KT-64: A New Block Cipher Suitable to Efficient FPGA Implementation

Nguyen Hieu Minh, Nguyen Thien Luan, and Luu Hong Dung
Faculty of Information Technology, Le Qui Don Technical University
100 Hoang Quoc Viet, Ha Noi, Viet Nam

Summary
This paper proposes a new block cipher called KT-64. We make a cipher using components that are believed secure. The structure of KT-64 is very simple, strong and efficient. We use the controlled substitution-permutation networks (CSPNs) based on controlled elements (CEs) for designing fast block cipher suitable to cheap hardware implementation. Security estimations of KT-64 cipher show that proposed cipher is high-level security. The synthesis results for hardware implementation (FPGA) prove that KT-64 is very efficient new cipher.

Keywords
Block cipher, Data-dependent operations, Hardware implementation.

1. Introduction

Security is a primary requirement of any wired and wireless communication. Encryption algorithms are meant to provide secure communications applications. Optimizations of the existing security standards as well as novel designs are proved issues of major importance in order the high needs for security to be satisfied.

The growing requirements for high-speed, high level secure communications forces the system designers to propose the hardware implementation of cryptographic algorithms. However, cryptographic algorithms impose tremendous processing power demands that can be a bottleneck in high-speed networks.

FPGA devices are a highly promising alternative for implementing block cipher algorithms. Compared to software-based implementations, FPGA implementations can achieve superior performance. The fine-granularity of FPGAs matches extremely well the operations required by block cipher algorithms (e.g., bit-permutations, bit-substitutions, look-up table reads, Boolean functions) [1]. As a result, such operations can be executed more efficiently in FPGAs than in a general-purpose computer. Furthermore, the inherent parallelism of the algorithms can be efficiently exploited in FPGAs as opposed to the serial fashion of computing in a processor environment. At the cryptographic-round level, multiple operations can be executed concurrently.

The use of data-dependent transformations has been an area of increasing interest for the designers of ciphers. Data-Dependent Permutations (DDP) has attracted much attention the last few years in cryptography [2-4]. DDP based are competitive with the other well used encryption algorithms, such as AES, for different variants of hardware implementation [4-5]. Recently a class of the advanced DDP-like Operations (DDOs) has been proposed [6] to increase the efficiency of the hardware implementation of the DDO-based ciphers. In particular, data-dependent (DD) operations (DDOs) provide a fast and simple cryptologic primitive when implemented in hardware [7]. Efficiency of the Data-Dependent operations was demonstrated with examples of ciphers RC5 [8], RC6 [9], and MARS [10], which are based on DD rotations with 32 different modifications.

In this paper, we propose a new block cipher KT-64 with 64-bit block length and 128-bit key length based on DDOs, which is suitable to efficient FPGA implementation. KT-64 has an 8-round iterative structure which is a variant of generalized controlled substitution-permutation networks (CSPNs).

Furthermore, two architectures for new block cipher are considered. The Basic Looping Architecture, where only one round is implemented, and the Full Loop Unrolling Architecture, where the rounds are fully unrolled with pipeline stages between the consecutive rounds. The performance metrics that are used for the ciphers comparison are: (1) throughput, defined as the number of bits encrypted (decrypted) in a unit of time, (2) throughput per slice that measures the hardware resource cost associated with the implementation resulting throughput.

This paper is organized as follows: In section 2, the elementary building blocks theory is described. Section 3 describes the structure of the new block cipher: eight-round KT-64 with 64-bit data input and 128-bit key length. Section 4 presents the results on security estimations. Section 5 presents the FPGA synthesis results and comparisons of the proposed cipher with other block ciphers. Finally, conclusion.

Manuscript received January 5, 2010
Manuscript revised January 20, 2010

2. Controlled Substitution-Permutation Networks as Variable operations

The controlled operations are implemented as uniform CSPNs constructed using the minimum size CEs as standard building blocks (Figure 1a show general topology of CSPN). Let CSPN with n-bit input and n-bit output be controlled with m-bit vector v. Then we shall denote such CSPN as controlled operation $\mathbf{F}_{n/m}$. Selecting a set of the fixed permutations connecting active layers, we define some particular topology of controlled operation. Each active layer represents $n/2$ parallel CEs. A minimum size CE is denoted as the $\mathbf{F}_{2/1}$ box. It transforms two-bit input vector (x_1, x_2) into two-bit output (y_1, y_2) depending on a controlling bit v.

A CE can be represented as a pair of the 2×2 substitutions (elementary S-boxes) selected depending on bit v (Figure 1b) with substitution S_1 (if $v = 0$) and S_2 (if $v = 1$) on two-bit vectors. Such substitutions are denoted as $\mathbf{F}^{(0)}{}_{2/1}$ and $\mathbf{F}^{(1)}{}_{2/1}$ and CE implements the transformation $(y_1, y_2) = \mathbf{F}^{(v)}{}_{2/1}(x_1, x_2)$.

The $\mathbf{F}_{2/1}$ element can be also represented with a pair of BFs in three variables (Figure 1c): $y_1 = f_1(x_1, x_2, v)$; $y_2 = f_2(x_1, x_2, v)$. If the substitution $\mathbf{F}^{(0)}{}_{2/1}$ is described with two BFs $y'_1 = f'_1(x_1, x_2)$ and $y'_2 = f'_2(x_1, x_2)$, $\mathbf{F}^{(1)}{}_{2/1}$ is described with two BFs $y''_1 = f''_1(x_1, x_2)$ and $y''_2 = f''_2(x_1, x_2)$, then CE $\mathbf{F}^{(v)}{}_{2/1}$ is described with two BFs in three variables x_1, x_2, and v: $y_1 = (v \oplus 1) f'_1(x_1, x_2) \oplus v f''_1(x_1, x_2) = v(y'_1 \oplus y''_1) \oplus y'_1$, $y_2 = (v \oplus 1) f'_2(x_1, x_2) \oplus v f''_2(x_1, x_2) = v(y'_2 \oplus y''_2) \oplus y'_2$.

