MIBS: A New Lightweight Block Cipher*

Maryam Izadi, Babak Sadeghiyan, Seyed Saeed Sadeghian, and Hossein Arabnezhad Khanooki

{izadi_maryam,basadegh,ssadeghian,qa_had}@aut.ac.ir

Abstract. In this paper, we propose a new lightweight 64-bit block cipher, which we call MIBS, suitable for resource-constrained devices, such as low-cost RFID tags. We also study its hardware implementation efficiency, as well as its security. The hardware implementation of MIBS requires 1400 gates on 0.18 μm technology, which is less than 2000 gates limit for low-cost RFID tags. We also show MIBS is secure against differential and linear cryptanalysis.

Keywords: Block Cipher, Lightweight, Low-cost RFID Tags, Resource-Constrained Devices.

1 Introduction

Radio frequency identification (RFID) is a technology for automated identification of objects and people. Although, this technology appeared quite a long time ago, it is recently used in wide range of applications due to technical improvements and dramatic cost decrease. There are security and privacy challenges concerning this technology. Cryptographic solutions require high computing resources and come with extra costs. Development of hardware efficient security primitive for resource-constrained devices such as low-cost RFID tags is a challenging task that recently is being more dealt with[1]. Gate constraints for security of low-cost tags are about 200-2000 gates, that is less than what is necessary for standard cryptographic primitives, so existing cryptographic algorithms can be hardly implemented under such resource constraint. In this paper, we propose a new light-weight block cipher to satisfy this requirement, and at the same time, it has the necessary security. In our design we have adopted several components which are already presented in other ciphers. The paper is organized as follows. Related works are described in section 2. In section 3, we present MIBS block cipher. In section 4, the design rationale of cipher is discussed. In section 5 we analyse the security of MIBS. The study of its hardware efficiency follows in section 6. We give a conclusion in section 7.

2 Related Works

In recent years, the lightweight cryptography for RFID tags has attracted much attention. Feldhofer et al. [2] have presented a hardware implementation of

* This project has been supported in part by Iran Telecommunication Research Center.

J.A. Garay, A. Miyaji, and A. Otsuka (Eds.): CANS 2009, LNCS 5888, pp. 334–348, 2009.

Advanced Encryption Standard (AES), with a gate count of 3595. Poschmann et al. [3] designed a lightweight variant of the Data Encryption Standard (DES) called DESL, which makes use of only one S-box mapping and can therefore be more compact than DES. Their implementation fits in 1848 gates. DESXL is an another version of DES that strengthened DESL with a key size of 184 bits and a hardware size of approximately 2168 gates [4]. PRESENT [5] is a 64-bit block cipher with a key length of 80 or 128 and consists of 31 rounds which has reasonable layout size for constrained environments such as low-cost RFID tags. Hardware implementation of PRESENT requires 1570 gates. But recently, Rolfes et al. [6] present a serialized architecture of the PRESENT that requires only 1000 gates. Other compact block ciphers like mCRYPTON [7], HIGHT [8], SEA [9], and PUFFIN [10] are also proposed, but they require more area to implement than PRESENT implementation. The very compact block cipher which we proposed has a reasonable area complexity. MIBS is a 32 rounds Feistel cipher with a block length of 64-bit, where two key lengths of 64-bit and 80-bit are supported.

3 MIBS Block Cipher

MIBS uses a Feistel structure with data block length of 64-bit and key lengths of 64-bit or 80-bit and consists of 32 rounds. The round structure is shown in Fig. 1. For applications that require moderate security levels, such as low-cost RFID tags, 64-bit security is adequate. In practice, there is a tradeoff between hardware efficiency and security. The F-function, depicted in Fig. 2, operates on half a block (32 bits), representing it into eight nibbles, and it consists of four stages: key addition, non-linear substitution layer, linear mixing layer, and nibble-wise permutation.

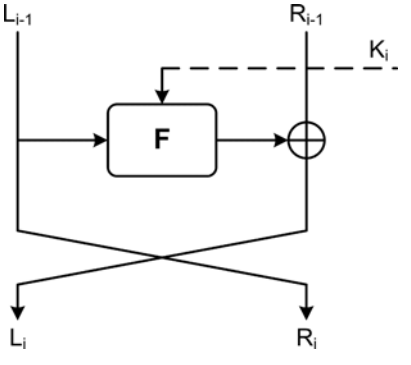

Fig. 1. Encrypt round of MIBS

Table 1. S-box mapping[7]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S	4	15	3	8	13	10	12	0	11	5	7	14	2	6	1	9

