Ultra-Lightweight Cryptography for Low-Cost RFID Tags: Hummingbird Algorithm and Protocol

Daniel Engels[2], Xinxin Fan[1], Guang Gong[1],
Honggang Hu[1], and Eric M. Smith[2]

[1] Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, N2L 3G1, CANADA
{x5fan@engmail, h7hu@ecemail, ggong@calliope}.uwaterloo.ca
[2] AuthentiCrypt Corporation
4500 Westgrove Drive, Suite 300, Addison, TX 75001, USA
esmith@authenticrypt.com

Abstract. Motivated by the design of the well-known Enigma machine, we present a novel ultra-lightweight encryption scheme, referred to as Hummingbird, and its applications to a privacy-preserving identification and mutual authentication protocol for RFID applications. Hummingbird can provide the designed security with a small block size and is therefore expected to meet the stringent response time and power consumption requirements described in the ISO protocol without any modification of the current standard. We show that Hummingbird is resistant to the most common attacks such as linear and differential cryptanalysis. Furthermore, we investigate some properties for integrating the Hummingbird into a privacy-preserving identification and mutual authentication protocol.

Keywords: RFID, lightweight cryptographic scheme, security analysis, privacy-preserving identification, and mutual authentication protocol.

1 Introduction

Radio Frequency Identification (RFID) is a rapidly developing technology enabling automatic object identification. In an RFID system, each object is labeled with a small transponder, called an RFID *tag*, which receives and responds to radio-frequency queries from a transceiver, called an RFID *reader*. An RFID tag is composed of a tiny integrated circuit for storing and processing identification information, as well as a radio antenna for wireless data transmissions. RFID tags usually have constrained capabilities in every aspect of computation, communication and storage due to the extremely low production cost. There are various applications for low-cost and low-power tags such as animal identification, point-of-sales, inventory management and so on.

Despite the low cost of RFID systems and their convenience in identifying an object without physical contact, the radio communications between RFID tags and readers also raise a number of security issues. For example, today's RFID systems do not conduct the mutual authentication between RFID readers and tags, so it is easy for an adversary to impersonate a reader or a tag to obtain sensitive information, and even launch Denial of Service (DOS) attacks. Moreover, RFID tags automatically emit their unique identifiers upon reader interrogation without alerting their users. Consequently, an adversary equipped with commodity RFID readers can effectively trace a person carrying a tagged item by linking two different sightings of the same RFID tag, which potentially violate

the owner's privacy. In addition, many possible security threats arise from unprotected wireless communications between RFID readers and tags.

To solve the aforementioned security and privacy issues, a privacy-preserving mutual authentication protocol is required for a reader and a tag to authenticate each other. After the mutual authentication, only a legitimate reader can access the contents of tags and the reader can be assured that the tags are authentic and have not been counterfeited at the same time. While a lot of effort has been made in designing authentication protocols for RFID systems over the past few years, we focus on the challenge-response protocols using symmetric key cryptography in this paper. To conduct the mutual authentication based on the challenge-response techniques, RFID tags must be able to execute secure symmetric key primitives. In [9, 10], Feldhofer *et. al.* proposed a low-power and compact ASIC core for 128-bit-key AES with 3400 gates. Their AES implementation only contains one S-box implemented as combinatorial logic and can encrypt a 128-bit data block within 1032 clock cycles. The authors also proposed a symmetric challenge-response authentication protocol which can be integrated into the existing ISO/IEC 18000 standard. Other work along the line of finding more compact AES implementations, say Hämäläinen *et. al.*'s work in [11] or designing new light cryptographic schemes, such as in [15] Leander *et. al.* suggested a lightweight DES variant called DESL (DES Lightweight), in [6] Bogdanov *et. al.* described an ultra-lightweight SP-network based block cipher PRESENT. Particularly, a serial version [18] of PRESENT can be implemented more compactly.

Feldhofer *et. al.*'s algorithm in [9] for integrating AES into the ISO/IEC 18000 standard specifically targets the ISO 18000-3 protocols. These are High Frequency (HF) protocols, i.e. 13.56MHz, where low power and small size are not significant advantages. EPCglobal class-1 generation-2 (EPC Gen2 in brief) [8] was approved as ISO 18000-6C in July 2006. It is widely believed that Gen2 tags will be the mainstream for RFID applications because of the large effective reading range. Note that EPC Gen2 protocols are Ultra High Frequency (UHF) protocols which require implementations that are significantly more power efficient and faster than the minimums required at HF. However, the very slow nature of AES requires a modification to the ISO 18000-3 protocol for the challenge-response to work.

In this paper, we present a new ultra-lightweight encryption scheme, referred to as Hummingbird, which are designed by Engels, Schultz, Schweitzer and Smith, for low-cost RFID tags and embedded micro chips. Hummingbird has a hybrid structure of block cipher and stream cipher and was developed with a minimal hardware footprint in mind. The hybrid model can provide the designed security with small block size and is therefore expected to meet the stringent response time and power consumption requirements described in the ISO 18000-6C protocol without any modification of the current standard.

This paper is organized as follows. Section 2 gives a description of the Hummingbird encryption and decryption scheme. The security analysis of the Hummingbird encryption scheme is presented in Section 3. In Section 4, we present the Hummingbird mutual authentication protocol and analyze its validity. Finally, Section 5 concludes this contribution.

Note that we have implemented Hummingbird in hardware. The hardware performance of Hummingbird will be discussed in a separate work.

