Journal of Universal Computer Science, vol. 15, no. 5 (2009), 1007-1022
submitted: 1/8/09, accepted: 28/2/09, appeared: 1/3/09 © J.UCS

Security Analysis of the Full-Round CHESS-64 Cipher Suitable for Pervasive Computing Environments

Changhoon Lee
(Hanshin University, Osan, Korea
chlee@hs.ac.kr)

Jongsung Kim[1]
(Korea University, Seoul, Korea
joshep@cist.korea.ac.kr)

Seokhie Hong
(Korea University, Seoul, Korea
hsh@cist.korea.ac.kr)

Yang-Sun Lee
(FUMATE Co., Daejeon, Korea
yslee@fumate.com)

Abstract: Wireless networks, telecommunications, and information technologies connected devices in pervasive computing environments require a high speed encryption for providing a high security and a privacy. The CHESS-64 based on various controlled operations is designed for such applications. In this paper, however, we show that CHESS-64 doesn't have a high security level, more precisely, we present two related-key differential attacks on CHESS-64. The first attack requires about 2^{44} data and 2^{44} time complexities (recovering 20 bits of the master key) while the second attack needs about 2^{39} data and 2^{39} time complexities (recovering 6 bits of the master key). These works are the first known cryptanalytic results on CHESS-64 so far.

Key Words: Block Cipher, CHESS-64, Data-Dependent Permutation, Data-Dependent Operation, Differential Cryptanalysis, Related-Key Attack.
Category: E.3, L.4, L.7

1 Introduction

Pervasive computing environments allow users to interact with embedded computers, depending on the users' current context. These new environments raise a variety of privacy and security challenges. For example, context-sensitive services can easily leak information about a user's context, and uncertainty about a user's context might lead to wrongfully disclosed information. But, these challenges can be solved by applying cryptographic algorithms in new ways and by evaluating these algorithms in prototype applications.

[1] Corresponding author: Jongsung Kim

Table 1: Results of our related-key differential attacks on the full-round CHESS-64 and of exiting related-key differential attacks on full rounds of selected DDP-based ciphers

Block Cipher	Complexity Data / Time	Number of Rec. Key Bits	Comment
CHESS-64 (8 rounds)	2^{44}RK-CP / 2^{44}	20	This paper
	2^{39}RK-CP / 2^{39}	6	This paper
	2^{44}RK-CP / 2^{108}	128 (full)	This paper
	2^{39}RK-CP / 2^{122}	128 (full)	This paper
Cobra-H64 (10 rounds)	$2^{15.5}$RK-CP / $2^{15.5}$	23	[Lee et al. 2005a]
	$2^{15.5}$RK-CP / 2^{105}	128 (full)	[Lee et al. 2005a]
Cobra-H128 (12 rounds)	2^{44}RK-CP / 2^{44}	63	[Lee et al. 2005a]
	2^{44}RK-CP / 2^{193}	256 (full)	[Lee et al. 2005a]

RK.Differential: Related-Key Differential Attack
RK-CP: Related-Key Chosen Plaintexts, Time: Encryption units, Rec.: Recovered

Recently, for encryption applications that require a fast hardware implementation with a low cost in pervasive computing environments, data-dependant permutation (DDP) based block ciphers, namely SPECTR-H64 [Goots et al. 2003], the CIKS family (CIKS-128 [Goots et al. 2003a], CIKS-128H [Sklavos et al. 2003a]), and the Cobra family (Cob ra-S128 [Goots et al. 2003b], Cobra-H64 [Sklavos et al. 2005], Cobra-H128 [Sklavos et al. 2005]), have been proposed. In order to achieve high speeds in such applications, these ciphers usually use agile key schedules as well as simple data transformation structures. So, they are also suitable for the network applications in the case of frequent change of keys.

However, since DDPs are just a linear primitive and conserve weights of transformed bit strings, the DDP-based ciphers have potential weaknesses against cryptanalytic attacks [Lee et al. 2002, Ko et al. 2004, Ko et al. 2004a, Lee et al. 2005].

To overcome this security problem of DDPs and update some DDP-based ciphers, variable data-dependent operations (DDOs) that change arbitrarily weights of transformed binary vectors were developed, and CHESS-64 [Moldovyan et al. 2005a] was proposed as an example of the DDO-based cipher. It is a 64-bit block cipher with a 128-bit key and 8 rounds, which achieves more efficient hardware implementations than the existing DDP-based ciphers. However, until now, there have been no known attack results of the DDO-based cipher CHESS-64 yet. Even though CHESS-64 employs the DDOs having a better security than the DDPs, its simple key schedule and structural weaknesses degrade the security of the cipher, especially against related-key attacks. In this paper, we first show that related-key differential cryptanalysis can be applied to devise two key recovery attacks on the full-round CHESS-64. The first attack allows us to recover 20 bits of the master key with 2^{44} related-key chosen plaintexts and 2^{44} encryptions while the second attack recovers 6 bits of the master key with 2^{39} related-key chosen plaintexts and 2^{39} encryptions. By the exhaustive search technique for the remaining key bits, our first and second attacks are converted into full-key recovering attacks having a data complexity of 2^{44} related-key chosen plaintexts, a time complexity of 2^{108} and a data complexity of 2^{39} related-key chosen plaintexts, a time complexity of 2^{122} encryptions, respectively. These works are the first known cryptanalytic results on CHESS-64 so far. Table 1 summarizes our results and existing cryptanalytic results on some of selected DDP-based ciphers.

It seems that the related-key attack is very difficult or even infeasible to conduct in many cryptographic applications, since it would certainly be unlikely that an attacker could persuade a sender to encrypt plaintexts under related keys unknown to the attacker. However, as demonstrated in [Kelsey et al. 1996, Phan and Handschuh 2004, Razali and Phan 2006], the related-key attack is feasible in some of the current real-world applications such as the IBM 4758 cryptoprocessor, PGV-type hash functions, message authentication codes, recent authenticated encryption modes, cases of key-exchange protocols that do not guarantee key integrity, and key-update protocols that updates session keys using a known function, for example, K, $K+1$, $K+2$, etc., where K is a session key.

This paper is organized as follows; in Section 2, we briefly describe DDO-boxes, used in CHESS-64. Section 3 describes CHESS-64 and their structural properties. In Sections 4 we present related-key differential attacks on CHESS-64. Finally, we conclude in Section 5.

2 Preliminaries

In this section, we introduce notations and controlled operations which are the components of CHESS-64. The following notation is used throughout this paper. A bit index will be numbered from left to right, starting with bit 1. If $I = (i_1, i_2, \cdots, i_n)$ then i_1 is the most significant bit(msb) and i_n is the least significant bit(lsb).