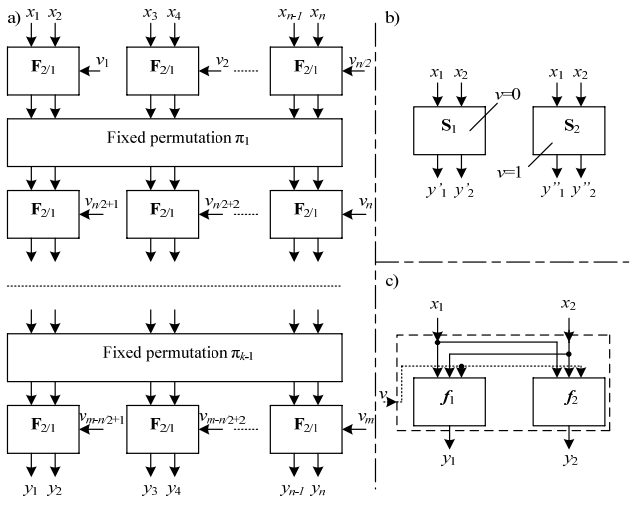


Figure 1. a) General structure of the $\mathbf{F}_{n/m}$ boxes, b) representation of the $\mathbf{F}_{2/1}$ as two 2×2 substitutions, c) or as a pair of BFs in three variables.

The selection of CEs $\mathbf{F}_{2/1}$ suitable to design efficient cryptographic DDO is based on the following criteria:

1. Each of two outputs of CEs should be a non-linear BF having maximum possible non-linearity NL for balanced BFs.
2. Each modification of CEs should be bijective transformation $(x_1, x_2) \rightarrow (y_1, y_2)$.
3. Each modification of CEs should be involution.
4. The linear combination of two outputs of CEs, i.e. $f = y_1 \oplus y_2$, should have maximum possible non-linearity NL for balanced BFs.

Trying all possible variants of the $\mathbf{F}_{2/1}$ elements we have established that there exist 24 different CEs $\mathbf{F}_{2/1}$ satisfying criteria 1-4. They implement only modifications shown in figure 2.

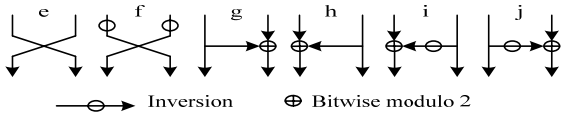


Figure 2. The 2×2 substitutions implemented by nonlinear controlled involutions.

Types of the CSPNs constructed using CEs $\mathbf{F}_{2/1}$ can be applied as DDOs suitable to designing fast hardware-oriented ciphers. For FPGA implementation, that has gained highly significant practical importance, all types of the CEs $\mathbf{F}_{2/1}$ are implemented using two 4-bit cells (Figure 1c), each implementing a Boolean Function (BF) with three variables. Advance of the DDO-based ciphers design, is to select and use non-linear CE with maximum non-linearity.

In [7] has show that with CEs $\mathbf{F}_{2/1}$ are divided into four subclasses $\{\mathbf{S}_{2/1}\}$, $\{\mathbf{R}_{2/1}\}$, $\{\mathbf{Z}_{2/1}\}$ and $\{\mathbf{L}_{2/1}\}$. The most interesting subclasses of CEs – namely $\{\mathbf{R}_{2/1}\}$ and $\{\mathbf{S}_{2/1}\}$ – for each specific type of CE also include its inverse element, it is meaning that $\mathbf{S}^{-1}{}_{2/1} = \mathbf{S}_{2/1}$ and $\mathbf{R}^{-1}{}_{2/1} = \mathbf{R}_{2/1}$. The subclasses $\mathbf{Z}_{2/1}$ and $\mathbf{L}_{2/1}$ be without such property, but $\mathbf{L}^{-1}{}_{2/1} = \mathbf{Z}_{2/1}$ and $\mathbf{Z}^{-1}{}_{2/1} = \mathbf{L}_{2/1}$.

When designing controlled operational blocks $\mathbf{F}_{n/m}$ for cryptographic applications, the order of controlled operations is interested specially. In this paper, block $\mathbf{F}_{n/m}$ $(\mathbf{F}_{32/96}, \mathbf{F}_{16/16})$ will be chosen as this block satisfies all the requirements for avalanche effect in proposed algorithm (proven in section 4) and it has the simplest structure so it will be perfectly suitable for constructing high performance algorithm.

Besides the NL value, differential characteristics (DCs) of the CE are important to characterize CEs as cryptographic primitives. Diferential characteristics (DCs) of the $\mathbf{F}_{n/m}$ boxes are defined by their topology and DCs of the elementary controlled boxes used as main building blocks while constructing the $\mathbf{F}_{n/m}$ boxes. As a general case, the differences passing through the $\mathbf{F}_{2/1}$ element are shown in figure 3, where $p\left(\Delta_i^Y / \Delta_j^X, \Delta_k^V\right)$ is probability to have the output difference is Δ_i^Y, if the input difference is Δ_j^X and the difference at the controlling input is Δ_k^V (indices indicate the number of non-zero bits in corresponding differences).

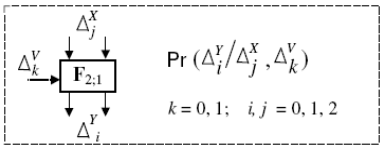


Figure 3. Differential characteristics of the $\mathbf{F}_{2/1}$ elements.

According to DCs, all non-linear elements $\mathbf{F}_{2/1}$ (the $\mathbf{S}_{2/1}$ type CEs) shows in table1 [7].

Table 1: Probabilities of Differential Characteristics

i	*j*	*k*	$\mathbf{S}_{2/1}$	*i*	*j*	*k*	$\mathbf{S}_{2/1}$
0	0	1	¼	2	1	0	½
1	0	1	½	1	2	0	1
2	0	1	¼	2	2	0	0
0	1	1	¼	0	2	1	¼
1	1	1	¼	1	2	1	½
2	1	1	½	2	2	1	¼
1	1	0	½	-	-	-	-

The results of the study of avalanche effect of different types CEs $\mathbf{F}_{2/1}$ used in controlled operational block $\mathbf{F}_{n/m}$ show that element types of **S** and **L** are with the best avalanche effect [7]. However, the element **L** does not have the reversibility property as mentioned above, thus in designing KT-64 algorithm the element $\mathbf{S}_{2/1}$ will be selected.