2 The Hummingbird Encryption Scheme

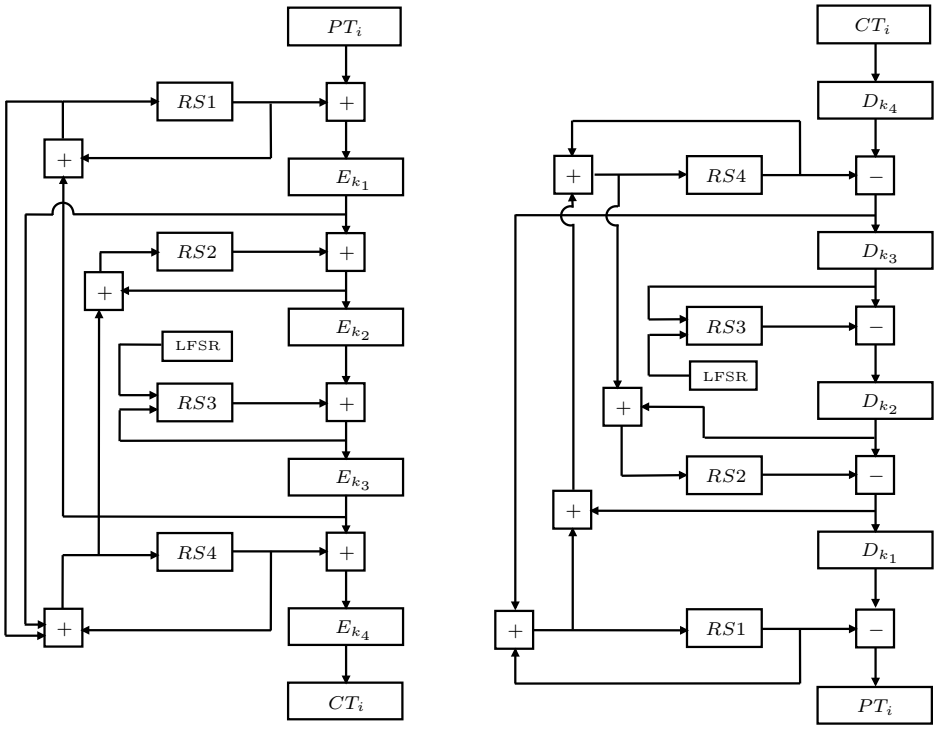
Different from existing (ultra-)lightweight cryptographic primitives which are either block ciphers or stream ciphers, Hummingbird is an elegant combination of the above two cipher structures with a 16-bit block size, 256-bit key size, and 80-bit internal state. The size of the key and the internal state of Hummingbird provides a security level which is adequate for many RFID applications. For clarity, we use the notation listed in Table 1 in the algorithm description. A top-level structure of the Hummingbird encryption is shown in the following Figure 1.

Table 1. Notation

PT_i	the i-th plaintext block, $i = 1, 2, \ldots, n$
CT_i	the i-th ciphertext block, $i = 1, 2, \ldots, n$
K	the 256-bit secret key
$\mathbf{E}_K(\cdot)$	the encryption function of Hummingbird with 256-bit secret key K
$\mathbf{D}_K(\cdot)$	the decryption function of Hummingbird with 256-bit secret key K
k_i	the 64-bit subkey used in the i-th block cipher, $i = 1, 2, 3, 4$, such that $K = k_1 \| k_2 \| k_3 \| k_4$
$E_{k_i}(\cdot)$	a block cipher encryption algorithm with 16-bit input, 64-bit key k_i, and 16-bit output, i.e., $E_{k_i} : \{0,1\}^{16} \times \{0,1\}^{64} \rightarrow \{0,1\}^{16}, i = 1, 2, 3, 4$
$D_{k_i}(\cdot)$	a block cipher decryption algorithm with 16-bit input, 64-bit key k_i, and 16-bit output, i.e., $D_{k_i} : \{0,1\}^{16} \times \{0,1\}^{64} \rightarrow \{0,1\}^{16}, i = 1, 2, 3, 4$
RSi	the i-th 16-bit internal state register, $i = 1, 2, 3, 4$
LFSR	a 16-stage Linear Feedback Shift Register with the characteristic polynomial $f(x) = x^{16} + x^{15} + x^{12} + x^{10} + x^7 + x^3 + 1$
$\boxplus$	modulo 2^{16} addition operator
$\boxminus$	modulo 2^{16} subtraction operator
$\oplus$	exclusive-or (XOR) operator
$m \ll l$	left circular shift operator, which rotates all bits of m to the left by l bits, as if the left and the right ends of m were joined.
$K_j^{(i)}$	the j-th 16-bit key used in the i-th block cipher, $j = 1, 2, 3, 4$, such that $k_i = K_1^{(i)} \| K_2^{(i)} \| K_3^{(i)} \| K_4^{(i)}$
S_i	the i-th 4-bit to 4-bit S-box used in the block cipher, $S_i : \mathbb{F}_2^4 \rightarrow \mathbb{F}_2^4, i = 1, 2, 3, 4$
NONCE_i	the i-th nonce which is a 16-bit random number, $i = 1, 2, 3, 4$

2.1 Encryption and Decryption

The overall structure of the Hummingbird encryption algorithm (see Figure 1(a)) consists of four 16-bit block ciphers $E_{k_1}, E_{k_2}, E_{k_3}$ and E_{k_4}, four 16-bit internal state registers $RS1, RS2, RS3$ and $RS4$, and a 16-stage LFSR. The 256-bit secret key K is divided into four 64-bit subkeys k_1, k_2, k_3 and k_4 which are used in the four block ciphers, respectively. A 16-bit plaintext block PT_i is encrypted by first executing a modulo 2^{16} addition of PT_i and the content of the first internal state register $RS1$. The result of the addition is then encrypted by the first block cipher E_{k_1}. This procedure is repeated in a similar manner for another three times and the output of E_{k_4} is the corresponding ciphertext CT_i. Furthermore, the states of the four internal state registers will also be updated in an unpredictable way based on their current states, the outputs of the first three block ciphers, and the state of the LFSR. The decryption process (see Figure 1(b)) follows the similar pattern as the



(a) Encryption Process (b) Decryption Process

Fig. 1. A Top-Level Description of the Hummingbird Cryptographic Algorithm

encryption. The exact encryption/decryption procedure and the internal state updating of Hummingbird are illustrated in the following Tabel 2. When using Hummingbird in practice, four 16-bit random nonce NONCE_i are first chosen to initialize the four internal state registers RSi $(i = 1, 2, 3, 4)$, respectively, followed by four consecutive encryptions on the message $RS1 \boxplus RS3$ by Hummingbird running in the initialization mode (see Figure 2). The final 16-bit ciphertext is used to initialize the LFSR. Moreover, the 13^{th} bit of the LFSR is always set to prevent a zero register. The LFSR is also stepped once before it is used to update the internal state register $RS2$. The initialization procedure of the Hummingbird is depicted in detail in Figure 2.

2.2 16-Bit Block Cipher

Four identical 16-bit block ciphers are employed in a consecutive manner in the Hummingbird encryption scheme. The 16-bit block cipher is a typical substitution-permutation (SP) network with 16-bit block size and 64-bit key as shown in Figure 3. It consists of 4 regular rounds and a final round that only includes the key mixing and the S-box substitution steps. Like any other SP network, one regular round comprises of three stages: a key mixing step, a substitution layer, and a permutation layer. For the key mixing, a simple exclusive or operation is used in this 16-bit block cipher for efficient implementation in both software and hardware. The substitution layer is composed of 4 Serpent-type S-boxes