- $e_{i,j}$: a binary string in which the i-th and j-th bits are one and the others are zeroes, e.g., $e_{2,3} = (0, 1, 1, 0, \cdots, 0)$.
- $\oplus$: bitwise-XOR operation
- $\lll(\ggg)$: left(right) cyclic rotation
- $Pr_{(\Psi)}(\Delta Y/\Delta X, \Delta V)$: a probability that the output difference of Ψ is ΔY when the input difference and controlling input difference of Ψ are ΔX and ΔV, respectively.

2.1 Controlled Permutations

The DDP-like operations can be performed with controlled permutation (CP) boxes, which are defined as follows: Let $C(X, V)$ be a function $C : \{0,1\}^n \times \{0,1\}^m \rightarrow$

$\{0,1\}^n$. C is called a Controlled Permutation box (CP-box), if $C(X,V)$ is a bijection for any fixed V.

Now, we describe CP-boxes, DDOs, which are denoted by $F_{n/m}$. The $F_{n/m}$ is the set of permutations on n-bit binary vectors X depending on some controlling m-bit vector V. It is constructed by using the basic building blocks $F_{2/1}$, which is defined by two specific boolean functions in three variables y_1=$f_1(x_1,x_2,v)$ and y_2=$f_2(x_1,x_2,v)$ as follows.

$$F_{2/1}(x_1,x_2,v) = \begin{cases} (x_2,x_1) & \text{if } v\text{=0} \\ (x_1,x_1\oplus x_2) & \text{if } v\text{=1} \end{cases}$$

To execute variable permutations, the $F_{n/m}$-box is generally constructed as a superposition of the operations performed on bit sets :

$$F_{n/m} = L^{V_1}\circ\pi_1\circ L^{V_2}\circ\pi_2\circ\cdots\circ\pi_{s-1}\circ L^{V_s}$$

where L is an active layer composed of $\frac{n}{2}$ $F_{2/1}$ parallel elementary boxes, $V_1,V_2,\cdots,V_s$ are controlling vectors of the active layers from 1 to $s=\frac{2m}{n}$, and $\pi_1,\pi_2,\cdots,\pi_{s-1}$ are fixed permutations (see Fig. 1).

Due to the symmetric structure $F_{n/m}$ and $F^{-1}_{n/m}$ differ only with the distribution of controlling bits over the boxes $F_{2/1}$. Thus to construct $F^{-1}_{n/m}$, it is sufficient to number the boxes $F_{2/1}$ from left to right and from bottom to top and to replace π_i by π^{-1}_{s-i}, e.g., as shown in Fig. 1-(g) and (h), $F^{V}_{32/96}$ and $F^{V'}_{32/96}$ are mutually inverse when V=(V_1,V_2,$\cdots$,V_6) and V'=(V_6,V_5,$\cdots$,V_1) where $|V_i|=16$ bits.

Similarly, $F'_{n/m}$ can be constructed by using another basic building block $F'_{2/1}$, which is defined as follows.

$$F'_{2/1}(x_1,x_2,v) = \begin{cases} (x_2\oplus 1,x_1\oplus 1) & \text{if } v\text{=0} \\ (x_1\oplus x_2\oplus 1,x_2) & \text{if } v\text{=1} \end{cases}$$

In the CHESS-64 block cipher, only $F_{32/96}$, $F^{-1}_{32/96}$, and $F'_{32/80}$ in Figs. 1 and 2 are used as a component.

3 CHESS-64

In this section, we briefly describe CHESS-64 which is designed by using new DDP-like DDOs with no other nonlinear operations. This cipher is composed of the initial transformation (IT), the round function $Crypt$, and the final transformation (FT) and its encryption procedure is performed as in Table 2.

3.1 Description of CHESS-64

CHESS-64 is a 8-round iterated block cipher with a 64-bit input and a 128-bit key. Its general structure and round function are shown in Fig. 3-(a) and -(b), respectively.