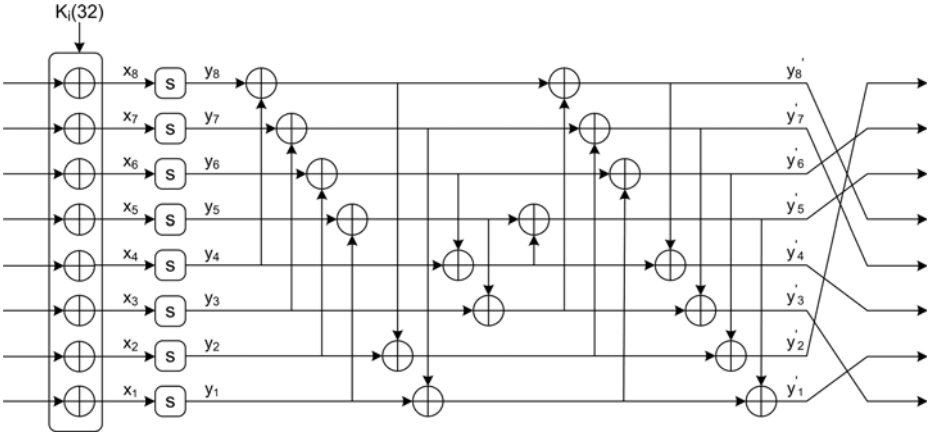

Fig. 2. The i-th round function of MIBS

Key addition. Current state $s_{31}, s_{30}, ..., s_0$, which is input to the F-function, is combined with a round subkey $k^i = k^i_{31}, k^i_{30}, \ldots, k^i_0 for 1 \leq i \leq 32$, using a bit-wise XOR operation. Since XOR is well-suited to hardware implementation, all subkeys are bitwise XORed with data before substitution layer.

$$s_j = s_j \oplus k^i_j, \text{for} 0 \leq j \leq 31$$

Substitution layer S. After adding subkey, the block is divided into eight nibbles $x_8, x_7, ..., x_1$, before processing by the S-boxes. The 4×4 S-box used in our cipher is the same as the first S-box used in mCRYPTON and is shown in Table 1. The non-linear layer is composed of eight identical 4×4 S-boxes, so in this transformation nibble-wise substitution is applied.

$$S : F_2^4 \rightarrow F_2^4 : x_i \rightarrow y_i = s\left(x_i\right), \text{for} 1 \leq i \leq 8$$

Mixing layer M. The linear transformation mixes eight nibbles as follows:

$$M : (GF(2)^4)^8 \rightarrow (GF(2)^4)^8, (y_8, y_7, \ldots, y_1) \rightarrow (y'_8, y'_7, \ldots, y'_1) \Leftrightarrow$$

Table 2. Permutation mapping

	1	2	3	4	5	6	7	8
P	2	8	1	3	6	7	4	5

$$
\begin{aligned}
y'_1 &= y_2 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_6 \oplus y_7 \\
y'_2 &= y_1 \oplus y_3 \oplus y_4 \oplus y_6 \oplus y_7 \oplus y_8 \\
y'_3 &= y_1 \oplus y_2 \oplus y_4 \oplus y_5 \oplus y_7 \oplus y_8 \\
y'_4 &= y_1 \oplus y_2 \oplus y_3 \oplus y_5 \oplus y_6 \oplus y_8 \\
y'_5 &= y_1 \oplus y_2 \oplus y_4 \oplus y_5 \oplus y_6 \\
y'_6 &= y_1 \oplus y_2 \oplus y_3 \oplus y_6 \oplus y_7 \\
y'_7 &= y_2 \oplus y_3 \oplus y_4 \oplus y_7 \oplus y_8 \\
y'_8 &= y_1 \oplus y_3 \oplus y_4 \oplus y_5 \oplus y_8
\end{aligned}
$$

Permutation layer P. Finally, the eight nibble outputs from the mixing layer are arranged according to Table 2. Each nibble is moved to a new position by P.

Key schedule for 64-bit key. The design principle of MIBS key schedule is adopted from the design principle of PRESENT key schedule. Our key schedule, generates 32-bit round key k^i, for $0 \leq i \leq 31$, from 64-bit user key K (represented as $k_{63}, k_{62}, ..., k_0$). We denote the key state of the i-th round as $state^i$. The key state for each round is updated as follows.

$$
\begin{aligned}
&state^0 = \text{user-key} \\
&state^i = state^i \ggg 15 \\
&state^i = \text{S-box}(state^i_{[63:60]})||state^i_{[59:0]} \\
&state^i = state^i_{[63:16]}||state^i_{[15:11]} \oplus \text{Round-Counter}||state^i_{[10:0]} \\
&k^i = state^i_{[63:32]}
\end{aligned}
$$

where $\ggg$ means rotation to right, $[i : j]$ indicates the i-th to the j-th bits are involved in the operation, and $||$ denotes concatenation. Also we use the same S-box as in the F-function. The round key k^i is the 32 left most bits of the current state.

