International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Research Article

High Speed Reconfigurable Architecture for Phelix

Amol Ingole†* and Nagnath Hulle‡

†Dept. of Electronics & Telecommunication G.H. Raisoni Institute of Engineering Technology Pune, India
‡Dept. of Electronics G.H. Raisoni Collage of Engineering Nagpur, India

Accepted 01 July 2016, Available online 11 July 2016, **Vol.6, No.4 (Aug 2016)**

Abstract

Phelix is 32 bit symmetric stream cipher. It provides encryption as well as authentication with inbuilt MAC function. It is compatible with both hardware and software. It is double faster than best one AES encryption algorithm. Throughput of existing Phelix cipher was increased by replacing the existing 232 modulo ripple carry adder with modulo Carry Look ahead Adder (CLA). Proposed adder reduces critical path delay in modulo addition operation. Input given to Phelix is a 128 bit nonce (N), 256 bit key (K) and plaintext (P). It also produces a MAC tag for authentication. Key stream generated from Phelix is XORed with plaintext to produce cipher text. Proposed architecture was coded by using VHDL language and device used was Xilinx Spartan3E, XC3S500E with package FG320.

Keywords: *Authentication, Decryption, Encryption, Helix, MAC, Phelix, Stream Cipher.*

1. Introduction

Now a day's security in data transmission is real big issue. Security demands both encryption as well as authentication. Cipher technique is solution to them basically cipher is of two types block cipher and stream cipher. In a block cipher encrypt and decrypt a data block wise typically 128bits, block cipher are generally used to encrypt the data such as files, E-mail, web, text communication. Different block ciphers are DES, AES, and Serpent. While stream cipher is used to encrypt a continuous stream of data such as an audio or video transmission. Stream cipher is substitution cipher that typically uses an X-OR operation which is quickly performed by computer quickly different types of stream cipher are Helix, Phelix, Rabbit, and Salsa20.

In 2003 Helix cipher is design very much similar to a phelix cipher. Phelix is nothing but the Penta Helix one full block in helix is two half block in phelix. In 2004 Muller claim a two attack on helix cipher first one is complexity of 288 and requires 212 adaptive chosen plaintext words, but it requires a reuse of nonce. But due to that Helix may loss its security property. Later on Author of Helix cipher Paul and Bart Preneel show that above attack be overcome with chosen plaintext (CP) rather than adaptive chosen plain text (ACP) with data complexity of 235.64 CP's. main motivation and reason behind the design of phelix algorithm is mullers differential attack.

2. Literature Survey

Many of stream cipher are only efficient on either software or hardware. Phelix algorithm has been successfully passed the first Evaluation Phase ad it is implemented with MAC on both hardware and software. While LEX algorithm only efficient for software implementation. Cipher based on Simple operation i.e. Addition, Xoring, and rotation is suitable to linear and algebraic attack.

In 2005 Doug Whiting and Bruce Schneier publishing the phelix algorithm as explain in these paper. Only brute force attack is proved to be possible now. If same Key and Nonce are used to encrypt a different messages then phelix may loss it's highly security property.

In 2007 junjie Yan and Harward Heys publishes a paper on comparison of Hardware implementation two stream cipher Salsa20 and Phelix. Author conclude that Phelix is Faster than Salsa and required a less no of gate on Hardware implementation about 12,400 and its Throughput speed is 260 Mbps.

3. Phelix Cipher

Phelix use 256 bit Key. And 128 bit nonce further it is extended to 256 bit by expansion method. Phelix work on 32 bit platform. If the Key size is less than the 256 bit then zero padding is used to make it 256 bit. Two key words Xi0, Xi1 are generated by using nonce N0……N7, and working Key K0…..K7, phelix consist of no of function block in series. Each block consist of nine word of 32 bit each, which are divided in two group first one is active group consist of five active word $Z_0^i, Z_1^i, Z_2^i, Z_3^i, Z_4^i$ and second group is consist of four old word z_4^{i-4}. Active state word are updated in each block output of one block is input to another block

*Corresponding author: **Amol Ingole**

while four old state word are only used to key stream output function.