Table 2. Encryption/Decryption and Internal State Updating of Hummingbird

Encryption Process	Decryption Process
$V12_t = E_{k_1}(PT_i \boxplus RS1_t)$	$V34_t = D_{k_4}(CT_i) \boxminus RS4_t$
$V23_t = E_{k_2}(V12_t \boxplus RS2_t)$	$V23_t = D_{k_3}(V12_t) \boxminus RS3_t$
$V34_t = E_{k_3}(V23_t \boxplus RS3_t)$	$V12_t = D_{k_2}(V23_t) \boxminus RS2_t$
$CT_i = E_{k_4}(V34_t \boxplus RS4_t)$	$PT_i = D_{k_1}(V34_t) \boxminus RS1_t$
Internal State Updating Process	
$\mathrm{LFSR}_{t+1} \leftarrow \mathrm{LFSR}_t$	
$RS1_{t+1} = RS1_t \boxplus V34_t$	
$RS3_{t+1} = RS3_t \boxplus V23_t \boxplus \mathrm{LFSR}_{t+1}$	
$RS4_{t+1} = RS4_t \boxplus V12_t \boxplus RS1_{t+1}$	
$RS2_{t+1} = RS2_t \boxplus V12_t \boxplus RS4_{t+1}$	

[1] with 4-bit inputs and 4-bit outputs, having additional properties whose selecting criteria will be given in Appendix A. The action of the four S-boxes in hexadecimal notation is also described in Figure 3. The permutation layer in this 16-bit block cipher is given by the linear transform $L : \{0,1\}^{16} \rightarrow \{0,1\}^{16}$ defined as follows:

$$L(m) = m \oplus (m \lll 6) \oplus (m \lll 10),$$

where $m = (m_0, m_1, \cdots, m_{15})$ is a 16-bit data block.

2.3 Design Rationale of Hummingbird

The design of the Hummingbird encryption scheme is motivated by the well-known Enigma machine and takes into account both security and efficiency simultaneously. The encryption/decryption process of the Hummingbird can be viewed as the continuous running of a rotor machine, where four small block ciphers act as four virtual rotors which perform permutations on 16-bit words. The salient characteristics of the Hummingbird lies in implementing extraordinarily large virtual rotors with custom block ciphers and using successively changing internal states to step each virtual rotor in various and unpredictable ways. Besides a novel cipher structure, the Hummingbird is also designed to protect against the most common attacks such as linear and differential cryptanalysis, which will be discussed in detail in Section 3. Moreover, extremely simple arithmetic and logic operations are extensively employed in the Hummingbird, which make it well-suited for resource-constrained environments.

Besides the novel cipher structure and the efficient implementation, Hummingbird fits very neatly within the ISO 18000-6C protocol, specifically within the existing identification process, and can be used in conjunction with the ISO protocol to perform a secure identification and mutual authentication process between an RFID reader and a tag.

2.4 The Selection of S-Boxes

Based on the nine criteria presented in Appendix A, we select 4 S-boxes, which are listed in Table 3.

Nonce Initialization:

$RS1_0 = \mathsf{NONCE}_0$
$RS2_0 = \mathsf{NONCE}_1$
$RS3_0 = \mathsf{NONCE}_2$
$RS4_0 = \mathsf{NONCE}_3$

Four Rounds Encryption:

for $t = 0$ to 3 **do**

$V12_t = E_{k_1}\left((RS1_t \boxplus RS3_t) \boxplus RS1_t\right)$
$V23_t = E_{k_2}(V12_t \boxplus RS2_t)$
$V34_t = E_{k_3}(V23_t \boxplus RS3_t)$
$TV_t = E_{k_4}(V34_t \boxplus RS4_t)$
$RS1_{t+1} = RS1_t \boxplus TV_t$
$RS2_{t+1} = RS2_t \boxplus V12_t$
$RS3_{t+1} = RS3_t \boxplus V23_t$
$RS4_{t+1} = RS4_t \boxplus V34_t$

end for

LFSR Initialization:

LFSR $= TV_3 \mid$ 0x1000

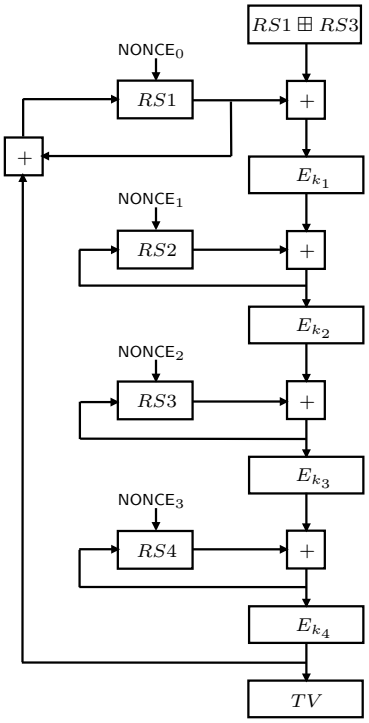


Fig. 2. The Initialization Procedure of the Hummingbird Cryptographic Algorithm

Table 3. Four S-Boxes

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_1(x)$	8	6	5	F	1	C	A	9	E	B	2	4	7	0	D	3
$S_2(x)$	0	7	E	1	5	B	8	2	3	A	D	6	F	C	4	9
$S_3(x)$	2	E	F	5	C	1	9	A	B	4	6	8	0	7	3	D
$S_4(x)$	0	7	3	4	C	1	A	F	D	E	6	B	2	8	9	5

Remark 1. To further reduce the consumption of the area and power of Hummingbird in hardware implementation, four S-boxes used in Hummingbird can be replaced by a single S-box, which is repeated four times in the 16-bit block cipher. The security analysis of the compact version of Hummingbird will be discussed in a separate work.

3 Security Analysis of the Hummingbird Encryption Scheme

Hummingbird encryption scheme is a hybrid mode of block cipher and stream cipher, which can be considered as a finite state machine where $(RS1, RS2, RS3, RS4, \text{LFSR})$ is the internal state. But the value of LFSR does not depend on those of $RS1$, $RS2$, $RS3$, and $RS4$. The purpose of employing the LFSR is to guarantee the period of the internal state is at least 2^{16}.