3. Design of the Cipher KT-64

In this section, we list brief description of design principles of KT-64:

1. Design new cipher with the goal of having low hardware implementation costs.
2. The encryption algorithm should be an iterated 64-bit block cipher and 128-bit key length.
3. The structure of KT-64 is generalized CSPN-like. Since encryption process is simply converted into decryption process, implementation of the circuit supporting both encryption and decryption processes does not require much more cost than the encryption-only circuit.
4. The cipher should be fast, in the case of frequent key refreshing. KT-64 using simple key scheme, so sure have to be key agile.

The KT-64 is a 64-bit block cipher using CSPNs that supports 128-bit key length. The general structure of the KT-64 is showed in figure 4. The procedure $\text{Crypt}^{(e)}$ is described figure 5.

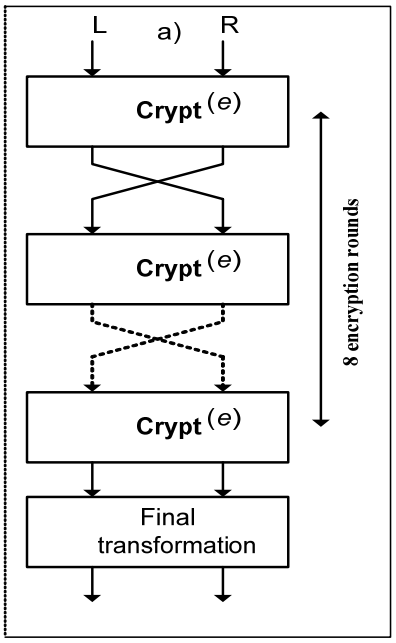


Figure 4. General structure of the KT-64.

In order to symmetries the full ciphering procedure we use very simple final transformation (FT) that is XORing two subkey with data subblocks. Due to FT in KT-64 the same algorithm performs both the encryption and the decryption, while different key scheduling is used. KT-64 uses 128-bit key $K = (K_1, K_2, K_3, K_4)$ (Q_j, $U_j \in K$, $K \in \{0, 1\}^{32}$) and very simple key scheduling that is the same while enciphering and deciphering. Thus, no preprocessing the secret key is used. The key scheduling is described in table 2.

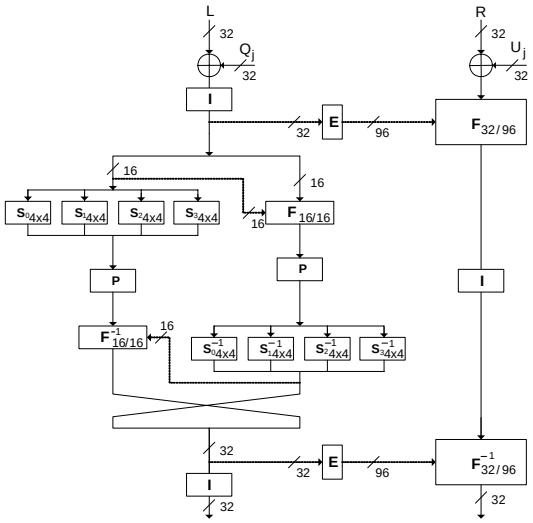


Figure 5. Cipher KT-64: procedure $\text{Crypt}^{(e)}$.

Table 2: The key scheduling in KT-64 (j = 9 corresponds to final transformation)

No. rounds j	1	2	3	4	5
Enc Q_j/U_j	K_1/K_2	K_3/K_4	K_3/K_1	K_4/K_1	K_2/K_3
Dec Q_j/U_j	K_1/K_3	K_3/K_4	K_2/K_1	K_4/K_3	K_3/K_2
No. rounds j	6	7	8	9	
Enc Q_j/U_j	K_3/K_4	K_1/K_2	K_4/K_3	K_1/K_3	
Dec Q_j/U_j	K_1/K_4	K_1/K_3	K_4/K_3	K_1/K_2	

Ciphering procedure of KT-64 is described as follows: $C = T^{(e=0)}(M, K)$ and $M = T^{(e=1)}(C, K)$, where M is the plaintext, C is the ciphertext ($M, C \in \{0,1\}^{64}$), T is the transformation function, and $e \in \{0,1\}$ is a parameter defining encryption ($e = 0$) or decryption ($e = 1$) mode.

First data block is divided into two 32-bit subblocks L and R and then using the procedure $\text{Crypt}^{(e)}$ eight encryption rounds are performed. The last round is followed by final transformation (FT).

The encryption algorithm is as follows:

1. For $i = 1$ to 7 do: $\{(L, R) \leftarrow \text{Crypt}^{(e)}(L, R, Q_j, U_j); (L, R) \leftarrow (R, L)\}$.
2. Perform transformation: $\{(L, R) \leftarrow \text{Crypt}^{(e)}(L, R, Q_8, U_8)\}$
3. Perform final transformation: $\{(L, R) \leftarrow (L \oplus Q_9, R \oplus U_9); (L, R) \leftarrow (L, R)\}$.