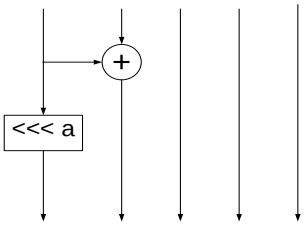

Fig.1 Typical Single Round of Phelix

A single block of phelix consist of round in helical each round of phelix consist of addition, fixed number rotation, and X-Oring serially as shown in fig.1 and that single block produces a one word of key stream to encrypt a one word of plaintext. Just after function H block phelix produces a MAC tag for authentication.

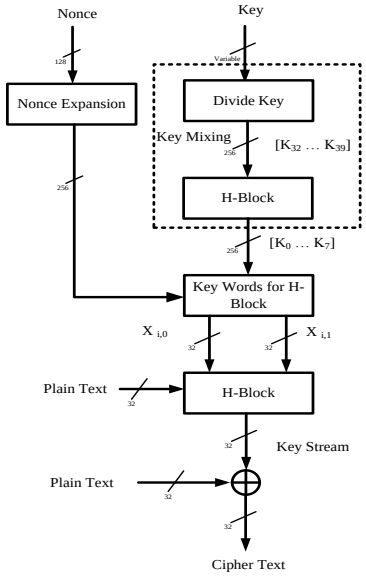

Fig.2 Typical Block Diagram of Phelix

Phelix consist of number of sequence of block serially each block assign with the unique number (i) and updated at end of block by (i+1). At a block I $Z_0^i, Z_1^i, Z_2^i, Z_3^i, Z_{4.}^i$ as shown in fig 3 are the five active words initially process through a X-Oring, Addition and fix Rotation. Two keywords Xi,0 & Xi,1 are added in each block plain text (pi) and old state word z_4^{i-4} are input to each block as shown in fig.3. At output side each block gives five new updated word $Z_0^{i+1}, Z_1^{i+2}, Z_2^{i+3}, Z_3^{i+4}, Z_{4.}^{i+5}$, and output of one block is input to another block. A single block of phelix consists of two half block of function H defines as follow. Phelix operated on 32 bit Word's platform exclusive or denoted by ⊕,Addition denoted by ⊞ and rotation denoted by <<<. A single block of phelix produces a one word of key stream by $S_i := y_4^i + z_4^{i-4}$. And that one word of keystream encrypts a single word of plain text by

$C_i := P_i \oplus S_i$.

Function $H(w0, w1, w2, w3, w4, K0, K1)$

Begin

$w0 := w0 \boxplus (w3 \oplus K0)$;	$w3 := w3 \lll 15$;
$w1 := w1 \boxplus w4$;	$w4 := w4 \lll 25$;
$w2 := w2 \oplus w0$;	$w0 := w0 \lll 9$;
$w3 := w3 \oplus w1$;	$w1 := w1 \lll 10$;
$w4 := w4 \boxplus w2$;	$w2 := w2 \lll 17$;
$w0 := w0 \boxplus (w3 \oplus K0)$;	$w3 := w3 \lll 30$;
$w1 := w1 \boxplus w4$;	$w4 := w4 \lll 13$;
$w2 := w2 \oplus w0$;	$w0 := w0 \lll 20$;
$w3 := w3 \oplus w1$;	$w1 := w1 \lll 11$;
$w4 := w4 \boxplus w2$;	$w2 := w2 \lll 5$;

Return$(w0, w1, w2, w3, w4)$;

End.

Function H block computed as follow
$(Y_0^i, Y_1^i, Y_2^i, Y_3^i, Y_4^i) := H(Z_0^i, Z_1^i, Z_2^i, Z_3^i, Z_4^i,\ 0, X_{i,0})$
$(Y_0^{i+1}, Y_1^{i+1}, Y_2^{i+1}, Y_3^{i+1}, Y_4^{i+1}) := H(Z_0^i, Z_1^i, Z_2^i, Z_3^i, Z_4^i, P_i, X_{i,1})$

Each block produces one word of key stream:
$S_i := y_4^i + Z_4^{i-4}$ (1)
And cipher text is given by
$C_i = P_I \oplus S_I$ (2)