Key schedule for 80-bit key. The key K is first initialized with the user key, and updates as follows.

$$
\begin{aligned}
&state^0 = \text{user-key} \\
&state^i = state^i \ggg 19 \\
&state^i = \text{S-box}(state^i_{[79:76]})||\text{S-box}(state^i_{[75:72]})||state^i_{[71:0]} \\
&state^i = state^i_{[79:19]}||state^i_{[18:14]} \oplus \text{Round-Counter}||state^i_{[13:0]} \\
&k^i = state^i_{[79:48]}
\end{aligned}
$$

After that, the round key k^i is the 32 left most bits of the key state. Test vectors for MIBS with 64 bit and 80 bit key are provided in the Appendix I.

4 Design Rationale

4.1 The Cipher Structure

MIBS is based on Feistel structure with an SPN round function. A large proportion of block ciphers have used this scheme since the US Federal Government adopted the DES. Moreover, DES has endured various attacks for over 20 years, even though its round function is very simple. Since Feistel construction operates on half of the block length in each iteration, therefore the size of code or circuitry required to implement it is nearly halved. Thus we use Feistel network as an overall structure with the purpose of minimizing computational resources, which certainly is one of the most important considerations in hardware design for tiny ubiquitous devices.

4.2 Round Function

For round function we selected the Substitution-Permutation Network (SPN). The SPN structure is directly based on the concepts of confusion and diffusion. The confusion component is a nonlinear substitution and the diffusion component is a linear mixing which is used for diffusing the cryptographic characteristics of substitution layer.

The substitution layer. The most important objective in designing a block cipher targeted to embedded applications such as RFID tags, is to achieve low complexity in hardware while providing sufficient security. Consequently an appropriate substitution layer of such a block cipher should meets the above balance. Although, large S-boxes can achieve better security but even in software, large S-boxes require high storage cost and they are far worse in hardware. On the other hand, too small S-boxes can hardly achieve suitable security. We observed the gate count increases exponentially with the size of S-box. As a result, we decided to use 4×4 S-boxes with regard to hardware efficiency and at the same time adequate security. Also existing lightweight block ciphers like PRESENT, and mCRYPTON have used 4×4 S-boxes too. The S-box used in MIBS block cipher is the same as the S0 mapping applied in mCRYPTON [7].

The linear transformation. In order to construct a fast and strong block cipher, we design a round function that is secure against differential and linear cryptanalysis and yield small values for the maximum differential and linear probabilities p, q. Kanda et al. [11], proposed a search algorithm for constructing an optimal linear transformation layer by using the matrix representation in order to minimize probabilities p, q as much as possible. They determined an optimal linear transformation layer among many candidates which has a lower computational complexity, which we used in MIBS. Additionally they showed that any linear transformation following a non-linear layer consists of 8 parallel S-boxes, can not have branch number more than 5. The branch number is the minimum number of active S-boxes in two consecutive rounds of a non-trivial

differential characteristic or non-trivial linear trail [12]. In this context, by Optimal we mean that the maximum differential and linear probabilities p, q are as small as possible. Similar linear transformations is used also in E2[13] and Camellia[14] block ciphers. The linear layer M, which we call mixing layer, is represented using only 16 nibble-wise XORs that is suitable for computational efficiency. For security against differential and linear cryptanalysis, the branch number of layer M is optimal. Consequently, the mixing layer piles up the number of active S-boxes every two rounds to minimize the maximum differential and linear probabilities.

5 Security Analysis

5.1 Differential and Linear Cryptanalysis

preliminaries. Two well-known attacks applicable against block ciphers are differential cryptanalysis, introduced by Biham and Shamir [15], and linear cryptanalysis proposed by Matsui[16]. Because of wide applicability of both attacks to numerous block ciphers, resistant against them should be considered in the design of block ciphers. The complexity of each attack is defined by the number of active S-boxes involved and their differential characteristic or linear approximation probabilities. Kanda et al. [17] show the minimum number of active S-boxes in differential and linear attacks for Feistel ciphers with SPN round function which is presented below.