A. Birthday Attack on the Initialization. For a fixed key, one may want to find two identical internal states $(RS1, RS2, RS3, RS4, \text{LFSR})$ initialized by two different IVs using birthday attack. However, for the initialization procedure of the Hummingbird encryption scheme, if we fix the key, then the mapping $(RS1_t, RS2_t, RS3_t, RS4_t) \rightarrow (RS1_{t+1}, RS2_{t+1},$

Four S-Boxes Given in Hexadecimal Notation

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_1(x)$	8	6	5	F	1	C	A	9	E	B	2	4	7	0	D	3
$S_2(x)$	0	7	E	1	5	B	8	2	3	A	D	6	F	C	4	9
$S_3(x)$	2	E	F	5	C	1	9	A	B	4	6	8	0	7	3	D
$S_4(x)$	0	7	3	4	C	1	A	F	D	E	6	B	2	8	9	5

Linear Transform $L : \{0,1\}^{16} \rightarrow \{0,1\}^{16}$

$L(m) = m \oplus (m \lll 6) \oplus (m \lll 10)$

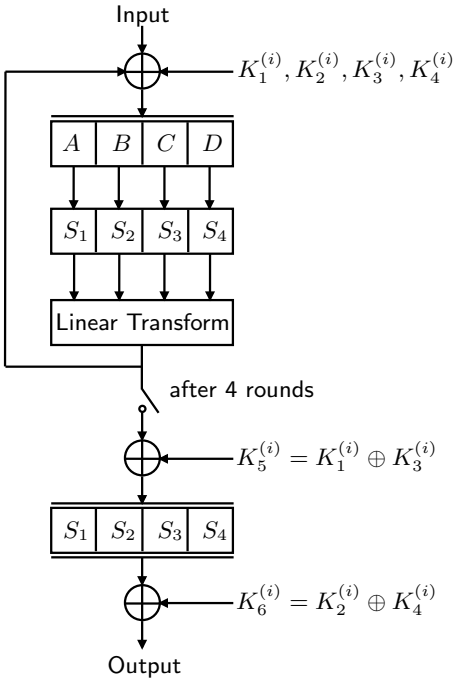


Fig. 3. The Structure of the Block Cipher in the Hummingbird Cryptographic Algorithm

$RS3_{t+1}, RS4_{t+1}$) is one-to-one. Hence the birthday attack does not work for the initialization procedure.

B. Differential Cryptanalysis. Let $\mathbf{E}_K(x)$ denote the encryption function of Hummingbird with 256-bit key K. Recall that $E_k(x)$, defined in Section 2, denotes the 16-bit block cipher encryption used in Hummingbird with 64-bit key k. Then $\mathbf{E}_K(x)$ is the composition of four $E_k(x)$. For a function $F(x)$ from $\mathbb{F}_2^m$ to $\mathbb{F}_2^m$, the differential between $F(x)$ and $F(x+a)$, where $+$ is the bit-wise addition by $D_F(a,b)$, is given by

$$D_F(a,b) = |\{x \,|\, F(x) + F(x+a) = b,\, x \in \mathbb{F}_2^m\}|.$$

For many keys, we computed the differentials of both $\mathbf{E}_K(x)$ and $E_k(x)$. Note that from Section 2, we know that there are five rounds in $E_k(x)$. We list the differential $D_{E_k}(a,b)$ for each round in Table (A) in Appendix B. For a substantially large amount of initial vectors IV and keys K, the differentials for both $E_k(x)$ and $\mathbf{E}_K(x)$ satisfy the following inequalities.

$$\max_{a,b \in \mathbb{F}_2^{16}, a \neq 0} \{D_{E_k}(a,b)\} \leq 20, \text{ and } \max_{a,b \in \mathbb{F}_2^{16}, a \neq 0} \{D_{\mathbf{E}_K}(a,b)\} \leq 20.$$

In other words, the differential of $\mathbf{E}_K(x)$ has the same upper bound as $E_k(x)$, the block cipher component in $\mathbf{E}_K$. We also tested the reduced version of Hummingbird for more instances of different pairs of (IV, K). From those experimental results, in general, the standard differential cryptanalysis method is not applicable for Hummingbird with practical time complexity.

C. Linear Cryptanalysis. For the linear cryptanalysis of $\mathbf{E}_K(x)$, we need to consider $|\hat{\mathbf{E}}_K(a,b)|$, the absolute value of the Walsh transform of $\mathbf{E}_K(x)$, where

$$\hat{\mathbf{E}}_K(a,b) = \sum_{x \in \mathbb{F}_2^{16}} (-1)^{<a,\mathbf{E}_K(x)>+<b,x>}, \ a,b \in \mathbb{F}_2^{16}, a \neq 0$$

where $< x, y >$ is the inner product of two binary vectors x and y (see Appendix A for the detail). Unlike the case for the differential of $\mathbf{E}_K(x)$ or $E_k(x)$, we cannot perform an exhaustive computation for $|\hat{\mathbf{E}}_K(a,b)|$ for all $a,b \in \mathbb{F}_2^{16}, a \neq 0$, since there are 2^{47} instances for (a,b) that need to be verified for a pair of fixed IV and key K. For many fixed pairs of (IV,K), we have exhaustively computed the cases for a,b with small Hamming weights, i.e., for all a,b with $wt(a), wt(b) \leq 3$ where $wt(x)$ is the Hamming weight of x, and some random pairs of (a,b). Those experimental results show that $|\hat{\mathbf{E}}_K(a,b)| \leq c \cdot \sqrt{2^{16}}$ where $c \leq 4.96875$. We list some data in Table (B) in Appendix B. We may view $\mathbf{E}_K(x)$ as a vectorial boolean function from $\mathbb{F}_2^{16}$ to $\mathbb{F}_2^{16}$. Then the above experimental results show that the correlation between each individual component of $\mathbf{E}_K(x)$ and linear function $< b, x >$ is low where $< b, x >= \sum_{i=1}^{16} b_i x_i$ where $b_i, x_i \in \{0,1\}$ with $wt(b) \leq 3$, so does a combination of any two components or three components of $\mathbf{E}_K(x)$. We also conducted the experiments for an 8-bit version of the Hummingbird encryption scheme which means that all the rotors $RS_i, i = 1,2,3,4$ and LFSR contain only 8 bits. The Walsh transform of this reduced version of Hummingbird is bounded by $5 \cdot \sqrt{2^8}$ for many pairs of IV and key. This is the supporting evidence (albeit weak) that the absolute value of the Walsh transform of Hummingbird encryption function could be bounded by the square root of 2^{16} multiplying by a constant. Hence, Hummingbird is resistant to linear cryptanalysis attack with practical time complexity in RFID applications.

D. Structural Attack. The Hummingbird encryption scheme may be viewed as a certain operation mode of a block cipher. For example, the ciphertext can be viewed as the internal state of a block cipher in CBC mode. In [2, 3], Biham investigated some operation modes of block ciphers. He found that many triple modes are not as secure as one expected. In [4], Biham and Knudsen broke the ANSI X9.52 CBCM Mode. However, the internal state transition in Hummingbird encryption scheme is much more complicated than those studied by [2–4]. Hence, those attacks cannot be simply applied to the Hummingbird encryption scheme. In [19], by choosing IV, Wagner presented some new attacks on some modes proposed by Biham. Because IV initialization is used in the Hummingbird encryption scheme, Wagner's attacks are not applicable.

