead to fabrication of cloned fake tags with the same *Access Password*.

1.3.5 Consumer Privacy Violation

A consumer carrying a tagged item can be identified, tracked and traced based solely on the tag's unique EPC number.

1.4 Proposed Countermeasures

- Tag ← Reader → EPC-IS mutual authentication, alleviates the threats from tag/reader impersonation, malicious readers, and cloned fake tags.
- Communicating-data confidentiality and integrity.
- Secure key-distribution and key-protection.
- Readers must not be provided with any of the keys, but only be permitted to relay obscured data between the tag and the EPC-IS/back-end server.
- Anonymity for the tags that are in the possession of a consumer.

2 Related Work

We studied many interesting protocols that addressed the above threats. Some of the previously proposed solutions are based on hash functions [1], [3],[24],[11],[24] and optimized implementations of block (AES, DES) [11], and stream ciphers, but passive low cost item-tags are not capable of executing such computationally intensive functions due to their constrained resources. Therefore in here we discuss only light-weight protocols [24] that utilize light-weight primitives like the Random Number Generator (RNG), Cyclic Redundancy Check (CRC), modular addition and bit-wise operators such as XOR, AND, OR, rotate, *etc*.

Juels *et al.* [14] first proposed HB+ protocol, which is based on `inner dot product' and satisfying NPhard - `Learning Parity with Noise' problem. HB+ and its later improvements have all been proved insecure against the man-in-the-middle attack [12], [23], exposing the tag's secret keys and these protocols consider only tag (not reader) authentication. They also require a minimum of: 500 bit keys, many 250 bit challenge strings, and a noise parameter of 0.25 [37], all of which may not be practical for item-tags.

Karthikeyan *et al.* [15] proposed a protocol that utilizes matrix-multiplication and XOR, but Chien *et al.* [4] showed it suffers from de-synchronization of session keys and replay (impersonation) attacks and proposed an improvement that uses RNG, CRC, and XOR. However, Peris-Lopez *et al.* [25] proved that [4] is still not secure from the very same attacks and later proposed three novel protocols that use XOR, AND, OR, and addition mod 2^m: LMAP [29], M2AP [31], and EMAP [30], but Li *et al.* [21], [22] proved that these protocols again suffer from de-synchronization and full-disclosure of tag's secret information. Konidala *et al.* [18] used only RNG, CRC, and XOR in their protocol, but Peris-Lopez *et al.* [26] showed key-disclosure attack. Then again, Chien *et al.* [6] pointed out the weakness of [21] and like-wise Arco *et al.* [7] proved that SASI protocol [4] (which additionally used rotate operation) is also prone to the above mentioned weaknesses.

Lastly Peris-Lopez et *al.* [24] and Burmester *et al.* [2] have also shown that the most recent light-weight protocols are also susceptible to: key disclosure, man-in-the-middle, de-synchronization, replay, and impersonation attacks.

2.1 Drawbacks of providing tag anonymity at supply chain

To achieve tag anonymity, previous protocols prevent the tag from emitting its EPC; instead use "per-transaction-updatable" tag Pseudo-IDs (PIDs).

The innovative measures proposed by Burmester *et al.* [2] and Peris-Lopez et *al.* [27] to: minimize exhaustive computation and DB search for a particular PID, restore PID synchronization between the tag and EPC-IS, resolve PID collisions in the DB, and session unlinkability; can still be a bit overkill/impractical, causing overhead, delay, and uncertainty at a large-scaled and fast-paced supply-chain processing. The speed, accuracy, and atomicity achieved with EPC is lost and as per the EPCglobal, it is the EPC that is used as an URL along with Object Naming Server to locate the appropriate EPC-IS. Therefore, using PIDs at the supply-chain level defeats the very purpose of RFID.

Our work doesn't undermine the contributions of [2] and [27], instead we consider that though the EPC is exposed at supply-chain level, we can alleviate the threats that demand the need for tag anonymity at the supply-chain level by simply allowing only authorized (stakeholders) readers to access EPC-IS. This prevents malicious readers from obtaining critical detailed information about items from the EPC-IS.