Definition 1. *For any given* $\Delta_x, \Delta_y, \Gamma_x, \Gamma_y \in GF(2)^m$*, the differential and linear probabilities of each S-box are defined as:*

$$DP^{S_i}(\Delta_x \rightarrow \Delta_y) = \frac{\#\{x \in GF(2)^m | S_i(x) \oplus S_i(x \oplus \Delta_x) = \Delta_y\}}{2^m}$$

$$LP^{S_i}(\Gamma_y \rightarrow \Gamma_x) = (2 \times \frac{\#\{x \in GF(2)^m | x\Gamma_x = S_i(\mathrm{x})\Gamma_y\}}{2^m} - 1)^2$$

Where $x \cdot \Gamma_x$ *, denotes the parity (0 or 1) of bitwise product of* x *and* Γ_x*.*

Definition 2. *The maximum differential and linear probabilities of S-boxes are defined as:*

$$p_s = \max_i \max_{\Delta x \neq 0, \Delta y} DP^{S_i}(\Delta x \rightarrow \Delta y)$$
$$q_s = \max_i \max_{\Gamma x, \Gamma y \neq 0} LP^{S_i}(\Gamma y \rightarrow \Gamma x)$$

Definition 3 ([11]). *A differential active S-box is defined as an S-box given a non-zero input difference, while a linear active S-box is defined as an S-box given a non-zero output mask value.*

As we mentioned earlier, the security against differential and linear cryptanalysis is evaluated using the branch number, and branch number is defined as follow[12].

Definition 4. *The differential branch number B_d is defined as:*

$$B_d = \min_{\Delta x \neq 0} (H_w(\Delta x) + H_w(\theta(\Delta x)))$$

where Δx is an input difference into the diffusion layer and $\theta(\Delta x)$ is an output difference from the layer. H_w denotes the number of non-zero nibbles as defined in [17].

In our case that the mixing transformation is bijective, the differential branch number B_d , and linear branch number B_l are identical ($B = B_d = B_l$).

Definition 5. *The minimum number of differential active S-boxes of the r-round Feistel cipher with SPN round function, is defined as:*

$$D^{(r)} = \min_{(\Delta x^{(0)}, \Delta x^{(1)}, \ldots, \Delta x^{(r+1)}) \neq (0,0,\ldots,0)} \sum_{i=1}^{r} H_w(\Delta x^{(i)})$$

where $H_w(\Delta x^{(i)})$ is the number of the ith-round differential active S-boxes.

Theorem 1 ([17]). *The minimum number of differential active s-boxes $D^{(4r)}$ for $4r$ round Feistel ciphers with SPN round function satisfies $D^{(4r)} \geq r \times B + \lfloor r/2 \rfloor$.*

Theoretical Analysis. Theorem 1 also holds for $L^{(r)}$, the number of non-zero nibbles in linear approximation of round r, because both non-linear and linear layers are bijective.

The maximum differential and linear probabilities of the S-boxes are $p_s = q_s = 2^{-2}$, and the branch number of the linear transformation is 5. According to theorem 1, the lower bound of the number of active S-boxes with respect to linear and differential cryptanalysis is as follows: $D^{(32)} \geq 8 \times 5 + 4 \Rightarrow D^{(32)} \geq 44$.

Therefore, the upper bound of maximum differential characteristic probability is $(2^{-2})^{44} = 2^{-88}$, and the bias of linear approximation according to pilling-up lemma, is $2^{43} \times (2^{-2})^{44} = 2^{-45}$. As a result our suggested number of rounds is a conservative choice and we have a fair amount of security margin.

Experimental Analysis

Differential cryptanalysis. In the previous section we presented the theoretical bound for Differential Cryptanalysis. Here we present the best non-trivial 4-round differential characteristic we have found. Table 3 illustrates this 4-round characteristic. The left and right columns represent the non-zero nibbles for each round. This characteristic has the least number of active S-boxes i.e 6, and results in the 4-round characteristic probability of 2^{-15}. If we assume it as an iterative characteristic, we can deduce that the 32-round characteristic should not have probability better than $(2^{-15})^8 = 2^{-120}$.

Table 3. 4 rounds differential characteristic

Round Number	Left	Right	Probability
Input	10000000	10001110	
1	00100001	10000000	1/8
2	01000010	00100001	1/16
3	00000100	01000010	1/64
4	00100101	00000100	1/4
		active S-boxes = 6	Total = 2^{-15}

Table 4. 4 rounds linear approximation

Round Number	Left	Right	Number of Active S-boxes	Bias
1	00000001	00000100	5	2^{-6}
2	00000100	11011101	1	2^{-2}
3	11011101	00000000	0	1
4	00000000	11011101	1	2^{-2}
			Total = 7	Total = 2^{-8}

Linear cryptanalysis. Here we show the best linear approximation we have found. The best linear approximation for 4-round MIBS is illustrated in Table 4. The left and right columns represent the Non-zero nibbles for input of each round.