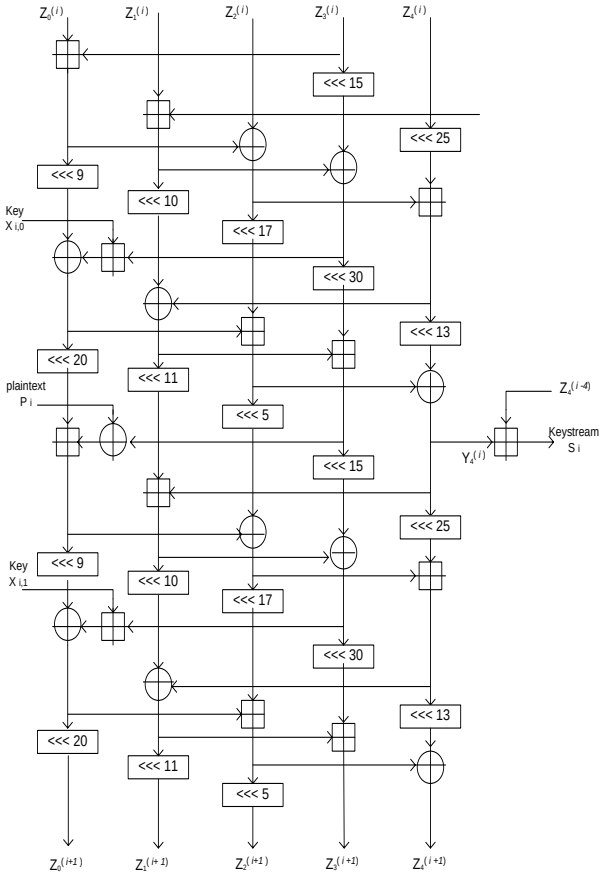

Fig.3 One H Function Block of Phelix

4. Architecture of Phelix

In proposed architecture we replace existing ripple carry adder with new carry look ahead adder, Hardware structure of phelix consist of Nonce expansion, Key mix, Sub key Generation, initialization, H function Block, Adder & multiplexers. H block is heart of phelix algorithm and adder is main part of h-block, function of each block is explain below.

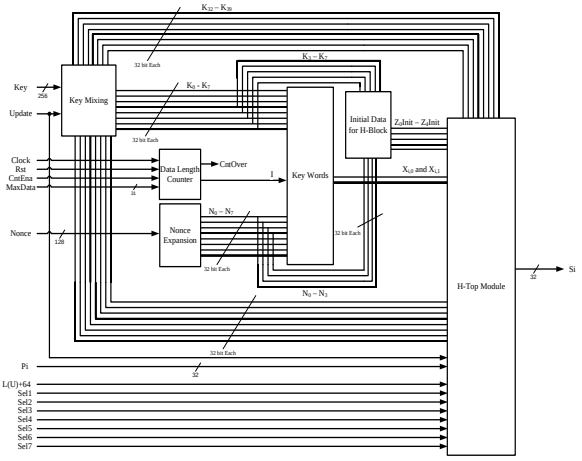

Fig.4 Proposed Architecture of Phelix

4.1 Initialization

Phelix require a eight number of block for initialization and old state word comes from previous fourth block (i-4) very first input given by

$$Z_j^{-8} := K_{j+3} \oplus N_i \quad \text{for j= 0,....,3,} \tag{3}$$

$$Z_j^{-8} := K_7 \tag{4}$$

$$Z_4^i := 0 \quad \text{for i= -12,.....,-9} \tag{5}$$

$$P_i := 0 \quad \text{for i = -8,.....,-1} \tag{6}$$

Here eight block number from -8 to -1 are used to initialization and key stream generated by this block is discarded plain text to these block is zero. Initialization sequence is shown in fig 4.