3 Contribution

We call our proposed protocol ***DCSTaR***, which takes a different approach, focusing and encouraging future research on the (above mentioned) simplified yet specific threats pertaining to item-tags in the supply chain and those in the possession of the consumer. Our proposed protocol has the following salient features:

- *DCSTaR* is a challenge-response protocol.
- It is a light-weight protocol satisfying all the above-mentioned countermeasures and consists of a simple cipher to encrypt the challenges from the tag.
- It utilizes only the primitives: RNG, CRC, and XOR and provides *Diffusion* and *Confusion* - the two fundamental properties for a secure cipher [34], taking in 32 bits and producing 64 ciphered bits. Diffusion: the output bits should depend on the input bits in a very complex way. Confusion: making the relationship between the key and the output bits as complex and involved as possible
- The tag encrypts the challenges that are sent to the interrogator, but doesn't have to do any decryption to verify the response from the interrogator.
- Our *Diffusion* and *Confusion* cipher is simple to implement and execute in a tag when compared to traditional block ciphers.
- *DCSTaR* may not provide a full-proof security but just enough security to justify the cost of affordable item-tags.
- Unlike the other protocols, *DCSTaR* is also an efficient way for consumers to verify if an item is genuine or fake. It provides anonymity where it is needed the most; not at the supply chain level but for the tags in the consumer's possession.

4 Proposed *DCSTaR* Protocol

4.1 Setup

As per the EPCglobal's C1G2 UHF RFID Protocol standard [10], the tag's *Reserved* memory bank is composed of 16 bit memory slots, where *Kill Password* $K[31:0]$ and *Access Password* $A[31:0]$ are stored at the addresses $00_h \sim 1F_h$ and $20_h \sim 3F_h$ respectively. As shown in Figure 2, we propose an expansion to the tag's reserved memory bank to include a 32 bit *Extra Key* $X[31:0]$, *Sixteen 8 bit unique Keys* $G_0[127:120] \sim G_{15}[7:0]$ and *Sixteen 4 bit unique Keys* $U_0[63:60] \sim U_{15}[3:0]$.

Addr.	**Reserved Memory Bank**	
$3F0_h \sim 3F3_h$	$U_{15}[3:0]$	16 x 4 bit unique keys =64 bits (1:1 mapping b/w U Addr. ↔ $U_{0\sim15}$)
⋮	⋮	
$300_h \sim 303_h$	$U_0[63:60]$	
$2F0_h \sim 2F7_h$	$G_{15}[7:0]$	16 x 8 bit unique keys =128 bits (1:1 mapping b/w G Addr. ↔ $G_{0\sim15}$)
⋮	⋮	
$200_h \sim 207_h$	$G_0[127:120]$	
$50_h \sim 5F_h$	Xtra Key: *X[15:0]*	
$40_h \sim 4F_h$	Xtra Key: *X[31:16]*	
$30_h \sim 3F_h$	Access Password: *A[15:0]*	
$20_h \sim 2F_h$	Access Password: *A[31:16]*	
$10_h \sim 1F_h$	Kill Password: *K[15:0]*	
$00_h \sim 0F_h$	Kill Password: *K[31:16]*	

Figure 2: Proposed Expansion of the Tag's Reserved Memory Bank

4.2 Assumptions

The keys: K, A, and X are unique for each tag. The keys: $G_{0\sim15}$ and $U_{0\sim15}$ must all be unique among each other, i.e., no two memory addresses should have the same key, satisfying 1:1 mapping between the address and the key. The criteria to choose s-boxes [19] [16] for block-ciphers can also be applied to choose the unique keys: $G_{0\sim15}$ and $U_{0\sim15}$ that are secure against differential and linear cryptanalyses, therefore such (many) sets of good unique keys could be ``wisely'' re-used among different tags. All of the above keys are kept secret between the tag and EPC-IS.

Before initiating *DCSTaR* protocol, we assume that the reader issues *Query* command to obtain the

EPC number from the tag and pass it on to the trusted and secure EPC-IS. We assume that the communication channel between the resource rich entities RFID Reader and EPC-IS, to be secure (SSL-TLS and X.509 Authentication Framework).