E. Algebraic Attacks. For the Hummingbird encryption scheme, for each S-box in a block cipher $E_k(x)$, its degree is maximized. In each block cipher $E_k(x)$, there are five rounds. Thus, in total, there are 20 rounds for the Hummingbird encryption scheme, i.e., $\mathbf{E}_K(x)$. On the other hand, the internal state transition involves modulo 2^{16} operation. Hence it is hard to apply efficient linearization techniques for algebraic attacks to Hummingbird.

F. Cube Attack. The success probability of the cube attack is high if the degree of the internal state transit function in a stream cipher is low. For example, the degree of internal state transit function of Trivium grows slowly [7]. However, for the Hummingbird encryption scheme, the degree of the internal state transit function is very high. In addition, Hummingbird encryption scheme is a hybrid mode of the block cipher and stream cipher. We have tested both the 16-bit block cipher $E_k(x)$ used in the Hummingbird encryption scheme and the Hummingbird encryption $\mathbf{E}_K(x)$. There are no linear equations of key

bits that can be used in the way as suggested in [7]. Note that the cube attack can be considered as a special case of high order differential cryptanalysis, proposed in [14]. Thus, it is worth to look at the high order differentials of Hummingbird encryption scheme which will be conducted in future.

4 The Hummingbird Mutual Authentication Protocol

The Hummingbird mutual authentication protocol is used to establish the trust relationship between RFID tags and readers based on the highly-efficient Hummingbird cryptographic algorithm. Assume that an RFID system consists of a reader and N RFID tags (N might be as large as one billion in a typical application scenario) and the reader shares a unique 256-bit key with each tag. The Hummingbird mutual authentication protocol includes the following three phases.

Phase 1. Privacy-Preserving Tag Identification

In this phase, the reader will determine the correct key shared with a tag it is communicating with without exposing the tag's identity to adversaries by performing a private identification procedure [12] as shown in Figure 4.

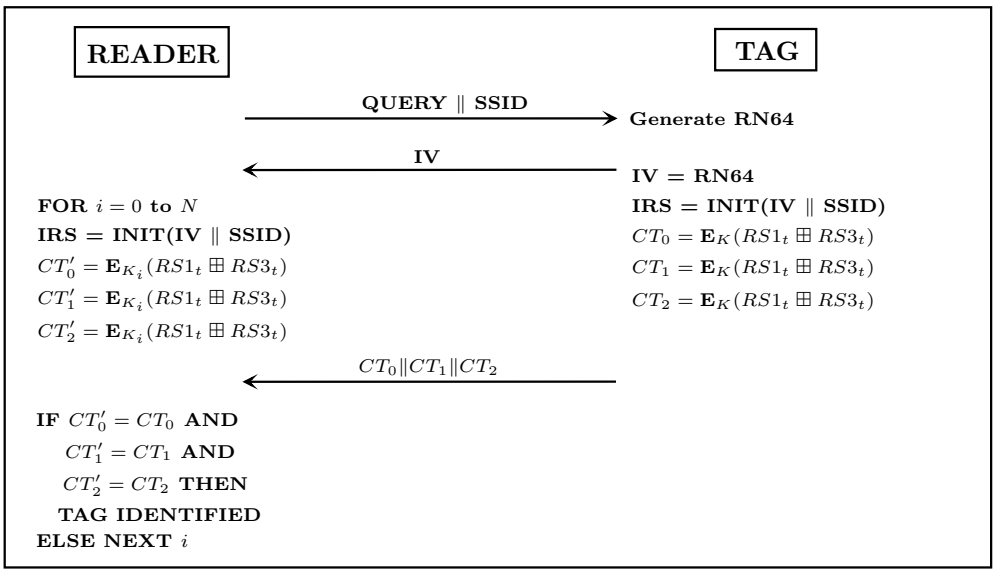


Fig. 4. The Hummingbird Private Identification Protocol

In the above private identification protocol, the reader first sends a QUERY together with a 16-bit session ID (SSID) to a tag. Upon receiving the QUERY and the SSID, the tag generates four 16-bit random numbers (IV) and transmits IV to the reader. Furthermore, the tag also uses IV as well as SSID to initialize four 16-bit internal state registers. After that, the tag encrypts the message $RS1 \boxplus RS3$ with the Hummingbird encryption algorithm three times and uses the resulting three ciphertexts CT_0, CT_1 and CT_2 as the tag indicators

which enable the reader to uniquely identify a tag without knowing the tag's identity. The reader initializes four 16-bit internal state registers with IV and SSID and performs a brute force search to identify the tag privately. More specifically, the reader tries every key stored in the database and computes three ciphertexts CT_0', CT_1' and CT_2' as the tag does until a match is found (i.e., $CT_0' = CT_0, CT_1' = CT_1$ and $CT_2' = CT_2$). If no any match occurs, the reader rejects the tag.

Note that for an RFID system with about one billion tags, a tag needs to generate three indicators (i.e., CT_0, CT_1 and CT_2 in Figure 4) in order to be uniquely identified by a reader in the Hummingbird private identification protocol. We next analyze this interesting property by showing that how many tag indicators (i.e., successive encryptions of internal states) are required for a reader to uniquely identify a tag in an RFID system with N tags. The analysis is based on the following basic fact in probability theory.

Fact 1. *Let $\mathcal{S}$ be a set of order n, and h be a given element in $\mathcal{S}$. Then on average we need to select $(n+1)/2$ elements such that h may be selected.*

Let $\mathbf{E}$ be a stream cipher with key length n, and every time $\mathbf{E}$ outputs m bits. Let $\mathcal{S}$ be the set of all users' keys, and for any two keys, the outputs associated with them are different. For any $k \in \mathcal{S}$, we denote the output of $\mathbf{E}$ under k by $\mathbf{E}_{k,0}, \mathbf{E}_{k,1}, \mathbf{E}_{k,2}, ...$, where $\mathbf{E}_{k,i}$ is m bits, $i = 0, 1, 2, ...$ Given the output $b_0, b_1, b_2, ...$ of one user's key, in order to identify such user, we may perform a brute force search if $|\mathcal{S}|$ is practical.

1) Select one key k from $\mathcal{S}$.
2) Compute $\mathbf{E}_{k,0}$. If $\mathbf{E}_{k,0} \neq b_0$, goto 1).
3) Compute $\mathbf{E}_{k,1}$. If $\mathbf{E}_{k,1} \neq b_1$, goto 1).
4) Compute $\mathbf{E}_{k,2}$. If $\mathbf{E}_{k,2} \neq b_2$, goto 1).
5) ...