The 96-bit controlling vectors V and V' corresponding to the $\mathbf{F}_{32/96}$ and $\mathbf{F}^{-1}{}_{32/96}$ boxes is formed with the extension box **E** described as follows:

The controlling vector V contains s components, where s = 6 is the number of the layers in the $\mathbf{F}_{32/96}$ box, i.e. $V = (V_1, V_2, V_3, V_4, V_5, V_6)$.

$$\mathbf{E}(X) = V = (V_1, V_2, V_3, V_4, V_5, V_6);$$

$$V1 = (v_7, v_8, v_1, v_2, v_{16}, v_{15}, v_{10}, v_9, v_5, v_6, v_3, v_4, v_{11}, v_{12}, v_{13}, v_{14});$$

$$V2 = (v_9, v_{10}, v_{11}, v_{12}, v_1, v_2, v_7, v_8, v_{13}, v_{14}, v_{15}, v_{16}, v_5, v_6, v_3, v_4);$$

$$V3 = (v_{13}, v_{14}, v_{15}, v_{16}, v_5, v_6, v_3, v_4, v_1, v_2, v_7, v_8, v_9, v_{10}, v_{11}, v_{12});$$

$$V4 = (v_{21}, v_{22}, v_{29}, v_{30}, v_{25}, v_{26}, v_{23}, v_{24}, v_{31}, v_{32}, v_{27}, v_{28}, v_{17}, v_{18}, v_{19}, v_{20});$$

$$V5 = (v_{31}, v_{32}, v_{27}, v_{28}, v_{17}, v_{18}, v_{19}, v_{20}, v_{29}, v_{30}, v_{25}, v_{26}, v_{21}, v_{22}, v_{23}, v_{24});$$

$$V6 = (v_{19}, v_{20}, v_{23}, v_{24}, v_{27}, v_{28}, v_{29}, v_{30}, v_{21}, v_{22}, v_{17}, v_{18}, v_{32}, v_{31}, v_{25}, v_{26}).$$

Where bits v_i correspond to vector $V = (v_1, v_2, ..., v_{32})$ that is input of the **E**-box and the output vector is $V = (V_1, V_2, ..., V_6)$.

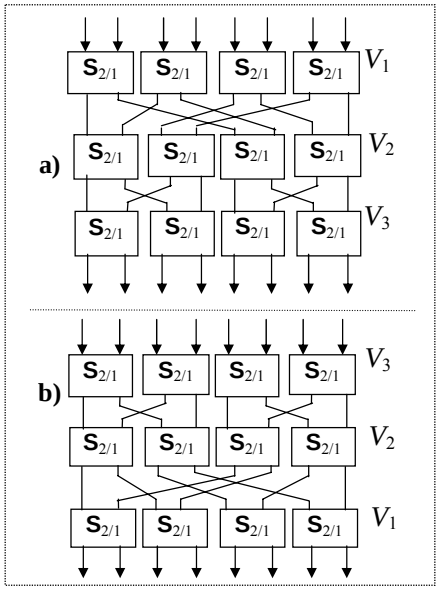
Design of the operational boxes $\mathbf{F}_{n/m}$ ($\mathbf{S}_{n/m}$) includes the following two items: (1) selection of the fixed permutations between active layers and (2) selection of the types of active layers.

Initially, we construct the $\mathbf{F}_{8/12}$ (figure 6a) and $\mathbf{F}^{-1}{}_{8/12}$ boxes (figure 6b) containing three active layers are used as main building blocks while constructing the six layer boxes $\mathbf{F}_{32/96}$ (figure 6c) and $\mathbf{F}^{-1}{}_{32/96}$ (figure 6d). The boxes $\mathbf{F}_{32/96}$ and $\mathbf{F}^{-1}{}_{32/96}$ that are mutual inverses (the box $\mathbf{F}^{-1}{}_{32/96}$ is constructed inverse with box $\mathbf{F}_{32/96}$).

The CSPNs implements the $\mathbf{F}_{8/12}$ and $\mathbf{F}_{32/96}$ operations built up using the (i, j) element as standard building block.

The fixed permutation involution $\mathbf{I}_1$ corresponding to connections between four parallel boxes $\mathbf{F}_{8/12}$ and four parallel boxes $\mathbf{F}^{-1}{}_{8/12}$ in $\mathbf{F}_{32/96}$-box is described as follows:

(1)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,29)(10)(11,18)(12, 26)(14)(15,22)(16,30)(19)(20,27)(23)(24,31)(28)(32).



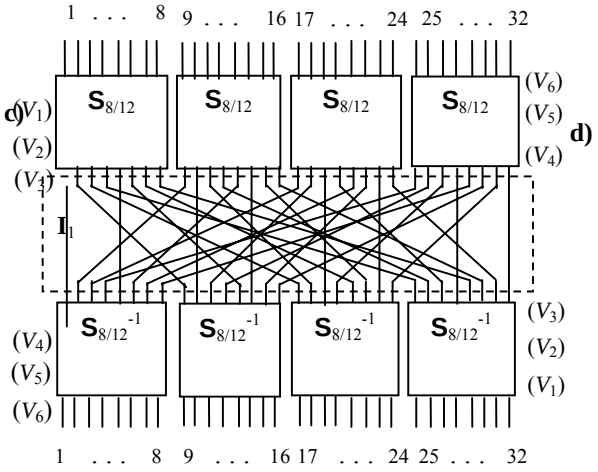


Figure 6. Topology of the DDO boxes: a) $\mathbf{F}_{8/12}$, b) $\mathbf{F}^{-1}{}_{8/12}$,) $\mathbf{F}_{32/96}$, b) $\mathbf{F}^{-1}{}_{32/96}$.

The boxes $\mathbf{F}_{32/192}$ and $\mathbf{F}^{-1}{}_{32/192}$ can be represented as the superposition $\mathbf{F}_{32/96} \bullet \mathbf{I} \bullet \mathbf{F}^{-1}{}_{32/96}$. The permutational involution **I** is described as follows:

(1)(2,9)(3,17)(4,25)(5)(6,13)(7,21)(8,29)(10)(11,18)(12, 26)(14)(15,22)(16,30)(19)(20,27)(23)(24,31)(28)(32).