4.3 Description

Our proposed *DCSTaR* protocol could be easily understood by studying the Figure 3.

RFID Tag	**RFID Reader**	**EPC-IS / Backend**
EPC#		EPC# (Tag Identified)
(K, A, X, $G_{0\sim15}$, $U_{0\sim15}$)		(K, A, X, $G_{0\sim15}$, $U_{0\sim15}$)
RNG(), CRC(), XOR ⊕, f()		*RNG(), CRC(), XOR ⊕, f^{-1}()*
	1. Generate 16 bit Random: R	
	2. R, *CRC*(R)* →	
3.1. Generate 16 bit Randoms: S_1, S_2, T_1 3.2. *CRC*(K, A, X, R, T_1) = T_2 3.3. $f_{S_1,S_2}(T_1,T_2) = C$		
	4. S_1, S_2, C, *CRC*(S_1, S_2, C)* →	
		5.1. $f^{-1}_{S_1,S_2}(C) = T_1, T_2$
		5.2. if: *CRC*(K, A, X, R, T_1) == T_2 then: Tag Authenticated *CRC*(X, A, S_2, T_1) ⊕ T_2 = J_1 *CRC*(T_2, S_1, K, X) ⊕ T_1 = J_2 else: Fake Tag, end communication
	6. J_1, J_2, *CRC*(J_1, J_2)* ←	
7. if: *CRC*(X, A, S_2, T_1) ⊕ T_2 == J_1 *CRC*(T_2, S_1, K, X) ⊕ T_1 == J_2 then: Reader / EPC-IS / Backend Authenticated else: Malicious, end communication		

f( ): Diffusion and Confusion Procedure (Fig. 4)
f^{-1}( ): Inverse of *f( )*
S_1, S_2: Seeds for generating Round Keys ($Y_{1\sim10}$) for *f( )*
*CRC( )** is verified for data integrity

Figure 3: Proposed DCSTaR Protocol

4.3.1 Diffusion-Confusion Cipher: $f(\,)$

The Figure 4 describes Diffusion & Confusion procedure: $f_{S_1,S_2}(T_1,T_2) = C$, which encrypts 32 bit T_1 and T_2 into a 64 bit cipher C .

- S_1 and S_2 are the seeds for the 16 bit *Round Keys* ($Y_{0\sim10}$).

$$Y_0 = CRC(K, A, X, S_1, S_2)$$
$$Y_n = CRC(Y_{n-1}, K, A, X, S_1, S_2) \; where \; n = 1 \cdots 10$$