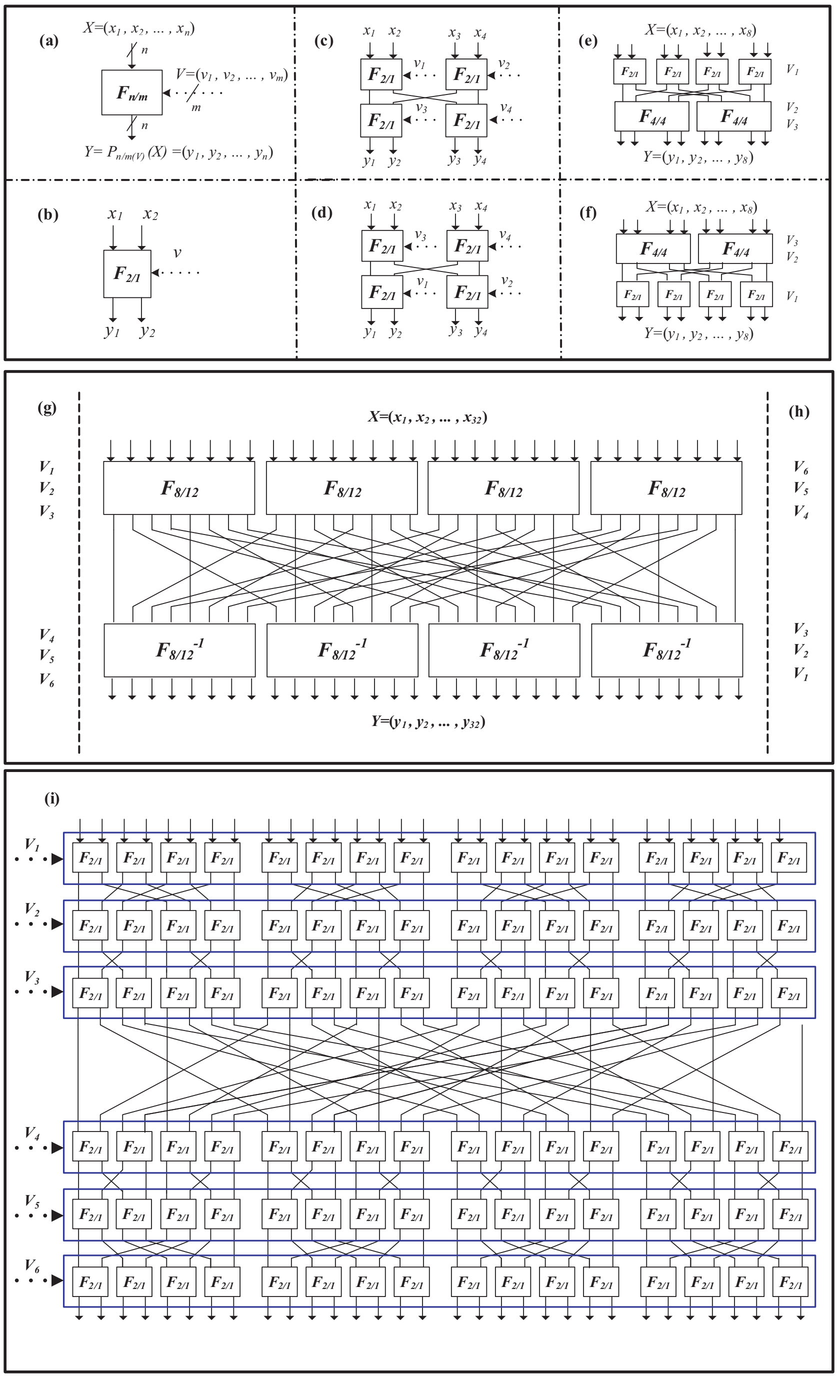

Figure 1: (a) $F_{n/m}$, (b) $F_{2/1}$, (c) $F_{4/4}$, (d) $F_{4/4}^{-1}$, (e) $F_{8/12}$, (f) $F_{8/12}^{-1}$, (g) $F_{32/96}$, (h) $F_{32/96}^{-1}$, (i) Detail of $F_{32/96}$

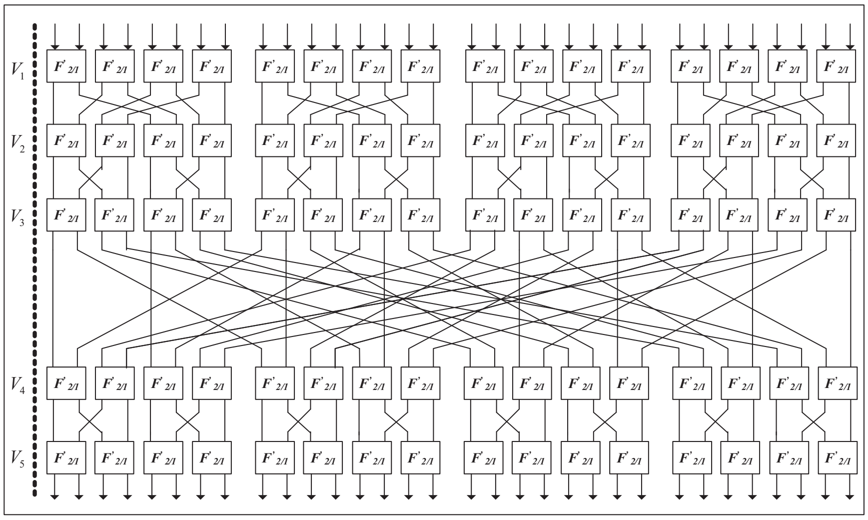

Figure 2: $F'_{32/80}$

Table 2: Encryption procedure CHESS-64

Encryption Procedure
[Step 1] An input block is divided into two subblocks P_L and P_R.
[Step 2] Perform IT : $P_L^1 = P_L \oplus RK_L^0$ and $P_R^1 = P_R \oplus RK_R^0$;
[Step 3] For $j = 2$ to r do :
$\circ$ $(P_L^j, P_R^j) := Crypt(P_L^{j-1}, P_R^{j-1}, RK^{j-1,(e)})$,
$\circ$ Swap the data subblocks : $T = P_R^j$, $P_R^j = P_L^j$, $P_L^j = T$;
[Step 4] $j = r + 1$ do : $(P_L^{r+1}, P_R^{r+1}) := Crypt(P_L^r, P_R^r, RK^{r,(e)})$;
[Step 5] Perform FT : $C_L = P_L^{r+1} \oplus RK_L^{r+1}$ and $C_R = P_R^{r+1} \oplus RK_R^{r+1}$;
[Step 6] Return the ciphertext block $C = (C_L, C_R)$.

As depicted in Fig. 3-(b) the $Crypt$ function is composed of two cyclic rotations ($\lll$ 16, $\ggg$ 7), three advanced DDO-boxes $F_{32/96}$, $F_{32/96}^{-1}$, $F'_{32/80}$, two extension boxes E, E', and an involution permutation I.

Given an input L=(l_1,$\cdots$,l_{32}), the extension E outputs V=(V_1,V_2,V_3,V_4,V_5, V_6)= (L_l,$L_l^{\ggg 6}$,$L_l^{\ggg 12}$,L_r,$L_r^{\ggg 6}$,$L_r^{\ggg 12}$) where L_l=(l_1,$\cdots$,l_{16}), L_r=(l_{17},$\cdots$,l_{32}), $|l_i|$=1 ($1 \leq i \leq 32$) and $|V_i|$=16 ($1 \leq i \leq 6$). E' forms 80-bit output vector W=(W_1,W_2, W_3,W_4,W_5) for given input Z'=(Z'_l,Z'_r) where W_1=Z'_l, W_2=$Z_l'^{\lll 5}$, W_3=$Z_l'^{\lll 10}$, W_4=Z'_r, W_5= $Z_r'^{\lll 5}$. The permutational involution I is defined by two rotations by eight bits: $I(X_1, X_2) = (X_1^{\ggg 8}, X_2^{\ggg 8})$.

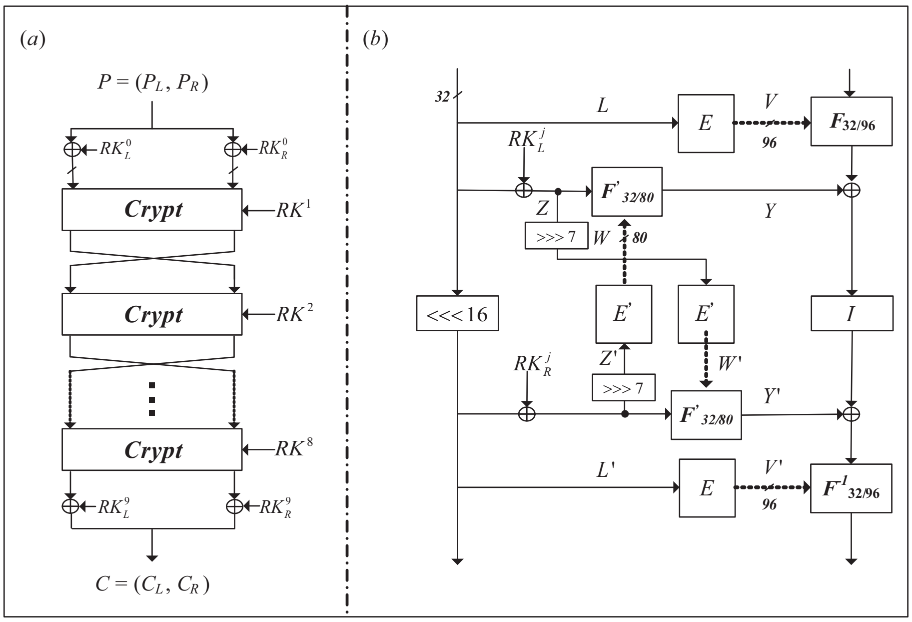

Figure 3: (a) General structure of CHESS-64, (b) $Crypt$ of CHESS-64

The key schedule of CHESS-64 is simple; as shown in Table 3, the 32-bit subkey K_i of a 128-bit secret key K=(K_1, K_2, K_3, K_4) are directly used as encryption round key $RK^j = (RK^j_L, RK^j_R)$ where RK^9 is the round key of the final transformation.