Due to symmetric topology the difference between the boxes $\mathbf{F}_{32/96}$ and $\mathbf{F}^{-1}{}_{32/96}$ consists only in the use of the controlling vector components V_1, V_2, ...,V_6.
Then, construct structure of the $\mathbf{F}_{16/16}$-box (figure 7):

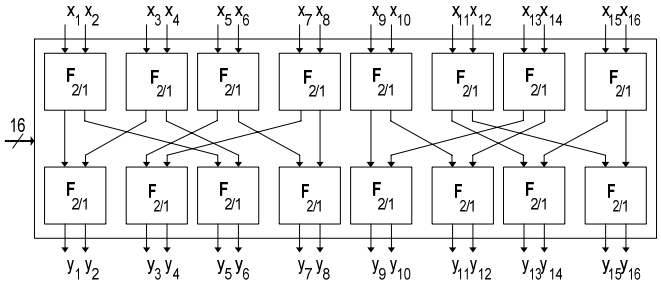


Figure 7. Topology of the $\mathbf{F}_{16/16}$–box.

KT-64 use 4×4 S-box substitutions: direct ones S_0, ..., S_3 and inverses S_0^{-1}, ..., S_3^{-1} boxes (specified in table 3). Four 4×4 S-boxes of the DES cipher (one from each of eight 6×4 S-boxes) have been selected as the S_0, …, S_3 boxes of KT-64 in order to inspire a high level of public confidence that no trapdoor are inserted in KT-64. Similar justification of the S-boxes selection has been earlier used in the design of the Serpent cipher [11].
The permutational involution **P** is described as follows:

(1)(2,5)(3,9)(4,13)(6)(7,10)(8,14)(11)(12,15)(16).

Table 3: Specification of the 4 x 4 Subsitutions Boxes S_0, ..., S_3 (S_0^{-1}, ..., S_3^{-1})

S	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\mathbf{S_0}/\mathbf{S_0}^{-1}$	14/14	4/3	13/4	1/8	2/1	15/12	11/10	8/15	3/7	10/13	6/9	12/6	5/11	9/2	0/0	7/5
$\mathbf{S_1}/\mathbf{S_1}^{-1}$	3/9	13/10	4/5	7/0	15/2	2/15	8/12	14/3	12/6	0/13	1/11	10/14	6/8	9/1	11/7	5/4
$\mathbf{S_2}/\mathbf{S_2}^{-1}$	10/1	0/8	9/14	14/5	6/13	3/7	15/4	5/11	1/15	13/2	12/0	7/12	11/10	4/9	2/3	8/6
$\mathbf{S_3}/\mathbf{S_3}^{-1}$	1/12	4/0	11/15	13/5	12/1	3/13	7/10	14/6	10/11	15/14	6/8	8/2	0/4	5/3	9/7	2/9

4. Security Estimations

4.1 Nessie Test

Investigation of statistic properties of KT-64 has been carried out with standard tests, which have been used in [12] for testing five AES finalists. Our research results have shown that three rounds of KT-64 are sufficient to satisfy the test criteria. Thus, KT-64 possesses good statistical properties like that of AES finalists.

4.2 Differential Cryptanalysis

The resistance of a block cipher against differential cryptanalysis [13] depends on the maximum probability of differential characteristics, which are paths from the plaintext difference to the ciphertext difference.
Formation schemes of the characteristic corresponding to the difference (Δ^L_1, Δ^R_0) and (Δ^L_0, Δ^R_1) are presented in figures 8, 9, 10, 11, 12, 13. Three cases will occur:

- One active bit Δ^L_1 passes through the left of the left branch of the KT-64 (figures 8 and 9).
- One active bit Δ^L_1 passes through the right of the left branch of the KT-64 (figures 10 and 11).
- One active bit Δ^L_1 passes through the right branch of the KT-64 (figures 12 and 13).

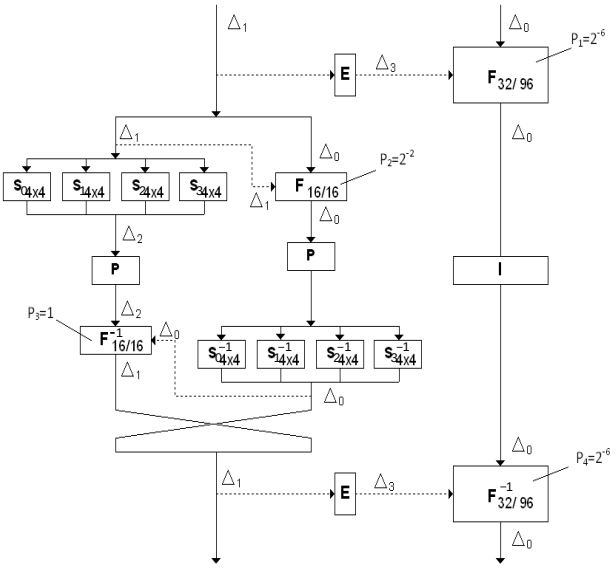


Figure 8. Formation of the one-round differential characteristic with the difference $(\Delta^L_1, \Delta^R_0) \rightarrow (\Delta^L_1, \Delta^R_0)$ with probability $P_1 = 2^{-14}$.

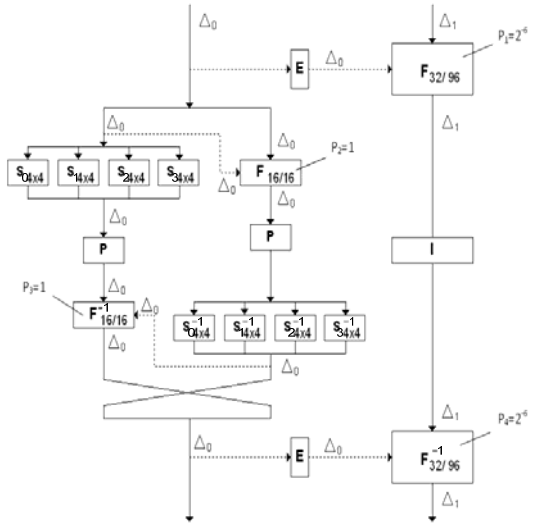
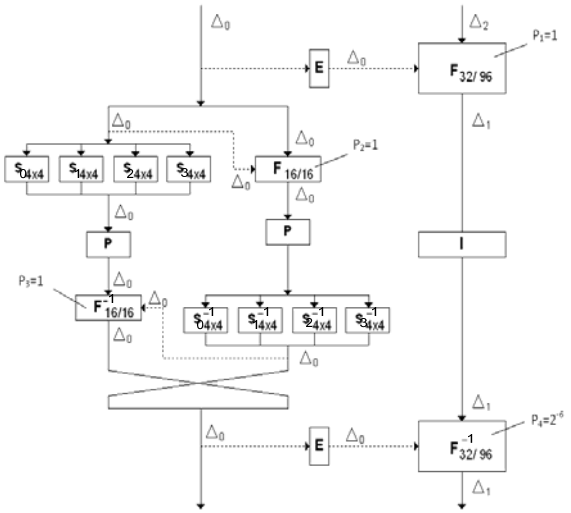