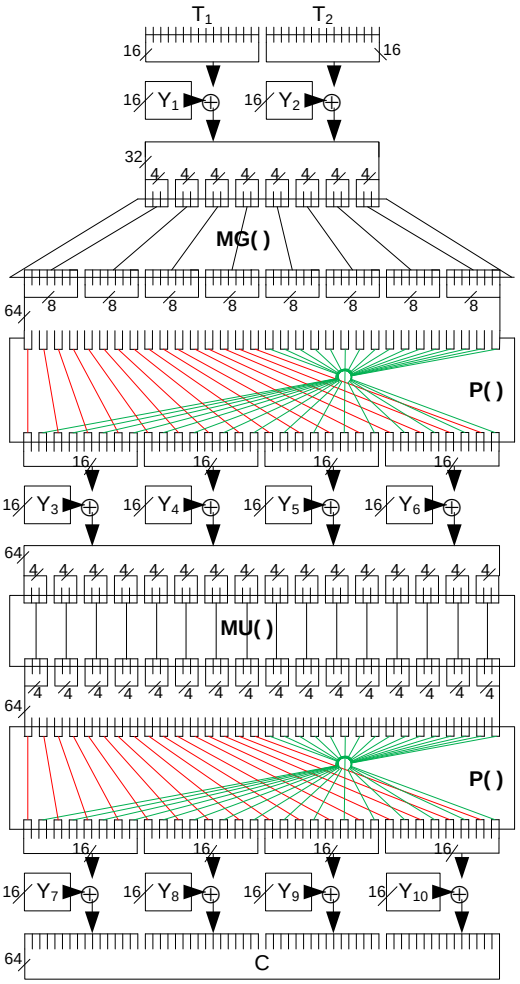

Figure 4: Diffusion & Confusion procedure: $f(\,)$

- The $MG(\,)$ is a 1:1 mapping function to $G_{0\sim15}$, where 4 input bits are replaced by an 8 bit unique Key. 4bit input → G Addr. → 8bit Key $G_{0\sim15}$
  E.g.,
  $MG(1111) = MG(F_h) = (2F0_h \sim 2F7_h) = G_{15}[7:0]$

- $MU(\,)$ is a 1:1 mapping function to $U_{0\sim15}$, where 4 input bits are replaced by a *4 bit unique Key*. 4bit input → G Addr. → 8bit Key $U_{0\sim15}$
  E.*g.*,
  $MU(0101) = MU(5_h) = (350_h \sim 353_h) = U_5[43:40]$

- The Bit Transpose $P(\,)$ is a concatenation of a sequence of two 'most significant bits' and two 'least significant bits' of its input. *E.g.*, if 64 bit $B = b_0 \sim b_{63}$, then
  $P(B) = b_0 b_1 b_{62} b_{63} \,||\, b_2 b_3 b_{60} b_{61} \,||\, \cdots \,||\, b_{30} b_{31} b_{32} b_{33}$

4.3.2 Inverse Diffusion-Confusion Cipher: $f^{-1}()$

To compute $f^{-1}_{S_1,S_2}(C) = T_1, T_2$, we first generate all the 16 bit *Round Keys* ($Y_{0\sim10}$) and proceed with the $f()$ procedure bottom-up, until T_1 and T_2 are recovered. In here the:

- $MG^{-1}(G_{15}[7:0]) = (2F0_h \sim 2F7_h) = F_h = 1111$
- $MU^{-1}(U_5[43:40]) = (350_h \sim 353_h) = 5_h = 0101$

5 Analysis of *DCSTaR* Protocol

5.1 Tag ← Reader → EPC-IS Mutual Authentication

Readers and EPC-IS authenticate and secure their communications via TLS/SSL protocol. An adversary can't randomly pick T_1 and T_2, as T_2 and C can only be computed by a tag possessing R, T_1, and the keys: $K, A, X, G_{0\sim15}, U_{0\sim15}$. Only the EPC-IS possessing these keys can recover T_1, T_2 and first verify $CRC(A.K, X, R, T_1) == T_2$ and then compute J_1 and J_2. The R, T_1, S_1 and S_2 are all unique for every transaction and are all linked together throughout the protocol to thwart any kind of reply attacks.

5.2 Data Integrity

Though $CRC()*$ provides data integrity check, any modification to steps 2, 4 & 6 (Figure 3) would fail the authentication process.

Additional feature: Let us assume that the EPC-IS wants to write some encrypted-user-data into the tag. If Z represents such an encrypted-user-data, then at Step 5.2 (Figure 3) EPC-IS computes $CRC(X, A, S_2, Z, T_1) \oplus T_2 = J_1$. An adversary can intercept and modify Z to Z', and send $\{J_1, J_2, Z', CRC(J_1, J_2, Z')\}$ to the tag at Step 6 (Figure 3). But the tag can detect this malicious modification of Z because: $CRC(X, A, S_2, Z', T_1) \oplus T_2 \neq J_1$.

5.3 Key Protection and Secure Key Distribution

It is evident that the steps 2, 4, & 6 (Figure 3) do not expose any of the keys: $K, A, X, G_{0\sim15}, U_{0\sim15}$.

DCSTaR protocol can be executed while the reader is connected online with manufacturer's EPC-IS. Alternatively, the manufacturer can remotely access, monitor, and manage a server at every stakeholder's supply-chain processing facility and update this server with relevant tags' keys.

5.4 Reader Relaying Only Obscured Data

It is evident from the steps 4 & 6 (Figure 3) that readers are relaying only obscured data between the tag and EPC-IS. Sensitive data like the keys $K, A, X, G_{0\sim15}, U_{0\sim15}$ and the challenges T_1 and T_2 are not revealed to the readers.