Table 3: Key schedule of CHESS-64

Round (j)	0	1	2	3	4	5	6	7	8	9
RK^j_L	K_4	K_3	K_2	K_4	K_1	K_4	K_1	K_2	K_3	K_1
RK^j_R	K_3	K_1	K_4	K_2	K_3	K_2	K_3	K_1	K_4	K_2

3.2 Properties for Components of CHESS-64

In this subsection, we describe some properties for components of $Crypt$ of CHESS-64, which allow us to construct its full-round related key differential characteristics. To begin with, we describe several basic properties of the controlled elements , which can induce the properties of components of $Crypt$.

Property 1. Let CE be a $F_{2/1}$. Then we obtain the following basic properties for $Pr_{(CE)}(\varDelta Y/\varDelta X, \varDelta V)$.

$a)$ $Pr_{(F_{2/1})}((0,0)/(0,0),0)$=1.

$b)$ $Pr_{(F_{2/1})}(\varDelta Y/\varDelta X,1)$=$2^{-2}$ for any $\varDelta X$, $\varDelta Y$.

$c)$ $Pr_{(F_{2/1})}(\varDelta Y_1/(0,1),0)$=$Pr_{(F_{2/1})}(\varDelta Y_2/\ (1,0),0)$=$Pr_{(F_{2/1})}(\varDelta Y_3/(1,1),0)$=$2^{-1}$ where $\varDelta Y_1 \in \{(0,1),(1,0)\}$, $\varDelta Y_2 \in \{(0,1),\ (1,1)\}$, $\varDelta Y_3 \in \{(1,0),(1,1)\}$.

$d)$ $Pr_{(F'_{2/1})}((0,0)/(0,0),0)$=1.

$e)$ $Pr_{(F'_{2/1})}(\varDelta Y/\varDelta X,1)$=$2^{-2}$ for any $\varDelta X$, $\varDelta Y$.

$f)$ $Pr_{(F'_{2/1})}(\varDelta Y_1/(0,1),0)$=$Pr_{(F'_{2/1})}(\varDelta Y_2/\ (1,0),0)$=$Pr_{(F'_{2/1})}(\varDelta Y_3/(1,1),0)$=$2^{-1}$ where $\varDelta Y_1 \in \{(1,0),(1,1)\}$, $\varDelta Y_2 \in \{(0,1),\ (1,0)\}$, $\varDelta Y_3 \in \{(0,1),(1,1)\}$.

The above properties are also extended into the following properties.

Property 2. Let CE be a $F_{n/m}$, or a $F^{-1}_{n/m}$. Then we obtain the following extended properties for $Pr_{(CE)}(\varDelta Y/\varDelta X, \varDelta V)$.

$a)$ $Pr_{(F_{n/m})}((0)/(0),e_i)$=$Pr_{(F^{-1}_{n/m})}((0)/(0),e_i)$ =2^{-2}.

Property 3. Let $F_{n/m(V)}(X) \oplus F_{n/m(V)}(X \oplus e_i) = e_j$ for some i and j where $1 \leq i \leq n$ and $1 \leq j \leq m$. Then we have the following property;

$a)$ If $(n,m) \in \{(2,1),(4,4),(8,12),(32,96)\}$ and $i = n$ then the exact one difference route from e_i to e_j via $F_{2/1}$-boxes is fixed. It also holds in $F^{-1}_{n/m}$-box.

For example, consider $i = n = 8$ and $j = 2$ in the $Property$ 3. Then, we can exactly know the 3 bits of controlling vectors (0,0,1) corresponding to three elements $F_{2/1}$-boxes of $F_{8/12}$-box with probability 1. See Fig. 4. In Fig. 4, the bold line denotes the possible difference route when the input, output, and control vector differences of $F_{8/12}$ are fixed as $\varDelta X = (0,...,0,1)$, $\varDelta Y = (0,1,0,...,0)$, and 0, respectively. This is a essential idea of our key recovery attack on CHESS-64.

4 Related-Key Differential Attack on CHESS-64

In this section, we show how to construct full-round related-key differential characteristics of CHESS-64 by using the properties presented in the previous section, and then present akey recovery attack on the full-round CHESS-64.

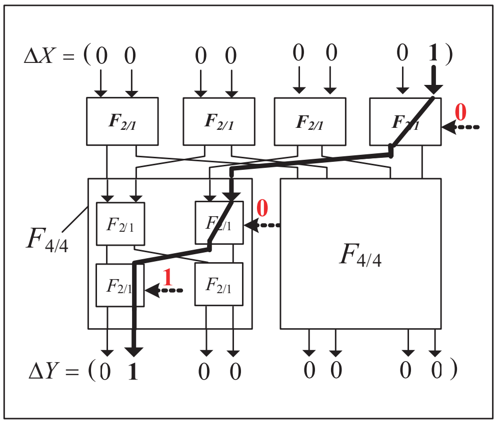

Figure 4: An example of the difference route when the input, output, and control vector differences of $F_{8/12}$ are fixed as $\Delta X = (0, ..., 0, 1)$, $\Delta Y = (0, 1, 0, ..., 0)$, and 0, respectively

4.1 Related-Key Differential Characteristics of CHESS-64

We assume that we encrypt plaintexts P and P' under an unknown key K and an unknown related-key K' such that $P \oplus P' = (0, e_{17})$ and $K \oplus K'$=$(0, 0, e_{17}, 0)$, respectively. Then, as described in Table 4, we can obtain our desired 32 full-round related-key differential characteristics $(0, e_{17})$→$(0, e_j)$ $(1 \leq j \leq 32)$ with probability 2^{-42}, which are built by alternatively using 2 one-round differential characteristics of $Crypt$, which we call *D1* and *D2* each. We also assume that the input difference of $Crypt$, ΔRI in the following two cases, is zero.