Using the above Fact 1, we can prove the following theorem:

Theorem 1. *With the protocol above, the number of computing $\mathbf{E}_{k,t}$ is around $|\mathcal{S}|/2^{mt+1}, t = 0, 1, 2, ...$ Let q be the integer such that $2^{(q-1)m} < |\mathcal{S}| \leq 2^{qm}$. Then q outputs $b_0, b_1, b_2, ..., b_{q-1}$ are enough to identify one user.*

Proof. By Fact 1, we need to select around $|\mathcal{S}|/2$ keys to identify one user. Hence the number of computing $\mathbf{E}_{k,0}$ is $|\mathcal{S}|/2$. Among such $|\mathcal{S}|/2$ keys, on average $|\mathcal{S}|/2^{m+1}$ keys can pass 2). Hence the number of computing $\mathbf{E}_{k,1}$ is $|\mathcal{S}|/2^{m+1}$. Similarly, for any $t = 2, 3, ...,$ the number of computing $\mathbf{E}_{k,t}$ is around $|\mathcal{S}|/2^{mt+1}$. Because only $1/2^m$ keys may pass each step. So q outputs are enough to identify one user. □

Example. If $m = 16$ and $|\mathcal{S}| = 2^{30}$, the number of computing $\mathbf{E}_{k,0}$ is around $2^{29} = 536870912$, and the number of computing $\mathbf{E}_{k,1}$ is around $2^{13} = 8192$. Among such 2^{13} keys, because $2^{13} < 2^{16}$, only one can pass 3). Hence the number of computing $\mathbf{E}_{k,2}$ is 1.

In practice, we tested a database with one billion keys. On average, the reader would have to perform 500 million CT_0 calculations, then 7629 CT_1 calculations, and then one CT_2 calculation. This experimental result confirms the theoretical analysis. Moreover, a single FPGA (Virtex 5 clocked at 500MHz) is capable of performing 16 billion Hummingbird encryptions per second. Hence on average a reader would be able to identify a correct key in a database of one billion keys within $156.25ms$ or 64 million keys within $10ms$.

Phase 2. Mutual Authentication between a Reader and a Tag

In this phase, the tag and the reader authenticates each other based on the shared secret as shown in the following Figure 5.

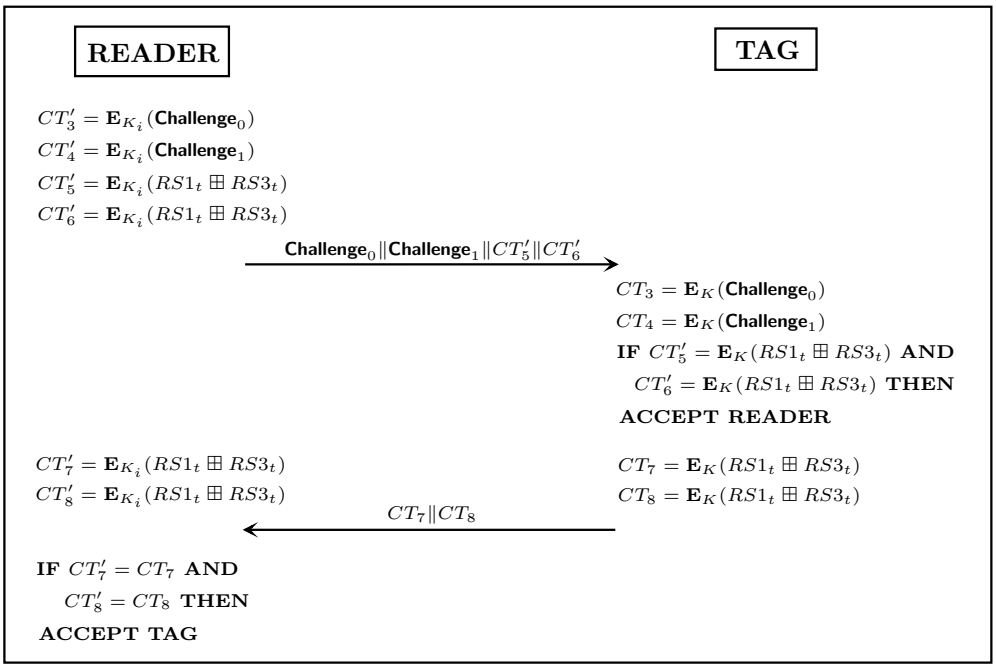


Fig. 5. The Hummingbird Mutual Authentication Protocol

Once the tag is identified in Phase 1, the reader generates two 16-bit random challenges, $\mathsf{Challenge}_0$ and $\mathsf{Challenge}_1$, and encrypts them based on the current internal state of the Hummingbird encryption algorithm. The reader also computes two authenticators CT'_5 and CT'_6, which involves the encryption of successive internal state $RS1 \boxplus RS3$, as the response. Two random challenges $\mathsf{Challenge}_0$ and $\mathsf{Challenge}_1$ and two authenticators CT'_5 and CT'_6 are then transmitted to the tag. After receiving two random challenges $\mathsf{Challenge}_0$ and $\mathsf{Challenge}_1$ from the reader, the tag first encrypts them and throws away the resulting ciphertexts CT_3 and CT_4. The tag then calculates CT_5 and CT_6 to check whether they match the received authenticators CT'_5 and CT'_6, respectively. If there is a match, the tag authenticates the reader successfully. The tag will then calculate two authenticators CT_7 and CT_8 and send them to the reader as the response. After receiving the response from the tag, the reader encrypts the internal state $RS1 \boxplus RS3$ twice and compares the resulting ciphertexts CT'_7 and CT'_8 with the received values CT_7 and CT_8, respectively. If there is a match, the tag is authenticated.

Phase 3. Command Execution

In this phase, an encryption-only RFID tag receives and executes an command issued by an authenticated reader as shown in the following Figure 6.

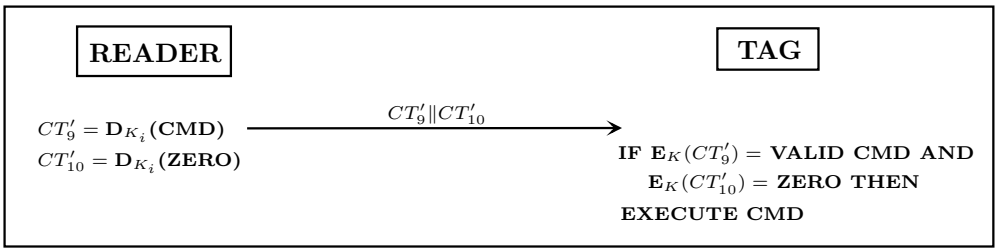


Fig. 6. The Reader Issues a Command to The Tag

Once the reader and the tag complete the mutual authentication, the reader transmits two messages CT'_9 and CT'_{10} to the tag, which comprise of the decryption of a command and the decryption of a 16-bit all zero word. The tag, which can only execute the Hummingbird encryption algorithm, encrypts the received messages to retrieve the command and the all zero word, respectively. If the command is valid and includes appropriate number of zeros, the tag will execute the command.