5.5 Tag Verification and Tag Anonymity for Consumers

A consumer can use his/her RFID reader-enabled portable device (*e.g.*, mobile phone) to *Query* and send R to the tag (as in Step 1-Figure 3). This RFID reader-enabled portable device obtains the EPC, S_1, S_2 and C from the tag, and send this data along with the R to the EPC-IS via 3G/4G network or Wi-Fi connection. EPC-IS would then verify C and replies to the device whether the item is genuine or fake. In here neither the tag's keys nor tag's sensitive data are exposed to the customer.

After purchasing an item the consumer would obtain the tag keys: $K, A, X, G_{0\sim15}, U_{0\sim15}$ from the store and store them into his/her device. Using these keys the consumer can execute the *DCSTaR* protocol and read-lock the *EPC* memory bank using the *Lock* command. As a result the tag no longer emits its *EPC* number, thus protecting the privacy of the consumer from eavesdropping malicious readers.

Since the tag no longer emits its *EPC* number, the consumer executes *DCSTaR* protocol by just sending R to the tag. The tag responds with its 64 bit C, which now becomes the tag's Pseudo-ID. The consumer uses this PID to do a brute force search of all the tags in his/her possession that give out the same C and thus arrives at the correct *EPC* number. A consumer would not have that many items/tags; therefore we can assume that there would be no PID collisions or computationally intensive database searches.

5.6 Performance Aspects

- *DCSTaR* achieves Tag ← Reader → EPC-IS mutual authentication in just three communication steps 2, 4, & 6 (Figure 3), whereas

EPCglobal's C1G2 UHF RFID Protocol standard [10] achieves only "one-way" reader authentication in six communication steps (Figure 1).

- *DCSTaR* strictly utilizes only the RNG, XOR, and CRC light-weight primitives/operations.
- The mapping functions $MG(\,)$ and $MU(\,)$ are implemented in a way that the input bits to these functions are used as a memory address to replace them with the KEY stored in that address. This simple approach requires no additional hardware implementation like the substitution and inverse tables.
- The tag needs to execute only $f(\,)$ procedure but not $f^{-1}(\,)$ procedure.
- *DCSTaR* protocol does require an additional memory space of 432 bits to accommodate the keys and to execute the diffusion-confusion cipher $f(\,)$ procedure. However we have to assume that low-cost passive item-tags can have a memory capacity of several bytes *e.g.*, 512 bytes, therefore *DCSTaR's* additional memory requirement can be easily incorporated.

5.7 Data Confidentiality

In Figure 3: Step 2, we can notice that even though R is exposed there is no threat to the protocol, as it's just one among four other secrets K, A, X and T_1 needed to compute T_2. The 64 bit C (Figure 3) obscures T_1 and T_2. Similarly, the 16 bit J_1 and J_2 are neither guessable nor exposing any sensitive data.

Additional feature: A tag may store few bytes of stakeholder's (user) data. We suggest that the reader Writes already encrypted user data it received from EPC-IS. At a later stage, the reader can retrieve the stored encrypted user data from the tag and relay it to the EPC-IS to be decrypted. Thus the data is secured in the tag and also while writing/reading to/from the tag.

To justify our use of only two round *Diffusion-Confusion* (Figure 4) and the strength of $f_{S_1,S_2}(T_1,T_2) = C$ procedure, we utilized TestU01 - a software library of `utilities for empirical statistical testing of RNGs' implemented in the C language [20]. TestU01 is comprehensive, frequently updated, and encompasses most of the other test-suites. We subjected several 150 megabytes of C values obtained under multiple trails and different keys to the following batteries of test: SmallCrush, PseudoDIEHARD, Alphabit, BlockAlphabit, Rabbit, and FIPS-140-2 (NIST std.: security requirements for cryptographic modules).

The batteries Rabbit, Alphabit and BlockAlphabit are for binary sequences from a cryptographic pseudorandom generator. Most of these batteries return *p-values* for all its tests, and those that are within the [0.001~0.9990] range are passed. To speed-up these tests, we utilized cluster computing and implemented *DCSTaR* as a parallel C program. *DCSTaR* passed all these batteries of tests.