<u>*D1:* ΔRK=(ΔRK_L,ΔRK_R)=(e_{17},0)</u>

Since ΔRK_L=e_{17}, ΔRK_R=0 and ΔRI= 0, ΔO=0 and ΔZ=e_{17} (See Fig. 3-b)). Then, by $Property$ 1-f), the output difference of the first $F'_{32/80}$, ΔY, is e_{17} with probability 2^{-5} and the input and output differences of I are e_{17} and e_{25}, respectively. Similarly, by $Property$ 1-e), $\Delta Y'$ is e_{25} with probability 2^{-4} because of $\Delta Z'$=0 and $\Delta W'$=$(0, 0, 0, e_8, e_{13})$ (refer to Fig. 3-b)). Thus the input difference of $F^{-1}_{32/96}$ is zero and then the corresponding output difference $F^{-1}_{32/96}$ is zero with probability 1. Hence if ΔRI and ΔRK have $(0, 0)$ and (e_{17},0), respectively, then the corresponding output difference of $Crypt$ is (0,0) with probability 2^{-9}.

<u>*D2:* ΔRK=(ΔRK_L,ΔRK_R)=(0,e_{17})</u>

Since ΔRK_L=0, ΔRK_R=e_{17}, and ΔRI= 0, ΔO=0, ΔZ=0, and ΔW=(0,0,0, e_8,e_{13}). Next, the differential pattern of the first $F'_{32/80}$ follows that of the second $F'_{32/80}$ with probability 2^{-4} in the case of *D1*. So, the input and output differences of I are e_{25} and e_{17}, respectively. Similarly, the differential pattern of the second $F'_{32/80}$ follows that of the first $F'_{32/80}$ with probability 2^{-5} in the case of *D1* because ΔZ=e_{17} and $\Delta W'$=0, and the input difference of $F^{-1}_{32/96}$ is zero and then the corresponding output difference $F^{-1}_{32/96}$ is zero with probability 1. Hence, if ΔRI and ΔRK have $(0,0)$ and (0,e_{17}), respectively, then the corresponding output difference of $Crypt$ is (0,0) with probability 2^{-9}.

However, in the last round, we use a differential characteristic *D1'* which is similar to *D1* for a simplicity of our key recovery attack.

[*D1':*] ΔRK=(ΔRK_L,ΔRK_R)=(e_{17},0)

Since ΔRK_L=e_{17}, ΔRK_R=0, and ΔRI= 0, ΔO=0, ΔZ=e_{17}, and ΔW=(0,0,0, 0,0). Then, by $Property$ 1-f), ΔY=e_{23} with probability 2^{-5} and the output difference of I is e_{31}. Since $\Delta Z'$=0 and $\Delta W'$=$(0,0,0,e_8,e_{13})$, by $Property$ 1-e) $\Delta Y'$=0 with probability 2^{-4}. So, the input difference of $F^{-1}_{32/96}$ is e_{31}. Furthermore, the output difference 16th $F_{2/1}$-box of the first layer in $F^{-1}_{32/96}$ can be fixed as (0,1) with probability 2^{-1} by $Property$ 1-c). Then, as like in Fig. 5, this nonzero one bit difference always moves to the jth-bit of the output difference in $F^{-1}_{32/96}$ with probability 2^{-5} $(1 \leq j \leq 32)$. Thus, for any fixed j, $\Delta F^{-1}_{32/96(\Delta V'=0)}(\Delta X = e_{31})$=$e_j$ with probability 2^{-6}. Hence the related-key differential characteristic of the last round holds with probability 2^{-15}.

Table 4: Related-key differential characteristic of CHESS-64

R (i)	ΔRI^i	ΔRK^i	Pro.	Ca.
IT	$(0,e_{17})$	$(0,e_{17})$	1	·
1	$(0,0)$	$(e_{17},0)$	2^{-9}	*C1*
2	$(0,0)$	$(0,0)$	1	·
3	$(0,0)$	$(0,0)$	1	·
4	$(0,0)$	$(0,e_{17})$	2^{-9}	*C2*
5	$(0,0)$	$(0,0)$	1	·
6	$(0,0)$	$(0,e_{17})$	2^{-9}	*C2*
7	$(0,0)$	$(0,0)$	1	·
8	$(0,0)$	$(e_{17},0)$	2^{-15}	*C1'*
FT	$(0,e_j)$	$(0,0)$	1	·
Outp.	$(0,e_j)$	·	·	·
Total	·	·	2^{-42}	·

- $j(1 \leq j \leq 32)$: fixed values, Outp.:Output, Pro.:Probability, Ca.:Case

4.2 The First Key Recovery Attack

In this attack, we apply our 32 full-round related-key differential characteristics to retrieve a part of the master key of CHESS-64. This attack is based on the fact that there is an unique difference route when the input and output differences with hamming weight 1 are fixed in $F^{-1}_{32/96}$.

To begin with, we encrypt 2^{43} plaintext pairs P=(P_L, P_R) and P'=$P \oplus (0, e_{17})$ under an unknown key K=$(K_1$,K_2,K_3,$K_4)$ and an unknown related-key K'=$(K_1$,K_2,$K_3 \oplus e_{17}$,$K_4)$, respectively, and then get the 2^{43} corresponding ciphertext pairs C and C', i.e., $E_K(P)$=C and $E_{K'}(P)$=C', where E is the block cipher CHESS-64. Since our full-round related-key differential characteristics of CHESS-64 have a probability of 2^{-42} each, we expect about 2 ciphertext pairs (C, C') such that $C \oplus C' = (0, e_j)$ for each j $(1 \leq j \leq 32)$. According to our differential characteristics described in Table 4, we can deduce that the jth one-bit difference in such (C, C') are derived from the output differences of $F_{2/1}^{({V_6}'^{16})}$ in $F^{-1}_{32/96}$ of the last round (refer to Figs. 5). That is, we can expect that there is a unique differential route.