It has 7 active S-boxes with the best bias in each S-box which is 2^{-2} ,that results in the 4-round approximation with bias $(2^{-2})^7 \times 2^6 = 2^{-8}$. So it yields that the bias for 32-round approximation is at least $(2^{-8})^8 \times 2^7 = 2^{-57}$, which requires $\lambda.2^{114}$ known plaintext, where λ is a small factor that is used for better success probability.

5.2 Multiple Linear Cryptanalysis

The idea of using multiple approximation in linear cryptanalysis is first introduced by Kaliski and Robshaw [18], their method has the restriction that only same bits of key can be used in approximations. In 2004 Biryukov et al. [19] introduced the general statistical framework which does not possess that restriction. As a result of their method the data complexity of attack becomes proportional to the capacity of approximations. Time complexity of attack is equal to the time for encrypting the known plaintexts and for each encryption, updating the counters for the approximations. Multidimentional approximations without the assumption of independence are introduced by Hermelin and Nyberg [20], which allows us to build 2^m approximations from m independent approximations. We have found several approximations with good bias and several can be obtained by changing the first round input mask and last round output mask. By taking that into account we can use 16 independent approximations and build 2^{16} approximations, so at the best case that all of the combined approximations have

the maximum bias we may be able to reduce the data complexity by the order of 2^{16} at the cost of updating 2^{16} counters for each known plaintext. So the data complexity of linear cryptanalysis can be reduced to $\lambda.2^{98}$.

5.3 Other Variants of Linear Cryptanalysis

Differential-Linear is a method for connecting a differential characteristic to linear approximation, which is introduced by Langford and Hellman [21] and later enhanced for probabilistic differential characteristic by Dunkleman and Biham [22]. It is useful when we have differential and linear characteristics with high probability for small number of rounds. However as we dont have such characteristics, this attack is not applicable to the full rounds. Non-linear cryptanalysis [23] is not applicable to MIBS since it is usually useful when we have large s-boxes,and is used in outer rounds of the cipher. So the improvement by this attack does not pose a threat to MIBS because of large security margin. Bi-Linear cryptanalysis[24] which is proposed for feistel schemes is not a large improvement to the attack so this is also not a practical attack against MIBS.

5.4 Algebraic Attack

Algebraic attack is a method for the cryptanalysis of ciphers, which was first presented by Courtois and Pieprzyk [25] to analyze AES. The attack aims to recover the secret key through solving an overdefined system of multivariate algebraic equations. A block cipher, which consists small S-boxes, may be represented as many equations with small number of variables. By solving these multivariate equations the key of the block cipher may be found, but the problem of solving a system of multivariate quadratic equations is in general NP-hard. Several methods for solving such systems of equations has been proposed for the special cases of overdefined and sparse systems[25,26], although some flaws in all such techniques are claimed in [27,28]. Anyway, any 4×4 bit S-box can be represented as 21 quadratic equations of 8 input/output bit variables over GF(2) [25]. MIBS-64 consists of $n = (32 \times 8) + 32 = 288$ S-boxes, as there are 8 S-boxes in each round of the 32-round cipher, and one S-box in each round of key scheduling. Thus, the cipher can be described with 6048 ($= 288 \times 21$) quadratic equations of 2304 ($= 288 \times 8$) variables. MIBS-80 has 32 S-boxes more than MIBS-64, so the number of quadratic equations is 6720 with 2560 variables.

According to [25], an estimation of the complexity of XSL attack on a block cipher can be calculated with work factor. For MIBS-64, W.F. is accordingly estimated as follows:

$$\begin{aligned} WF &\approx \Gamma^{\omega} \cdot \left((\text{Block size}) \cdot (\text{Number of rounds})^2\right)^{\omega \lceil \frac{t}{r} \rceil} \\ &= (2^6)^{2.37} \cdot \left((64) \cdot (\frac{32}{2})^2\right)^{2.37 \lceil \frac{37}{21} \rceil} \\ &= 2^{80.58} \end{aligned}$$

Which is greater than 2^{64} operations needed for exhaustive search, making the attack impractical.

5.5 Related Key Attack

Slide attack [29] and related-key [30] attack are a form of cryptanalysis which use some weakness of key schedule. The attacker can observe the operation of a cipher under several different keys whose values are initially unknown, but where some mathematical relationship connecting the keys is known to the attacker. The design rationale of the key schedule of MIBS is similar to the key schedule of PRESENT. Since key schedule uses the round-dependent counter and a non-linear operation to mix the contents of the key register K, it is secure against these attacks.