To meet the requirements of a variety of application scenarios, the idea of modular design is adopted in the Hummingbird mutual authentication protocol. More specifically, the Hummingbird protocol separates the phases of the private identification and the mutual authentication. Note that some RFID applications only need identification whereas others require both identification and authentication. Therefore, users are able to flexibly choose different modules for specific applications.

5 Concluding Remarks

In this paper we present a novel ultra-lightweight encryption scheme, Hummingbird, which is a combination of the block cipher and stream cipher, and its application to privacy-preserving identification and authentication for RFID. We show that Hummingbird is resistant to the most common attacks to block ciphers and stream ciphers including birthday attacks, differential and linear cryptanalysis, structure attacks, algebraic attacks, and cube attacks. We also describe a validity of the Hummingbird privacy identification and mutual authentication protocol. As our future work, we will further investigate and generalize the operation mode of the Hummingbird encryption scheme and their security analysis. Moreover, a more compact hardware implementation with 4-bit datapath is also desirable.

The combination of the block cipher and stream cipher in Hummingbird can provide the designed security with small block size which can meet the stringent response time and power consumption requirements described in the ISO 18000-6C protocol without any modification of the current standard. From the experimental results, Hummingbird encrypts 16 bits at a time and does so within 20 clock cycles. Thus, in the privacy-preserving identification and mutual authentication process between an RFID reader and a tag, the first block and a half can be encrypted before a response starts, and the response can start on time with an encrypted block being sent out by the tag while the next block is being encrypted. Note that the smallest tag response parameter given in the ISO protocol is $15.625\mu s$ (micro-seconds). Our empirical evaluation of the Gen2 artifacts is

that they operate in the MHz range for their on-tag functions. In particular, 16 bits can be encrypted in $20\mu s$ and $15.625\mu s$ at 1MHz and 1.28MHz, respectively. Thus, we can integrate Hummingbird into the Gen2 identification and authentication process without slowing down that process.

References

1. R. Anderson, E. Biham, and L. Knudsen, "Serpent: A Proposal for the Advanced Encryption Standard", available at `http://www.cl.cam.ac.uk/~rja14/Papers/serpent.pdf`.
2. E. Biham, "Cryptanalysis of Multiple Modes of Operation", *J. Cryptology* 11(1), pp. 45-58, 1998.
3. E. Biham, "Cryptanalysis of Triple Modes of Operation", *J. Cryptology* 12(3), pp. 161-184, 1999.
4. E. Biham and L. R. Knudsen, "Cryptanalysis of the ANSI X9.52 CBCM Mode", *J. Cryptology* 15(1), pp. 47-59, 2002.
5. E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, New York, 1993.
6. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher", *The 9th International Workshop on Cryptographic Hardware and Embedded Systems - CHES 2007*, LNCS 4727, P. Paillier and I. Verbauwhede (eds.), Berlin, Germany: Springer-Verlag, pp. 450-466, 2007.
7. I. Dinur and A. Shamir, "Cube Attacks on Tweakable Black Box Polynomials", *Advances in Cryptology - EUROCRYPT 2009*, LNCS 5479,A. Joux (ed.), Berlin, Germany: Springer-Verlag, pp. 278-299, 2009.
8. EPCglobal, Inc., `http://www.epcglobalinc.org/`, 2005.
9. M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong Authentication for RFID Systems Using the AES Algorithm", *The 6th International Workshop on Cryptographic Hardware and Embedded Systems-CHES 2004*, LNCS 3156, M. Joye and J.-J. Quisquater (eds.), Berlin, Germany: Springer-Verlag, pp. 357-370, 2004.
10. M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "AES Implementation on a Grain of Sand", *IEE Proceedings Information Security*, vol. 15, no. 1, pp. 13-20, 2005.
11. P. Hämäläinen, T. Alho, M. Hännikäinen, and T. D. Hämäläinen, "Design and Implementation of Low-Area and Low-Power AES Encryption Hardware Core", *The 9th EUROMICRO Conference on Digital System Design: Architectures, Methods and Tools - DSD 2006*, pp. 577-583, IEEE Computer Society, 2006.
12. International Organization for Standardization, *ISO/IEC 9782-2: Information Technology – Security Techniques – Entity Authentication Mechanisms Part 2: Entity Authentication using Symmetric Techniques*, 1993.
13. T. Jakobsen and L. Knudsen, "The Interpolation Attack on Block Ciphers", *The 4th Annual Fast Software Encryption Workshop - FSE 1997*, LNCS 1267, E. Biham (ed.), Berlin, Germany: Springer-Verlag, pp. 28-40, 1997.
14. X. Lai, "Higher Order Derivatives and Differential Cryptanalysis", *Proceedings of Symposium on Communication, Coding and Cryptography*, in honor of James L. Massey on the occasion of his 60'th birthday, 1994.
15. G. Leander, C. Paar, A. Poschmann, and K. Schramm, "New Lightweight DES Variants", *The 14th Annual Fast Software Encryption Workshop - FSE 2007*, LNCS 4593, A. Biryukov (ed.), Berlin, Germany: Springer-Verlag, pp. 196-210, 2007.
16. G. Leander and A. Poschmann, "On the Classification of 4 Bit S-Boxes", *The 1st International Workshop on the Arithmetic of Finite Fields - WAIFI 2007*, LNCS 4547, C. Carlet and B. Sunar (eds.), Berlin, Germany: Springer-Verlag, pp. 159-176, 2007.
17. M. Matsui, "Linear Cryptanalysis Method for DES Cipher", *Advances in Cryptology - EUROCRYPT'93*, LNCS 765, T. Helleseth (ed.), Berlin, Germany: Springer-Verlag, pp. 386-397, 1993.
18. C. Rolfes, A. Poschmann, G. Leander, and C. Paar, "Ultra-Lightweight Implementations for Smart Devices-Security for 1000 Gate Equivalents", *The 8th Smart Card Research and Advanced Application IFIP Conference - CARDIS 2008*, LNCS 5189, G. Grimaud and F.-X. Standaert (eds.), Berlin, Germany: Springer-Verlag, pp. 89-103, 2008.
19. D. Wagner, "Cryptanalysis of Some Recently-Proposed Multiple Modes of Operation", *The 5th Annual Fast Software Encryption Workshop - FSE 1998*, LNCS 1372, S. Vaudenay (ed.), Berlin, Germany: Springer-Verlag, pp. 254-269, 1998.

20. A. Youssef and G. Gong, "On the Interpolation Attacks on Block Ciphers", *The 7th Annual Fast Software Encryption Workshop - FSE 1998*, LNCS 1978, B. Schneier (ed.), pp. 109-120, Berlin, Germany: Springer-Verlag, 2001.