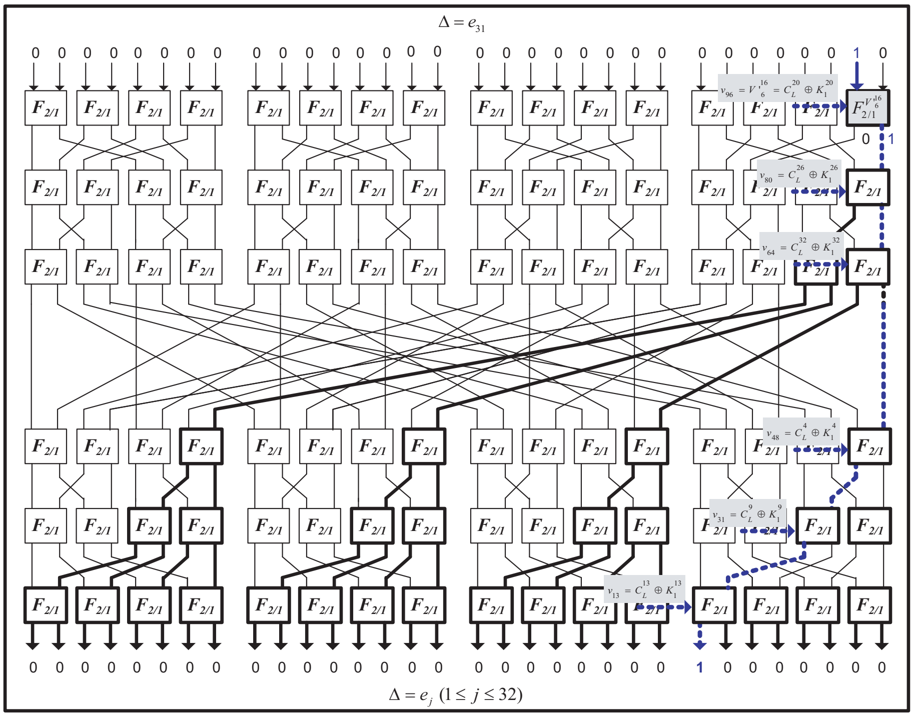

Figure 5: The possible routes of the nonzero output of the $F_{2/1}^{V_6'^{16}}$ in $F^{-1}_{32/96}$

Then, by using Property 3 we can extract 6 bits of control vectors for this route. For example, assume that the difference of C_R is e_{25}. Then we can obtain the following 6 bits of control vectors, which are expressed as a linear equation for K_j^i and C_L^i (refer to Fig. 5 and Table 5). Here, we let v_i, K_j^i, C_L^i denote the ith-bit of a controlling vector V, a subkey K_j, and of a left half of ciphertext C_L, respectively. Thus, we can know 6 bits key information because v_i and C_L^i are known values.

$$\begin{pmatrix} v_{96} = C_L^{20} \oplus K_1^{20} = 0, \\ v_{80} = C_L^{26} \oplus K_1^{26} = 1, \\ v_{64} = C_L^{32} \oplus K_1^{32} = 1, \\ v_{48} = C_L^{4} \oplus K_1^{4} = 0, \\ v_{31} = C_L^{9} \oplus K_1^{9} = 0, \\ v_{13} = C_L^{13} \oplus K_1^{13} = 0. \end{pmatrix}$$

Based on this idea we can devise the following key recovery algorithm on the full-round CHESS-64.

1. For CHESS-64, prepare 2^{43} plaintext pairs (P_i, P_i'), $i = 1, \cdots, 2^{43}$, which have the $(0, e_{17})$ difference each. All P_i are encrypted using a master key K and all P_i' are encrypted using a master key K' where K and K' have the $(0, 0, e_{17}, 0)$ difference. Encrypt each plaintext pair (P_i, P_i') to get the corresponding ciphertext pair (C_i, C_i').
2. Check that $C_i \oplus C_i' = (0, e_j)$ for each i and j ($1 \leq j \leq 32$).
3. For each ciphertext pair (C_i, C_i') passing Step the test of 2, extract some bits of controlling vectors by chasing a difference route between this j-PBO and the position of the $31st$ input bit in $F_{32/96}^{-1}$ (See Fig. 5). Then find the corresponding bits of K_1. Note that the controlling vector V' of $F_{32/96}^{-1}$ in the last round is formatted with $C_L \oplus K_1$.

The data complexity of this attack is 2^{44} related-key chosen plaintexts. The time complexity of Step 1 is 2^{44} full-round CHESS-64 encryptions, which is a much larger complexity than those of Step 2 and 3. By our related-key differential characteristics each ciphertext pair can pass Step 2 with probability at least 2^{-42} and thus the expectation of ciphertext pairs with the $(0, e_j)$ difference that pass this test is at least 1. So we have in Step 3 at least one ciphertext pair with the $(0, e_j)$ difference for each j ($1 \leq j \leq 32$). Thus we can retrieve 28 bits of information of keys in the lower layer of $F_{32/96}^{-1}$ and 4 bits of information of keys in the upper layer of $F_{32/96}^{-1}$ with a data and a time complexities of 2^{44}. Note that these 32 bits are recovered from $F_{2/1}$-boxes with bold line. However, since there are overlapping bits among the recovered key information, we actually know 20-bit key information of K_1. This attack can also be simply extended to retrieve the whole of master key pair by performing an exhaustive search for the remaining keys: since the exhaustive search has a running time of 2^{108} encryptions, recovering the full key based on the above attack works with the same data complexity and with a time complexity of about 2^{108} full-round CHESS-64 encryptions.

4.3 The Second Key Recovery Attack

We construct another full-round related-key differential characteristic with probability 2^{-37} which is the same as that used in the first attack except for the differential patterns of the 2nd, 3rd, 4th, 5th layers of $F_{32/96}^{-1}$ in the last round. In this attack, for increasing the differential probability, we use differential patterns of the 2nd, 3rd, 4th, 5th, 6th layers of $F_{32/96}^{-1}$ in the last round with probability 1. In other words, we use the fact that the nonzero input bit difference of 16th $F_{2/1}$ always moves to the jth-bit output difference of the $6th$ layer with probability 1 because when the input and controlling vector differences of $F_{2/1}$ are $(0, 1)$ and 0, respectively, the hamming weight of the corresponding output difference is always 1. Thus, if we apply them to a modification of attack algorithm then we can succeed in finding a 6-bit key with data and time complexities of 2^{39}. This is the same as the first algorithm only except the second step. In this algorithm the following Step 2' is used instead of STEP 2 of the first attack algorithm:

– Step 2' Check that $C_i \oplus C'_i = (0, e_j)$ for each i, where $j \in [1, 32]$.

Furthermore, we can extend it to recover the whole of master key pair by performing an exhaustive search for the remaining keys: since the exhaustive search has a running time of 2^{122} encryptions, recovering the full key based on the above attack works with the same data complexity and with a time complexity of about 2^{122} full-round CHESS-64 encryptions.

5 Conclusion

CHESS-64 have been designed for giving a fast and cheap hardware implementation and a high security as well. In this paper, however, we have presented the first known attack results on CHESS-64. According to our results, the full-round CHESS-64 is broken by using a related-key differential attack with 2^{39} related-key chosen plaintexts and 2^{39} encryptions. Our results demonstrate that CHESS-64 can induce a security risk in a cryptographic system connected pervasive computing environments where CHESS-64 is used with related keys for a relatively long period.