6 Hardware Implementation

MIBS block ciphers is designed for very efficient hardware implementations, and each component is carefully constructed with hardware implementations in mind. In order to check hardware complexity, MIBS was implemented in a standard cell library based on TSMC 0.18μm CMOS technology. The block cipher is described in Verilog and simulated using ModelSim SE PLUS 6.2b. The synthesis is only done for encryption using typical transistors with the aim of area optimization by LEONARDO SPECTRUM 2005a.82. The data path of MIBS is depicted in Fig. 3. Each round consists of key addition, substitution layer, mixing layer, permutation layer, and right data addition. The substitution layer is composed of eight 4×4 S-boxes which are used in parallel. 4×4 S-box is implemented with simple combinational logic of 4-bit Boolean function. The key addition, and the

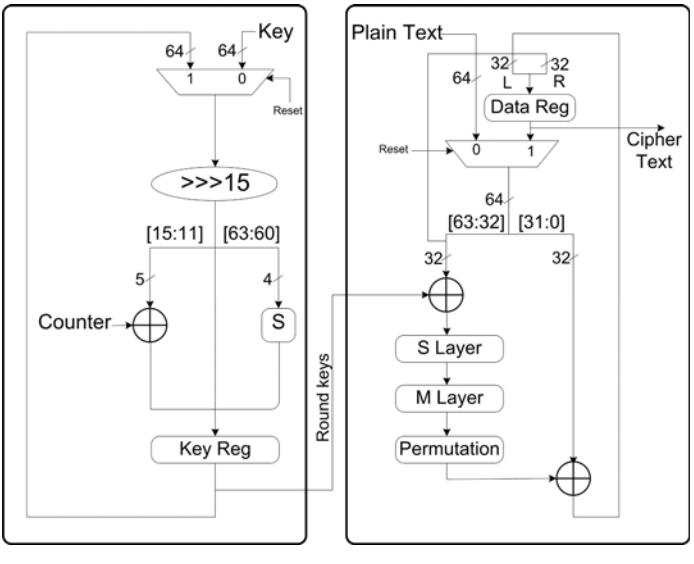

Fig. 3. Data path of MIBS-64

Table 5. Hardware complexity of MIBS-64

Module (Round Function)	GE	Module (Key Schedule)	GE
Data Register	384.68	Key Register	384.68
Substitution Layer	192	S-box	24
Key Xor	85.44	Right Rotation	0
Mixing Layer	170.84	Counter Xor	13.35
Permutation Layer	0	Total Key Schedule	422.03
Right Data Xor	85.44		
Total Round Function	918.4		
Control Unit	46		
Other Logics	8.74		
Total	1395.17		

Table 6. Hardware complexity of MIBS-80

Module (Round Function)	GE	Module (Key Schedule)	GE
Data Register	384.68	Key Register	484.46
Substitution Layer	192	S-box	48
Key Xor	85.44	Right Rotation	0
Mixing Layer	170.84	Counter Xor	13.35
Permutation Layer	0	Total Key Schedule	545.81
Right Data Xor	85.44		
Total Round Function	918.4		
Control Unit	46		
Other Logics	19.35		
Total	1529.56		

Table 7. Hardware complexity comparison of lightweight ciphers

Block ciphers	Block size	Key size	Cycles per block	Logic process	Area (GE)	Throughput at 100 KHZ(Kbps)
MIBS-64	64	64	32	0.18 μm	1396	200
PRESENT-80[5]	64	80	32	0.18 μm	1570	200
PRESENT-80[6]	64	80	563	0.18 μm	1075	11.4
AES-128[2]	128	128	1032	0.35 μm	3400	12.4
mCRYPTON[7]	64	64	13	0.13 μm	2420	492.3
HIGHT[8]	64	128	34	0.25 μm	3048	188.2
PUFFIN[10]	64	128	-	0.18 μm	2577	194
DESL[3]	64	64	144	0.18 μm	1848	44.4
DESXL[4]	64	184	144	0.18 μm	2168	44.4

right data addition are implemented as bit-wise XORs, and the mixing layer is implemented as nibble-wise XORs. The permutation layer is a simple wiring and does not have extra gates. The round keys used for each round function can be generated on-the-fly and, hence, there is no need to store all the round keys.

The implemented MIBS requires 32 clock cycles to encrypt a 64 bit plain text with 64 bit key (a single round per clock cycle), which result in throughput of 200 kilobit per second considering 100 KHz clock. MIBS implementation requires 1396 gates (2-input NAND gates). Table 5 shows the detailed gate counts of each component. The estimated area for MIBS with 80-bit key is about 1530 gates which is illustrated in table 6. A comparison for the hardware efficiency of MIBS and other lightweight block ciphers is shown in Table 7.