Figure 9. Formation of the two-round iterative differential characteristic with the difference $(\Delta^L_1, \Delta^R_0) \rightarrow (\Delta^L_0, \Delta^R_1)$ with probability $P_2 = 2^{-26}$.



Formation of the two-round iterative differential characteristic with the difference $(\Delta^L_1, \Delta^R_0) \rightarrow (\Delta^L_0, \Delta^R_1)$ with probability $P_2 = 2^{-30}$.

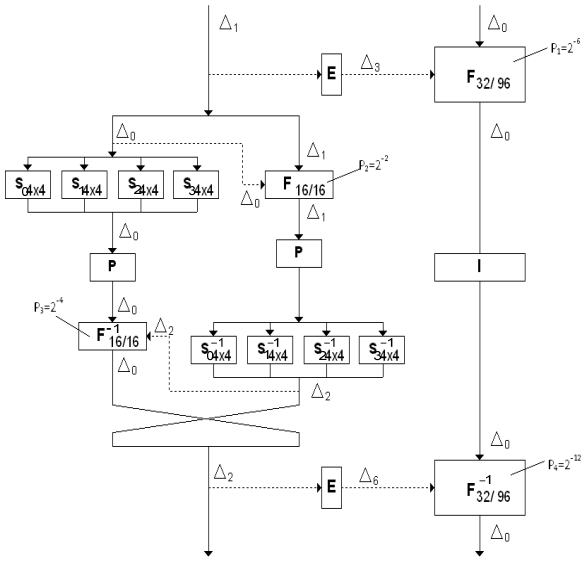
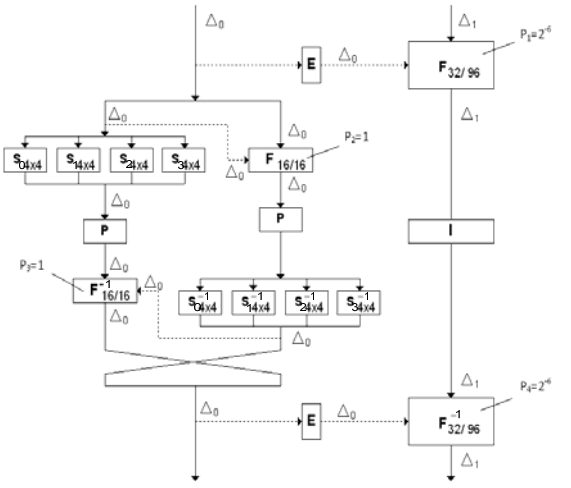


Figure 10. Formation of the one-round differential characteristic with the difference $(\Delta^L_1, \Delta^R_0) \rightarrow (\Delta^L_2, \Delta^R_0)$ with probability $P_1 = 2^{-24}$.



Formation of the one-round differential characteristic with the difference $(\Delta^L_0, \Delta^R_1) \rightarrow (\Delta^L_0, \Delta^R_1)$ with probability $P_1 = 2^{-12}$.

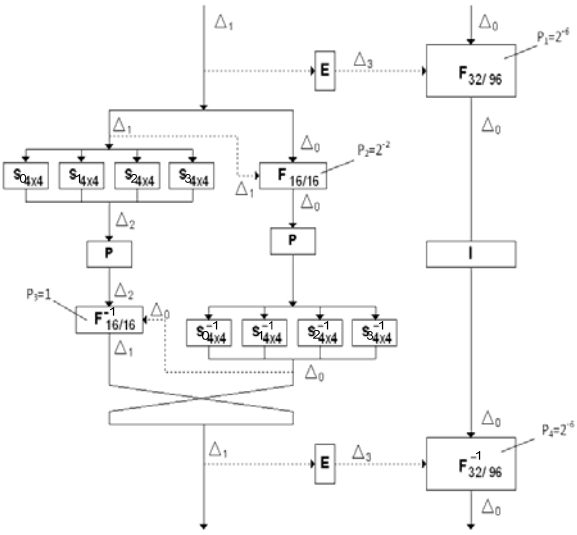


Figure 11. Formation of the two-round iterative differential characteristic with the difference $(\Delta^L_0, \Delta^R_1) \rightarrow (\Delta^L_1, \Delta^R_0)$ with probability $P_2 = 2^{-26}$.

The best iterative differential characteristic are presented in table 4, where $(\Delta^L_1, \Delta^R_0) \rightarrow (\Delta^L_0, \Delta^R_1)$ denote input and output differences, respectively.

Table 4: The Best Differential Characteristics KT-64

Input difference	Output difference	n	P_n
(Δ^L_1, Δ^R_0)	(Δ^L_0, Δ^R_1)	2	2^{-26}
(Δ^L_1, Δ^R_0)	(Δ^L_0, Δ^R_1)	2	2^{-30}
(Δ^L_0, Δ^R_1)	(Δ^L_1, Δ^R_0)	2	2^{-26}
(Δ^L_1, Δ^R_0)	(Δ^L_0, Δ^R_1)	6	2^{-78}
(Δ^L_1, Δ^R_0)	(Δ^L_0, Δ^R_1)	6	2^{-90}
(Δ^L_0, Δ^R_1)	(Δ^L_1, Δ^R_0)	6	2^{-78}

Table 4 shows that the probability of the existence of the differential trail after the second round is less than 2^{-26} and after the six round the probability of the differential trail is less than 2^{-78} thus 6 rounds is enough to prevent the difference cryptanalysis. In order for the security eight rounds is selected to prevent other types of attacks.