Acknowledgement

The second and third authors were supported by the Second Brain Korea 21 Project.

Table 5: Classes of the controlling vectors corresponding to the difference route in Fig. 5

Class	e_i	Controlling vectors
CL_1	e_1	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 0(0), v_{63} = C_L^{31} \oplus K_1^{31} = 0(0)$
	(e_2)	$v_{36} = C_L^{8} \oplus K_1^{8} = 0(0), v_{19} = C_L^{13} \oplus K_1^{13} = 0(0), v_{1} = C_L^{1} \oplus K_1^{1} = 0(1)$
CL_2	e_3	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 0(0), v_{63} = C_L^{31} \oplus K_1^{31} = 0(0)$
	(e_4)	$v_{36} = C_L^{8} \oplus K_1^{8} = 0(0), v_{19} = C_L^{13} \oplus K_1^{13} = 1(1), v_{2} = C_L^{2} \oplus K_1^{2} = 0(1)$
CL_3	e_5	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 0(0), v_{63} = C_L^{31} \oplus K_1^{31} = 0(0)$
	(e_6)	$v_{36} = C_L^{8} \oplus K_1^{8} = 1(1), v_{20} = C_L^{14} \oplus K_1^{14} = 0(0), v_{3} = C_L^{3} \oplus K_1^{3} = 0(1)$
CL_4	e_7	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 0(0), v_{63} = C_L^{31} \oplus K_1^{31} = 0(0)$
	(e_8)	$v_{36} = C_L^{8} \oplus K_1^{8} = 1(1), v_{20} = C_L^{14} \oplus K_1^{14} = 1(1), v_{4} = C_L^{4} \oplus K_1^{4} = 0(1)$
CL_5	e_9	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 0(0), v_{63} = C_L^{31} \oplus K_1^{31} = 1(1)$
	(e_{10})	$v_{40} = C_L^{12} \oplus K_1^{12} = 0(0), v_{23} = C_L^{1} \oplus K_1^{1} = 0(0), v_{5} = C_L^{5} \oplus K_1^{5} = 0(1)$
CL_6	e_{11}	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 0(0), v_{63} = C_L^{31} \oplus K_1^{31} = 1(1)$
	(e_{12})	$v_{40} = C_L^{12} \oplus K_1^{12} = 0(0), v_{23} = C_L^{1} \oplus K_1^{1} = 1(1), v_{6} = C_L^{6} \oplus K_1^{6} = 0(1)$
CL_7	e_{13}	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 0(0), v_{63} = C_L^{31} \oplus K_1^{31} = 1(1)$
	(e_{14})	$v_{40} = C_L^{12} \oplus K_1^{12} = 1(1), v_{24} = C_L^{2} \oplus K_1^{2} = 0(0), v_{7} = C_L^{7} \oplus K_1^{7} = 0(1)$
CL_8	e_{15}	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 0(0), v_{63} = C_L^{31} \oplus K_1^{31} = 1(1)$
	(e_{16})	$v_{40} = C_L^{12} \oplus K_1^{12} = 1(1), v_{24} = C_L^{2} \oplus K_1^{2} = 1(1), v_{8} = C_L^{8} \oplus K_1^{8} = 0(1)$
CL_9	e_{17}	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 1(1), v_{64} = C_L^{32} \oplus K_1^{32} = 0(0)$
	(e_{18})	$v_{44} = C_L^{16} \oplus K_1^{16} = 0(0), v_{27} = C_L^{5} \oplus K_1^{5} = 0(0), v_{9} = C_L^{9} \oplus K_1^{9} = 0(1)$
CL_{10}	e_{19}	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 1(1), v_{64} = C_L^{32} \oplus K_1^{32} = 0(0)$
	(e_{20})	$v_{44} = C_L^{16} \oplus K_1^{16} = 0(0), v_{27} = C_L^{5} \oplus K_1^{5} = 1(1), v_{10} = C_L^{10} \oplus K_1^{10} = 0(1)$
CL_{11}	e_{21}	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 1(1), v_{64} = C_L^{32} \oplus K_1^{32} = 0(0)$
	(e_{22})	$v_{44} = C_L^{16} \oplus K_1^{16} = 1(1), v_{28} = C_L^{6} \oplus K_1^{6} = 0(0), v_{11} = C_L^{11} \oplus K_1^{11} = 0(1)$
CL_{12}	e_{23}	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 1(1), v_{64} = C_L^{32} \oplus K_1^{32} = 0(0)$
	(e_{24})	$v_{44} = C_L^{16} \oplus K_1^{16} = 1(1), v_{28} = C_L^{6} \oplus K_1^{6} = 1(1), v_{12} = C_L^{12} \oplus K_1^{12} = 0(1)$
CL_{13}	e_{25}	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 1(1), v_{64} = C_L^{32} \oplus K_1^{32} = 1(1)$
	(e_{26})	$v_{48} = C_L^{4} \oplus K_1^{4} = 0(0), v_{31} = C_L^{9} \oplus K_1^{9} = 0(0), v_{13} = C_L^{13} \oplus K_1^{13} = 0(1)$
CL_{14}	e_{27}	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 1(1), v_{64} = C_L^{32} \oplus K_1^{32} = 1(1)$
	(e_{28})	$v_{48} = C_L^{4} \oplus K_1^{4} = 0(0), v_{31} = C_L^{9} \oplus K_1^{9} = 1(1), v_{14} = C_L^{14} \oplus K_1^{14} = 0(1)$
CL_{15}	e_{29}	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 1(1), v_{64} = C_L^{32} \oplus K_1^{32} = 1(1)$
	(e_{30})	$v_{48} = C_L^{4} \oplus K_1^{4} = 1(1), v_{32} = C_L^{10} \oplus K_1^{10} = 0(0), v_{15} = C_L^{15} \oplus K_1^{15} = 0(1)$
CL_{16}	e_{31}	$v_{96} = C_L^{20} \oplus K_1^{20} = 0(0), v_{80} = C_L^{26} \oplus K_1^{26} = 1(1), v_{64} = C_L^{32} \oplus K_1^{32} = 1(1)$
	(e_{32})	$v_{48} = C_L^{4} \oplus K_1^{4} = 1(1), v_{32} = C_L^{10} \oplus K_1^{10} = 1(1), v_{16} = C_L^{16} \oplus K_1^{16} = 0(1)$

References

[Biham and Shamir 1993] Biham E., Shamir A.: *Differential Cryptanalysis of the Data Encryption Standard*, ISBN: 0-387-97930-1, 3-540-97930-1, 1993.