5.6.1 FIPS_140_2 Test Suite

This NIST package contains 15 tests, oriented primarily toward the testing and certification of RNGs used in cryptographic applications [33]. The results of this test are presented in the Table 1.

Summary results of FIPS-140-2
Number of bits: 20000

Test	s-value	p-value	FIPS Decision
Monobit	9979	0.61	Pass
Poker	18.87	0.22	Pass
0 Runs, length 1	2508		Pass
0 Runs, length 2	1233		Pass
0 Runs, length 3	634		Pass
0 Runs, length 4	306		Pass
0 Runs, length 5	168		Pass
0 Runs, length 6+	152		Pass
1 Runs, length 1	2450		Pass
1 Runs, length 2	1300		Pass
1 Runs, length 3	653		Pass
1 Runs, length 4	307		Pass
1 Runs, length 5	152		Pass
1 Runs, length 6+	139		Pass
Longest run of 0	13	0.50	Pass
Longest run of 1	13	0.50	Pass
All values are within the required intervals of FIPS-140-2			

Table 1: NIST (FIPS_140_2) package: testing & certification of RNGs for cryptographic applications

6 Conclusion

We are confident that *DCSTaR* protocol would encourage further research especially on low-cost item-tags implementing simple ciphers and meeting the minimum security requirements as suggested in this paper. Our future work would include practical design and implementation of *DCSTaR* protocol and evaluate its throughput, the die size, clock cycles, and power consumption.

Acknowledgment

This research was supported by the ICT Standardization program of the Republic of Korea's MKE (The Ministry of Knowledge Economy). We thank PEDRO PERIS-LOPEZ of Delft University of Technology (TU-Delft), NETHERLANDS, and TIEYAN LI, and JIANYING ZHOU of Institute for Infocomm Research (I2R), A*STAR, SINGAPORE for their valuable comments and suggestions.

References

[1] Avoine, G., and Oechslin, P., *A Scalable and Provably Secure Hash-Based RFID Protocol*, Proceedings of Workshop on Pervasive Computing and Communications Security, PerSec'05, pp. 110-114, IEEE Press, 2005.

[2] Burmester, M., and Munilla, J., *A Flyweight RFID Authentication Protocol*, RFIDSec`09, 2009. http://www.cosic.esat.kuleuven.be/rfidsec09/Papers/paper-burmester-munilla.pdf

[3] Burmester, M., De Medeiros, B., and Motta, R., *Anonymous RFID authentication supporting constant-cost key-lookup against active adversaries*, International Journal of Applied Cryptography, vol. 1, no. 2, pp. 79-90, 2008.

[4] Chien, H.Y., *SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity*, IEEE Transactions on Dependable & Secure Computing, pp.337-340, 2007.

[5] Chien, H.Y., and Chen, C.H., *Mutual authentication protocol for RFID conforming to EPC class-1 generation-2 standards*, Computer Standards & Interfaces, vol. 29, pp.254-259, 2007.

[6] Chien, H.Y., and Huang, C.W., *Security of Ultra-Lightweight RFID Authentication Protocols and Its Improvements*, ACM Operating System Rev., vol. 41, no. 2, pp. 83-86, July 2007.

[7] D'Arco, P., and De Santis, A., *Weaknesses in a Recent Ultra-Lightweight RFID Authentication Protocol*, AFRICACRYPT 2008, LNCS 5023, pp. 27-9, 2008.

[8] EPCglobal Inc., http://www.EPCglobalinc.org

[9] EPCglobal Ratified Specification, *The EPCglobal Architecture Framework*, 2009. http://www.epcglobalinc.org/standards/

[10] EPCglobal Ratified Standard, *EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860MHz - 960MHz Version 1.2.0*, 2008. http://www.epcglobalinc.org/standards/

[11] Feldhofer, M., Wolkerstorfer, J., and Rijmen, V., *AES implementation on a grain of sand*, IEE Proceedings In Information Security, vol. 152, no.1, pp. 13-20, 2005.