5. Architectures and FPGA Implementations

KT-64 is examined in hardware implementation by using two different architectures: the basic looping architecture (BLA), and the full loop unrolling architecture (FLUA) for FPGA device. Figures 14a and 14b show the block diagrams of BLA and FLUA implementations, respectively.

In the first one, only one round of cipher KT-64 is implemented in order to decrement the required hardware resources. The output of the basic round unit is buffered and one additional register is used for the input plaintext storage. During initialization the multiplexer chooses the plaintext and then chooses the output of the basic round unit. The major disadvantage of this architecture is the requirement of more clock cycles in order to perform the complete cipher. This is because for an n-round cipher, n clock cycle is required to perform encryption (8 for KT-64). The key expansion unit produces the appropriate round keys, which are stored and loaded in the used RAM blocks. One round of the encryption algorithm is performed by the Data Transformation Round Core. This core is a flexible combinational logic circuit and it is supported by an n-bit register and n-bit multiplexer (64-bit for KT-64). In the first clock cycle, the n-bit plaintext/ciphertext is forced into the data transformation round core. Then in each clock cycle, one round of the cipher is performed and the transformed data are stored into the register. According to BLA a 64-bit data block is completely transformed every 8 clock cycles for KT-64 (8 transformation rounds).

In a loop unrolling architecture where all n-rounds of the data encryption part and the key scheduling part are unrolled and implemented, the required hardware resources are increased. The key scheduling part is implemented with pipeline stages in order to balance the pipelining in each data encryption round. This approach minimizes the number of clock cycles required for encryption and increases the throughput. For an n-round cipher, n-rounds are unrolled, n pipeline stages are used and it is capable to process n data blocks simultaneously. The pipelining architecture offers the benefit of the high-speed performance. The implementation can be applied in applications with hard throughput needs. This goal is achieved by using a number of operating blocks with a final cost to the covered area. The proposed architecture uses 8 basic round blocks for KT-64, which is cascaded by using equal number of pipelined registers. Based on this design approach, 8 of 64-bit data blocks can be processed at the same time for KT-64. Pipelined proposed architecture produces a new plaintext/ciphertext block every clock cycle.

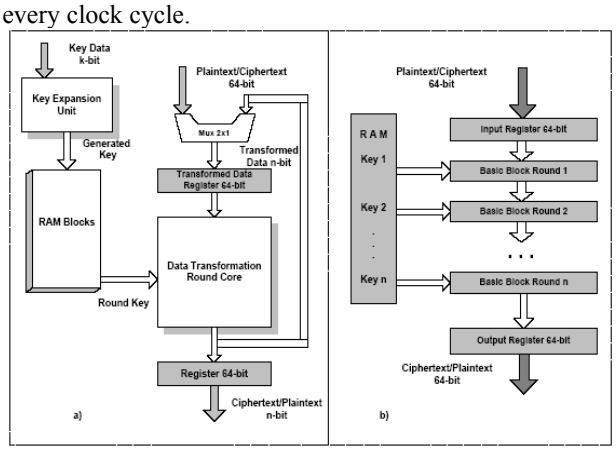


Figure 12. a) block diagram of the BLA implementation, b) block diagram of the FLUA implementation.

KT-64 was captured by using VHDL, with structural description logic. The VHDL codes were synthesized for XILINX (VIRTEX) FPGA device [14], using the Xilinx ISE Design Suite v10.1 tool. Xilinx Virtex2 xc2v2000-6 FPGA was used for all the implementations of the KT-64.
In table 5, the implementation results of the KT-64 are presented. In the same table comparisons with the most widely block ciphers are given.

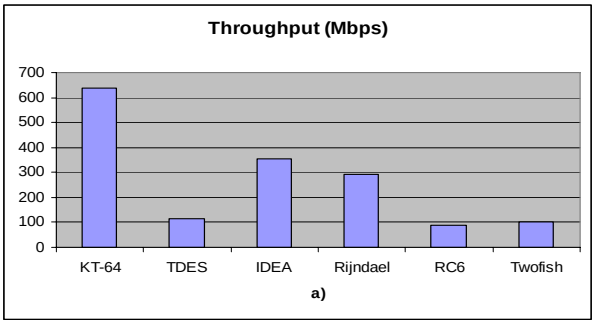
Table 5: KT-64 Implementation Results and Comparisions

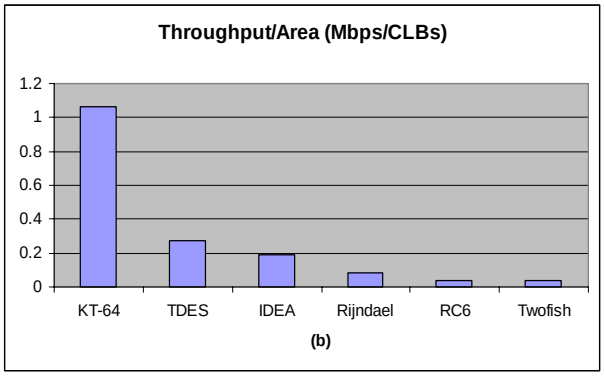
Architecture	Area (CLBs)	Frequency (MHz)	Throughput (Mbps)
KT-64_BLA Proposed	605	80	640
KT-64_FLUA Proposed	4840	94	6016
TDES_BLA [1]	431	86	115
TDES_FLUA [1]	14240	108	6900
IDEA_BLA [1]	1852	50	356
IDEA_FLUA [1]	11700	47	3008
Rijndael_BLA [15]	3528	25.3	294
RC6_BLA [15]	2638	13.8	88.5
Twofish_BLA [15]	2666	13	104

In figure 15, throughput comparisons are presented between the proposed FPGA implementations of the KT-64 and the most widely block ciphers. The comparisons are made in terms of throughput and throughput-to-area ratio requirements. The throughput results are obtained by the following equation:

$$\text{Throughput} = n.\ (\#\text{bit}/\ \#\text{cycles}).\ \text{Frequency}$$

where #bits is the number of bits at the ciphers input, #cycles is the number of clock cycles that the block cipher needs in order to encrypt/decrypt the 64-bit input and Frequency is the operation clock frequency. In the BLA implementation n is equal to 1, while in the FLUA implementation is equal to subsequent data blocks that each cipher can process in parallel. The throughput-to-area ratio reveals the hardware utilization efficiency of each implementation. It is only used in the non-feedback implementations.