[Goots et al. 2003] Goots N. D., Izotov B. V., Moldovyan A. A., Moldovyan N. A.: *Modern cryptography: Protect Your Data with Fast Block Ciphers*, Wayne, A-LIST Publish., 2003.

[Goots et al. 2003a] Goots N. D., Izotov B. V., Moldovyan A. A., Moldovyan N. A.: *Fast Ciphers for Cheap Hardware : Differential Analysis of SPECTR-H64*, *MMM-ACNS'03*, LNCS 2776, pp. 449-452, Springer-Verlag, 2003.

[Goots et al. 2003b] Goots N. D., Moldovyan N. A., Moldovyanu P. A., Summerville D. H.: *Fast DDP-Based Ciphers: From Hardware to Software*, *46th IEEE Midwest International Symposium on Circuits and Systems*, 2003.

[Goots et al. 2001] Goots N. D., Moldovyan A. A., Moldovyan N. A.: *Fast Encryption Algorithm Spectr-H64*, *MMM-ACNS'01*, LNCS 2052, pp. 275-286, Springer-Verlag, 2001.

[Kavut and Yücel 2002] Kavut S., Yücel M. D.: *Slide Attack on Spectr-H64*, *INDOCRYPT'02*, LNCS 2551, pp. 34-47, Springer-Verlag, 2002.

[Kelsey et al. 1996] Kelsey J., Schneier B., Wagner D.: *Key Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES*, *Advances in Cryptology - CRYPTO '96*, LNCS 1109, pp. 237-251, Springer-Verlag, 1996.

[Kelsey et al. 1997] Kelsey J., Schneier B., Wagner D.: *Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA*, *ICICS'97*, LNCS 1334, pp. 233-246, Springer-Verlag, 1997.

[Ko et al. 2003] Ko Y., Hong D., Hong S., Lee S., Lim J.: *Linear Cryptanalysis on SPECTR-H64 with Higher Order Differential Property*, *MMM-ACNS03*, LNCS 2776, pp. 298-307, Springer-Verlag, 2003.

[Ko et al. 2004] Ko Y., Lee C., Hong S., Lee S.: *Related Key Differential Cryptanalysis of Full-Round SPECTR-H64 and CIKS-1*, *ACISP 2004*, LNCS 3108, pp. 137-148, Springer-Verlag, 2004.

[Ko et al. 2004a] Ko Y., Lee C., Hong S., Sung J., Lee S.: *Related-Key Attacks on DDP based Ciphers: CIKS-128 and CIKS-128H*, *Indocrypt 2004*, LNCS 3348, pp. 191-205, Springer-Verlag, 2004.

[Lee et al. 2002] Lee C., Hong D., Lee S., Lee S., Yang H., Lim J.: *A Chosen Plaintext Linear Attack on Block Cipher CIKS-1*, *ICICS 2002*, LNCS 2513, pp. 456-468, Springer-Verlag, 2002.

[Lee et al. 2005] Lee C., Kim J., Hong S., Sung J., Lee S.: *Related-Key Differential Attacks on Cobra-S128, Cobra-F64a, and Cobra-F64b*, *MYCRYPT 2005*, LNCS 3715, pp. 245-263, Springer-Verlag, 2005.

[Lee et al. 2005a] Lee C., Kim J., Sung J., Hong S., Lee S.: *Related-Key Differential Attacks on Cobra-H64 and Cobra-H128*, *Tenth IMA International Conference On Cryptography and Coding (CCC 2005)*, LNCS 3796, pp. 201-219, Springer-Verlag, 2005.

[Matsui 1993] Matsui M.: *Linear cryptanalysis method for DES cipher*, *Advances in Cryptology - EUROCRYPTO'93*, LNCS 765, pp. 386-397, Springer-Verlag, 1993.

[Moldovyan 2002] Moldovyan A. A., Moldovyan N. A.: *A cipher Based on Data-Dependent Permutations*, *Journal of Cryptology*, volume 15, no. 1, pp. 61-72, 2002.

[Moldovyan et al. 2005] Moldovyan N. A., Sklavos N., Koufopavlou O.: *Pure DDP-Based Cipher: Architecture Analysis, Hardware Implementation Cost and Performance up to 6.5 Gbps*, *The International Arab Journal of Information Technology*, volume 2, no. 1, pp. 24-27, 2005.

[Moldovyan et al. 2005a] Moldovyan N. A., Sklavos N., Moldovyan A. A., Koufopavlou O.: *CHESS-64, a Block Cipher Based on Data-Dependent Operations: Design Variants and Hardware Implementation Efficiency*, *Asian Journal of Information Technology*, Grace Publications Network, volume 4, no. 4, pp. 323-334, 2005.

[Sklavos et al. 2005] Sklavos N., Moldovyan N. A., Koufopavlou O.: *High Speed Networking Security: Design and Implementation of Two New DDP-Based Ciphers*, *Mobile Networks and Applications-MONET*, *Kluwer Academic Publishers*, Vol. 25, Issue 1-2, pp. 219-231, 2005.

[Sklavos et al. 2003] Sklavos N., Moldovyan N. A., Koufopavlou O.: *Encryption and Dada Dependent Permutations: Implementation Cost and Performance Evaluation*, *MMM-ACNS 2003*, LNCS 2776, pp. 337-348, Springer-Verlag, 2003.

[Sklavos et al. 2003a] Sklavos N., Moldovyan N. A., Koufopavlou O., *A New DDP-based Cipher CIKS-128H: Architecture, Design & VLSI Implementation Optimization of CBC-Encryption & Hashing over 1 GBPS*, proceedings of *The 46th IEEE Midwest Symposium on Circuits & Systems*, December 27-30, Cairo, Egypt, 2003.

[Sklavos and Koufopavlou 2003] Sklavos N., Koufopavlou O.: *Dada Dependent Rotations, a Trustworthy Approach for Future Encryption and Systems/Ciphers: Low Cost and High Performance*, *Computers and Security*, *Elsevier Science Journal*, Vol. 22, No 7, 2003.

[Phan and Handschuh 2004] Phan R. C.-W, Handschuh H., *On Related-Key and Collision Attacks: The case for the IBM 4758 Cryptoprocessor*, *ISC 2004*, LNCS 3225, pp. 111-122, Springer-Verlag, 2004.

[Razali and Phan 2006] Razali E., Phan R. C.-W: *On the Existence of Related-Key Oracles in Cryptosystems Based on Block Ciphers*, *OTM Workshops 2006*, LNCS 4277, pp. 425-438, Springer-Verlag, 2006.