Appendix A. Criteria for Selection of S-Boxes

Let $\mathbb{F}_2 = \{0, 1\}$ and $\mathbb{F}_2^m = \{(x_0, x_1, \cdots, x_{m-1}) \,|\, x_i \in \mathbb{F}_2\}$. If $F(x)$ is a function from $\mathbb{F}_2^m$ to $\mathbb{F}_2^m$, i.e., a vectorial boolean function, then it is also called a S-box from from $\mathbb{F}_2^m$ to $\mathbb{F}_2^m$. The Walsh transform of $F(x)$ is defined by

$$\widehat{F}(a, b) = \sum_{x \in \mathbb{F}_2^m} (-1)^{<a, F(x)> + <b, x>}, \ a, b \in \mathbb{F}_2^m$$

where $< y, z >= \sum_{i=0}^{m-1} y_i z_i$, the inner product of two binary vectors $y = (y_0, y_1, \cdots, y_{m-1})$ and $z = (z_0, z_1, \cdots, z_{m-1})$ where $y_i, z_i \in \mathbb{F}_2$.

A. Serpent-type S-boxes

Let $S(x)$ be an S-box from $\mathbb{F}_2^4$ to $\mathbb{F}_2^4$. Then the Walsh transform of $S(x)$ is given by by

$$\widehat{S}(a, b) = \sum_{x \in \mathbb{F}_2^4} (-1)^{<a, S(x)> + <b, x>}$$

for any $a, b \in \mathbb{F}_2^4$. $S(x)$ is called a Serpent-type S-box if it satisfies the following properties.

(a) For any nonzero $a, b \in \mathbb{F}_2^4$, $|\widehat{S}(a, b)| \leq 8$.
(b) For any $a, b \in \mathbb{F}_2^4$ with $wt(a) = wt(b) = 1$, $|\widehat{S}(a, b)| = 4$ where $wt(x)$ represents the Hamming weight of a binary string $x \in \mathbb{F}_2^m$.
(c) For any nonzero $a, b \in \mathbb{F}_2^4$,

$$|\{x \in \mathbb{F}_2^4 | S(x) + S(x + a) = b\}| \leq 4.$$

(d) For any $a, b \in \mathbb{F}_2^4$ with $wt(a) = wt(b) = 1$,

$$|\{x \in \mathbb{F}_2^4 | S(x) + S(x + a) = b\}| = 0.$$

The first two properties are for resistance to linear cryptanalysis and the last two, for resistance to differential cryptanalysis. In [16], all Serpent-type S-boxes are classified, and presented by a representative in one class.

B. Additional Requirements for Serpent-type S-boxes

We can write $S(x)$ in terms of its boolean form as follows.

$$S(x) = (f_0(x_0, x_1, x_2, x_3), f_1(x_0, x_1, x_2, x_3), f_2(x_0, x_1, x_2, x_3), f_3(x_0, x_1, x_2, x_3)).$$

In order to resist the algebraic attacks, we select S-boxes such that the degree of f_i is 3 for $0 \leq i \leq 3$, i.e., all components of $S(x)$ should have maximum degree. For hardware implementation, the number of terms of f_i should be small.

On the other hand, $S(x)$ can be viewed as a mapping from $\mathbb{F}_{2^4}$ to $\mathbb{F}_{2^4}$ where $\mathbb{F}_{2^4}$ is a finite field with 2^4 elements. So $S(x)$ has a polynomial representation over $\mathbb{F}_{2^4}$. Consequently, any component of $S(x)$ has a polynomial form over $\mathbb{F}_{2^4}$. In order to resist the interpolation attack, $S(x)$ and f_i should have as many monomial terms as possible under a polynomial representation [13, 20].

According to the above considerations, in addition to the Serpent-type properties, we list the criteria for selection of S-boxes as follows.

(e) The algebraic degrees of all four component functions in boolean representation are maximized, i.e., 3.

(f) The number of monomial terms of each component function in boolean representation is small.

(g) All component functions should have an approximately same number of monomial terms.

(h) The number of monomial terms of each component function in a polynomial representation should be large under all defining polynomials for $\mathbb{F}_{2^4}$.

(i) Select one S-box from each equivalent class.

The four S-boxes are selected according to the above nine criteria. For the number of monomial terms in both boolean and polynomial representations of the selected S-boxes, please see Tables 4 and 5 below.

Note that maximum number of monomial terms for a S-box from $\mathbb{F}_2^4$ to $\mathbb{F}_2^4$ or its component function is 15 whenever it is represented in a boolean form or a polynomial form. In Table 4, the number under f_i is the number of monomial terms in f_i. For example, for S_1, both f_0 and f_1 have 7 monomial terms, and both f_2 and f_3 have 6 monomial terms.

Table 4. The Boolean Form of S-Boxes

	f_0	f_1	f_2	f_3	Total Monomial Terms	Least Degree
S1	7	7	6	6	26	3
S2	7	8	6	7	28	3
S3	7	6	7	7	27	3
S4	7	6	7	6	26	3

In Table 5, the number under each defining polynomial of $\mathbb{F}_{2^4}$ denotes the number of monomial terms in each corresponding function. For example, under the defining polynomial $x^4 + x + 1$ for $\mathbb{F}_{2^4}$, S-box $S_1(x)$ has 15 terms, and the components $f_i, i = 0, 1, 2, 3$ have 14, 14, 14, and 15 monomial terms respectively.

Table 5. The Polynomial Form of S-Boxes

		x^4+x+1	x^4+x^3+1	$x^4+x^3+x^2+x+1$
S1	$S_1(x)$	15	15	14
	f_0	14	14	14
	f_1	14	14	14
	f_2	14	14	14
	f_3	15	13	15
S2	$S_2(x)$	13	13	14
	f_0	14	14	12
	f_1	14	14	14
	f_2	14	14	14
	f_3	14	12	12
S3	$S_3(x)$	15	14	14
	f_0	14	12	14
	f_1	13	15	15
	f_2	14	14	14
	f_3	14	10	14
S4	$S_4(x)$	14	14	14
	f_0	10	12	12
	f_1	14	14	14
	f_2	12	14	12
	f_3	14	14	14

Appendix B

Table (A). Differential Properties of the 16-Bit Block Cipher	
# of Rounds	$\max_{a\neq 0,b} D_{E_k}(a,b)$
0	16384
1	1024
2	98
3	20
4	20

Table (B). Constant c		
$wt(a)$	$wt(b)$	c
1	1	4.703125
1	2	4.359375
1	3	4.500000
2	1	4.390625
2	2	4.281250
2	3	4.828125
3	1	4.968750
3	2	4.718750
3	3	4.781250