[12] Gilbert, H., Robshaw, M.J.B., and Seurin, Y., *Good variants of HB+ are hard to find*, Financial Cryptography`08, LNCS 5143, pp. 156-170, 2008.

[13] Gosset, F., Standaert, F.-X., and Quisquater, J.-J., *FPGA Implementation of SQUASH*, 29th Symposium on Information Theory in the Benelux, pp 231-238, 2008.

[14] Juels, A., and Weis, S., *Authenticating Pervasive Devices with Human Protocols*, CRYPTO`05, LNCS 3261, pp. 293-308, 2005.

[15] Karthikeyan, S., Nesterenko, M., *RFID security without extensive cryptography*, SASN'05, ACM, pp.63-67, 2005.

[16] Kim, K., *Construction of DES-like S-boxes based on Boolean Functions Satisfying the SAC*, Advances in Cryptology - Proc. of Asiacrypt'91, LNCS. 739, pp.59-72, 1991.

[17] Klaus, F., *RFID Handbook - Fundamentals and Applications in Contactless Smart Cards and Identification*, 2nd Edition, John Wiley & Sons, ISBN-13: 978-0470844021, 2003.

[18] Konidala, D.M., Kim, Z., and Kim, K., *A Simple and Cost-effective RFID Tag-Reader Mutual Authentication Scheme*, RFIDSec'07, pp.141-152, 2007.

[19] Leander, G., and Poschmann, A., *On the Classification of 4 Bit S-Boxes*, Arithmetic of Finite Fields - WAIFI`07, LNCS 4547, pp.159-176, 2007.

[20] L'Ecuyer, P., and Simard, R., *TestU01: a C library for empirical testing of random number generators*, ACM Transactions on Mathematical Software, vol.33, no.4, article 22, 2007. http://www.iro.umontreal.ca/~simardr/testu01/tu01.html

[21] Li, T., and Wang, G., *Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols*, 22nd IFIP TC-11, 2007.

[22] Li, T., and Deng, R.H., *Vulnerability Analysis of EMAP-An Efficient RFID Mutual Authentication Protocol*, AReS'07, 2007.

[23] Ouafi, K., Overbeck, R., and Vaudenay, S., *On the security of HB# against a man-in-the-middle attack*, ASIACRYPT`08, LNCS 5350, pp. 108-124, 2008.

[24] Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., and van der Lubbe, J.C.A., *Security Flaws in a Recent Ultra-lightweight RFID Protocol*, RFIDSec`10 Asia, 2010.

http://arxiv.org/PS_cache/arxiv/pdf/0910/0910.2115v1.pdf
[25] Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., and Ribagorda, A., *Cryptanalysis of a Novel Authentication Protocol Conforming to EPC-C1G2 standard*, Elsevier-Computer Standards & Interfaces, 31(2), pp.372-380, 2009.
[26] Peris-Lopez, P., Li, T., Lim, T.L., Hernandez-Castro, J. C., and Estevez-Tapiador, J. M., *Vulnerability Analysis of a Mutual Authentication Scheme under the EPC Class-1 Generation-2 Standard*, RFIDSec'08.
[27] Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., and Ribagorda, A., *Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol*, Workshop on Information Security Applications, LNCS 5379, pp. 56-68, 2008.
[28] Peris-Lopez, P., *Lightweight Cryptography in Radio Frequency Identification (RFID) Systems*, PhD. thesis, Computer Science Dept., Carlos III University of Madrid, 2008. http://www.lightweightcryptography.com/
[29] Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., and Ribagorda, A., *LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags*, RFIDSec'06, 2006.
[30] Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., and Ribagorda, A., *EMAP: An efficient mutual authentication protocol for low-cost RFID tags*, IS'06, LNCS 4277, pp.352-361, 2006.
[31] Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., and Ribagorda, A., *M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags*, UIC'06, LNCS 4159, pp.912-923, 2006.
[32] Rizomiliotis, P., Rekleitis, E., and Gritzalis, S., *Security analysis of the Song-Mitchell authentication protocol for low-cost RFID tags*, IEEE Communications Letters, vol 13, no. 4, pp. 274-276, 2009.
[33] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., and Vo, S., *A statistical test suite for random and pseudo-random number generators for cryptographic applications*, NIST special publication 800-22, National Institute of Standards and Technology (NIST), 2001. http://csrc.nist.gov/rng/
[34] Shannon, C.E., *Communication Theory of Secrecy Systems*, Bell System Technical Journal, vol.28-4, pp. 656-715, 1949.
[35] Song, B., and Mitchell, C.J., *RFID authentication protocol for low-cost tags*, First ACM Conference on Wireless Network Security, pp. 140-147, 2008.
[36] VeriSign, *The EPCglobal Network: Enhancing the Supply Chain*, White Paper 2005, http://www.verisign.com/stellent/groups/public/documents/white_paper/002109.pdf
[37] Zenner, E., *Authentication for RFID Tags: Observations on the HB Protocols*, 4th Interdisciplinary Seminar on Applied Mathematics, 2009.http://www.erikzenner.name/docs/2009_Aalborg_Talk.pdf