7 Conclusion

In this paper, we have presented a new lightweight block cipher MIBS with a 64-bit block length and 64/80-bit key lengths. Our goal in the design of MIBS was to provide security for resource-constrained applications, such as low-cost RFID tags, while having a lower hardware complexity in comparison with other compact block ciphers. MIBS is based on Feistel structure with SPN round function. We use Feistel network as an overall structure with the purpose of minimizing computational resources which is one of the important considerations in hardware design for tiny ubiquitous devices. For round function we selected the Substitution-Permutation Network. We use 4×4 S-boxes with regard to hardware efficiency and at the same time adequate security. The diffusion layer, which we named mixing layer M, is composed of 16 nibble-wise XORs, while for security against differential and linear cryptanalysis, its branch number is optimal. The hardware implementation of MIBS-64 requires 1400 gates on 0.18 μm technology, which is less than 2000 gates limit for low-cost RFID tags. We also studied the security of MIBS against several known attacks, where it showed adequate security margins. MIBS is a secure block cipher, while having a lower gate counts than PRESENT implemented in [5]. Although serialized implementation of PRESENT with lower gate counts is reported in [6], but we have not yet designed a serialized implementation for MIBS. Such an implementation remains as a future work.

References

1. Calmels, B., Canard, S., Girault, M., Sibert, H.: Low-cost cryptography for privacy in RFID systems. In: Domingo-Ferrer, J., Posegga, J., Schreckling, D. (eds.) CARDIS 2006. LNCS, vol. 3928, pp. 237–251. Springer, Heidelberg (2006)
2. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong authentication for RFID systems using the AES algorithm. In: Joye, M., Quisquater, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 85–140. Springer, Heidelberg (2004)
3. Leander, G., Paar, C., Poschmann, A., Schramm, K.: New lightweight DES variants. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 196–210. Springer, Heidelberg (2007)
4. Poschmann, A., Leander, G., Schramm, K., Paar, C.: A family of light-weight block ciphers based on DES suited for RFID applications. In: Proceedings of FSE (2007)

5. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
6. Rolfes, C., Poschmann, A., Leander, G., Paar, C.: Ultra-Lightweight Implementations for Smart Devices-Security for 1000 Gate Equivalents. In: Grimaud, G., Standaert, F.-X. (eds.) CARDIS 2008. LNCS, vol. 5189, pp. 89–103. Springer, Heidelberg (2008)
7. Lim, C.H., Korkishko, T.: mCrypton – A lightweight block cipher for security of low-cost RFID tags and sensors. In: Song, J.-S., Kwon, T., Yung, M. (eds.) WISA 2005. LNCS, vol. 3786, pp. 243–258. Springer, Heidelberg (2006)
8. Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B.S., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., Chee, S.: HIGHT: A new block cipher suitable for low-resource device. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer, Heidelberg (2006)
9. Standaert, F.X., Piret, G., Gershenfeld, N., Quisquater, J.J.: SEA: A scalable encryption algorithm for small embedded applications. In: Domingo-Ferrer, J., Posegga, J., Schreckling, D. (eds.) CARDIS 2006. LNCS, vol. 3928, pp. 222–236. Springer, Heidelberg (2006)
10. Cheng, H., Heys, H., Wang, C.: PUFFIN: A Novel Compact Block Cipher Targeted to Embedded Digital Systems. In: Proceedings of the 2008 11th EUROMICRO Conference on Digital System Design Architectures, Methods and Tools, pp. 383–390. IEEE Computer Society, Washington (2008)
11. Kanda, M., Takashima, Y., Matsumoto, T., Aoki, K., Ohta, K.: A strategy for constructing fast round functions with practical security against differential and linear cryptanalysis. In: Tavares, S., Meijer, H. (eds.) SAC 1998. LNCS, vol. 1556, p. 264. Springer, Heidelberg (1999)
12. Rijmen, V., Daemen, J., Preneel, B., Bosselaers, A., Win, E.D.: The cipher SHARK. In: Gollmann, D. (ed.) FSE 1996. LNCS, vol. 1039, pp. 99–111. Springer, Heidelberg (1996)
13. Kanda, M., Moriai, S., Aoki, K., Ueda, H., Takashima, Y., Ohta, K., Matsumoto, T.: E2–A New 128-Bit Block Cipher. IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences 83(1), 48–59 (2000)
14. Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., Tokita, T.: Camellia: A 128-bit block cipher suitable for multiple platforms design andanalysis. In: Stinson, D.R., Tavares, S. (eds.) SAC 2000. LNCS, vol. 2012, pp. 39–56. Springer, Heidelberg (2001)
15. Biham, E., Shamir, A.: Differential cryptanalysis of the full 16-round DES. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 487–496. Springer, Heidelberg (1993)
16. Matsui, M.: Linear cryptanalysis method for DES cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)
17. Kanda, M.: Practical security evaluation against differential and linear cryptanalyses for feistel ciphers with SPN round function. In: Stinson, D.R., Tavares, S. (eds.) SAC 2000. LNCS, vol. 2012, pp. 324–338. Springer, Heidelberg (2001)
18. Kaliski Jr., B.S., Robshaw, M.J.B.: Linear cryptanalysis using multiple approximations. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 26–39. Springer, Heidelberg (1994)
19. Biryukov, A., Canniere, C.D., Quisquater, M.: On multiple linear approximations. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 1–22. Springer, Heidelberg (2004)

20. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional linear cryptanalysis of reduced round serpent. In: Mu, Y., Susilo, W., Seberry, J. (eds.) ACISP 2008. LNCS, vol. 5107, pp. 203–215. Springer, Heidelberg (2008)
21. Langford, S.K., Hellman, M.E.: Differential-linear cryptanalysis. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 17–25. Springer, Heidelberg (1994)
22. Biham, E., Dunkelman, O., Keller, N.: Enhancing differential-linear cryptanalysis. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 587–592. Springer, Heidelberg (2002)
23. Knudsen, L.R., Robshaw, M.J.B.: Non-linear approximations in linear cryptanalysis. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 224–236. Springer, Heidelberg (1996)
24. Courtois, N.T.: Feistel schemes and bi-linear cryptanalysis. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 23–40. Springer, Heidelberg (2004)
25. Courtois, N.T., Pieprzyk, J.: Cryptanalysis of block ciphers with overdefined systems of equations. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 267–287. Springer, Heidelberg (2002)
26. Courtois, N., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer, Heidelberg (2000)
27. Cid, C., Leurent, G.: An analysis of the xsl algorithm. In: Roy, B. (ed.) ASIACRYPT 2005. LNCS, vol. 3788, pp. 333–352. Springer, Heidelberg (2005)
28. Diem, C.: The xl-algorithm and a conjecture from commutative algebra. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 323–337. Springer, Heidelberg (2004)
29. Biryukov, A., Wagner, D.: Slide attacks. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 245–259. Springer, Heidelberg (1999)
30. Biham, E.: New types of cryptanalytic attacks using related keys. Journal of Cryptology 7(4), 229–246 (1994)

Appendix I

Test vectors of MIBS for each key lenght are given here.The data are expressed in hexadecimal form.

Table 8. Test vectors for 64 bit key

Plaintext	Key	Ciphertext
00000000 00000000	00000000 00000000	6D1D3722 E19613D2
00000000 00000001	00000000 00000000	D79C5610 0851488A
00000000 00000000	FFFFFFFF FFFFFFFF	E538379F 99337F4A
00000000 00000001	FFFFFFFF FFFFFFFF	EF0840A9 4FCC2EAF
FFFFFFFF FFFFFFFF	00000000 00000000	66F21F5B 1F96D626
FFFFFFFF FFFFFFFE	00000000 00000000	5D86E9E2 96B4527F
FFFFFFFF FFFFFFFF	FFFFFFFF FFFFFFFF	595263B9 3FFE6E18
FFFFFFFF FFFFFFFE	FFFFFFFF FFFFFFFF	598CE962 22A34BDE

Table 9. Test vectors for 80 bit key

Plaintext	Key	Ciphertext
00000000 00000000	00000000 00000000 0000	F575004B 83ABA59F
00000000 00000001	00000000 00000000 0000	C80A965F 0969BB70
00000000 00000000	FFFFFFFF FFFFFFFF FFFF	F2144A89 F33C2AF0
00000000 00000001	FFFFFFFF FFFFFFFF FFFF	7A443766 74739625
FFFFFFFF FFFFFFFF	00000000 00000000 0000	DE2860FD B436725E
FFFFFFFF FFFFFFFE	00000000 00000000 0000	4617D4EB 1CE9E088
FFFFFFFF FFFFFFFF	FFFFFFFF FFFFFFFF FFFF	3185C8A3 5B51EB23
FFFFFFFF FFFFFFFE	FFFFFFFF FFFFFFFF FFFF	FC835FF 013970A5