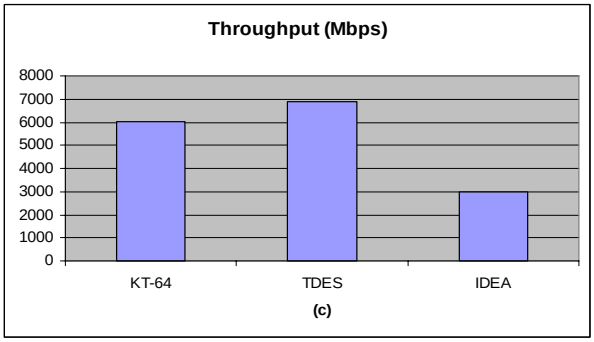


Figure 13. Throughput comparisions of the FPGA implementations: a) throughput comparisions between BLA implementations, b) throughput/area comparisions between BLA implementations, c) throughput comparisions between FLUA implementations.

The above synthesis results for implementations FPGA prove that the proposed cipher KT-64 achieves higher throughput values and covers lower area resources. In addition, it appears that its algorithmic philosophy matches better to the FPGA characteristics (due to the high Throughput/Area value).

6. Conclusion

In this paper, we propose a new fast cipher KT-64. This cipher is based on DDO transformations. Security analysis has show that the cipher is secure against know attacks.
The cipher achieve high-speed rate in FPGA device. The implementation rate and area is compared with the most widely block ciphers. These comparisons prove the suitability of the proposed cipher for efficient FPGA implementations.

References

[1] P. Kitos, N. Sklavos, M. D. Ganalis, and O. Kaufopavlou, "64-bit block ciphers: hardware implementations and comparision analysis," Computers and Electrical Engineering 30 (2004) 593–604.

[2] A. A. Moldovyan and N. A. Moldovyan, "A cipher based on data-dependent permutations", Journal of Cryptology, vol. 15, no. 1 (2002), pp. 61-72.

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

[4] N. Sklavos, A. A. Moldovyan, and O. Koufopavlou, "Encryption and Data Dependent Permutations: Implementation Cost and Performance Evaluation", proceedings of the International workshop, Methods, Models, and Architectures for Network Security, LNCS, Springer-Verlag, Berlin: 2003.

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

[6] M. A. Eremeev, A. A. Moldovyan, and N. A. Moldovyan, "Data Encryption Transformations Based on New Primitive", Avtomatika i Telemehanika (Russian Academy of Sciences) no 12 (2002), pp. 35-47.

[7] N. A. Moldovyan, A. A. Moldovyan, Data-driven Ciphers for Fast Telecommunication Systems. Auerbach Publications. Talor & Francis Group, New York, p.202, 2007.

[8] R. L. Rivest, "The RC5 encryption algorithm," in Proceedings of the 2nd International Workshop, Fast Software Encryption-FSE'94, LNCS 1008, pp. 86-96, Springer-Verlag,1995.

[9] R. L. Rivest, M. J. B. Robshaw, R. Sidney, and Y. L.Yin, "The RC6 block cipher," in 1st Advanced Encryption Standard Candidate Conference Proceedings, Venture, California, Aug. 20-22, 1998.

[10] C. Burwick, D. Coppersmith, E. D'Avingnon, R. Gennaro, Sh. Halevi, Ch. Jutla, Jr. S. M. Matyas, L. O'Connor, M. Peyravian, D. Safford, and N. Zunic, "MARS-a candidate cipher for AES," in 1st Advanced Encryption Standard Candidate Conference Proceedings, Venture, California, Aug. 20-22, 1998.

[11] R. Anderson, E. Biham, and L. Knudsen, "Serpent: a proposal for the advanced encryption standard," in 1st Advanced Encryption Standard Candidate Conference Proceedings, Venture, California, Aug. 20-22, 1998.

[12] B. Preneel et al., Comments by the NESSIE project on the AES finalists, May 24, 2000 (http://www.nist.gov/aes).

[13] E. Biham, A. Shamir, "Differential Cryptanalysis of the Data Encryption Standard," Springer-Verlag, 1993.

[14] Xilinx Inc., San Jose, California, USA, Virtex, 2.5 V Field Programmable Gate Arrays, 2003. Available from: www.xilinx.com.

[15] A. J. Elbirt, W. Yip, B. Ghetwynd, C. Paar, "FPGA implementation and performance evaluation of the AES block cipher candidate algorithm finalists," in 3rd Advanced Encryption Standard Conference Proceedings, New York, NY, USA, Apr.13-14,2000.

Nguyen Hieu Minh is a Lecturer with the Le Qui Don Technical University (Ha Noi, Viet Nam). His research interests include cryptography, communication and network security. He has authored or co-authored more than 30 scientific articles, books chapters, reports and patents, in the areas of his research. He received his Ph.D. from the Saint Petersburg Electrical Engineering University (2006).

Nguyen Thien Luan is a Lecturer with the Le Qui Don Technical University (Ha Noi, Viet Nam). His research interests include fuzzy logical, image processing, communication and network security. He has authored or co-authored more than 20 scientific articles, books chapters, reports and chaired many scientific research projects, in the areas of his research. He received his Ph.D. (1989).

Luu Hong Dung is a Lecturer with the Le Qui Don Technical University (Ha Noi, Viet Nam). His research interests include cryptography, communication and network security. He has authored or co-authored more than 15 scientific articles, books chapters, reports and patents, in the areas of his research. He received his Master from the Le Qui Don Technical University (2001).