Divyan M. Konidala received the B.E. degree in Computer Science engineering from Bangalore University, India, in 2000 and the M.S. degree in the School of Engineering from the Information and Communications University [now merged with Korea Advanced Institute of Science and Technology (KAIST)], Daejeon, Korea, in 2004. He received the Ph.D. degree in Information and Communications Engineering, KAIST, Daejeon, Korea, in 2011.

His research interests include: Secure and privacy-preserving (cryptographic) protocols for: RFID, mobile-RFID, ubiquitous computing applications, location-based services, and mobile payments. Light-weight cryptology, block/stream ciphers, and digital signatures. GS1 EPCglobal RFID Specification Development–UHF Gen 2.

Prof. Kwangjo Kim has received the B.S and M.S. degrees of Electronic Engineering in Yonsei University, Korea in 1881 and 1983, respectively. He has also finished his Ph.D course in Div. of Electrical and Computer Engineering in Yokohama National University, Japan in 1991. He was Section Head in ETRI (1983~1997) and Professor at School of Engineering in ICU (1998~2008). Currently he is Full Professor at Computer Science Department in KAIST, Korea.

Prof. Kim has served President of KIISC (2009), Editor of JCN (~ 2007) and IJIS (~ 2008), and IEICE Special Issue on Information Security and Cryptography (2007). He was elected to serve Board of Director Member of IACR (1999~2004) and Chair of Asiacrypt Steering Committee (2005 ~ 2008).

He is currently Member of IEEE, IACR, and

IEICE. He is also Honorable President of KIISC and an editor of JMC. He has been served as Program Committee or Organizing Member in a variety of the international conferences on information security and cryptology more than about 10 times each year. He got award from Presidential citation (2009) in Korea.

Made Harta Dwijaksara received B.S. degree in Informatics Engineering from Institut Teknologi Bandung (ITB) in Indonesia in 2008. After graduation he was invited for IT-International Student Invitation Program by Ajou University, South Korea. He is currently a master degree student in department of computer science at the Korean Advanced Institute of Science and Technology (KAIST), South Korea under KT&G's scholarship program. His research interest is smart phone and wireless network security and privacy for ubiquitous environment.

Daeyoung Kim received the B.S. and M.S. degrees in computer science from Pusan National University, Busan, Korea, in 1990 and 1992, respectively, and the Ph.D. degree in computer engineering from the University of Florida, Gainesville, in 2001.

He was a Research Staff Member with the Electronics and Telecommunications Research Institute, Daejeon, from January 1992 to August 1997. From September 2001 to January 2002, he was a Research Assistant Professor with the Arizona State University, Tempe. He was an Associate Professor with the Department of Computer Science and Engineering, Information and Communications University, Daejeon, from 2002 to 2009. Since March 2009, he has been an Associate Professor with the Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Korea. He is a Director of Auto-ID Lab Korea (www.autoidlabs.org) and a Director of the Global USN National Research Laboratory. His research interests include sensor networks, real-time and embedded systems, and robotics.