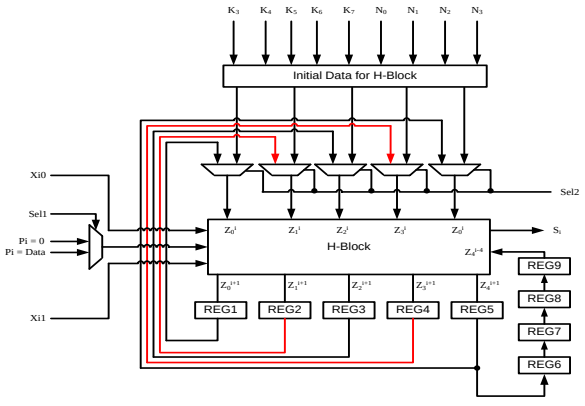

Fig.5 Initialization of H-block

4.2 Encryption

After initialization plain text is been encrypted by $k := [(l(p) + 3)/4]$ where k is the number of word in plaintext. Number of word in a plain text is k then required number of block for encryption is 0 to $k - 1$ as one block encrypt only one word of plaintext.

4.3 Computing the MAC-

As a last bit of plain text is encrypted the internal state word Z_0^k is XOred with the value 0x912d94f1.[2] then after Z_0^k word is used to post mixing of block $k, \dots \dots, k + 7$ for these block length of P_i is $l(P) mod 4$, anType equation here.d generated key stream is discarded. After a post mixing of blocks $k + 8, \dots \dots, k + 11$,using same plain text input word. These four blocks $k + 8, \dots \dots, k + 11$ generate the MAC tag.

4.4 Key mixing

the function of the key mix block is to convert a variable length key U to the fixed length working key, K. input key U is expanded with the $32 - l(U)$. Where $l(U)$ is variable length of key i.e 256 - $l(U)$ number of zero should be added to form 256 bit key K0,....., K7.

$$\left(K_{4i,\dots,}K_{4i+3}\right) := R\left(K_{4i+4,\dots}K_{4i+7}\right) \oplus \left(K_{4i+8,\dots,}K_{4i+11}\right)$$
$$\text{For } i = 7, \dots \dots, 0. \tag{7}$$

K0,..... K7 are working key and function R is define as

Function $R(w0, w1, w2, w3,)$
Begin
Local variable $w_4 := l(U) + 64;$
$(w0, w1, w2, w3, w4,) := H(w0, w1, w2, w3, w4, 0, 0):$
$(w0, w1, w2, w3, w4,) := H(w0, w1, w2, w3, w4, 0, 0):$
End.

4.5 Sub key Generation

The keywords Xi,0 & Xi,1 are generated by using 256 bit extended Nonce N0,.....,N7 and 256 bit working key K0,....,K7. Key word Xi,0 & Xi,1 are given by.

$$X_{i,0} := K_i mod\ 8 \tag{8}$$

$$X_{i,1} := K_{(i+4) mod 8} + N_{imod\ 8} + X_i' + i + 8 \tag{9}$$

$$X_i' := \begin{cases} \left[\frac{i+8}{2^{31}}\right] & if\ i\ mod 4 = 3 \\ 4.l(U) & if\ i\ mod\ 4 = 1 \\ 0 & otherwise \end{cases} \tag{10}$$

4.6 Preliminaries

Phelix work on 32 bit words platform. but the input and output are the 8 bit bytes in all situation. That conversion is done by the Xi denotes sequence of bytes Xj denotes sequence of 32 bit words.

$$X_j \coloneqq \sum_{k=3}^{3} x(4j+k)^{.2^{8k}} \quad (11)$$

$$X_i \coloneqq \left[\frac{X_{\frac{i}{4}}}{2^{8(i \bmod 4)}}\right] mod 2^8 \quad (12)$$

$$X_0 = x(3)\,\&\,x(2)\,\&\,x(1)\,\&\,x(0) \quad for\ j=0$$

$$X_1 = x(7)\,\&\,x(6)\,\&\,x(5)\,\&\,x(4) \quad for\ j=1$$

$$X_0 = x(11)\,\&\,x(10)\,\&\,x(9)\,\&\,x(8) \quad for\ j=2$$

5. Proposed CLA Adder for Phelix Cipher

A carry-look ahead adder (CLA) or fast adder is a type of adder used in digital logic. A carry-look ahead adder improves speed by reducing the amount of time required to determine carry bits. It can be contrasted with the simpler, but usually slower, ripple carry adder for which the carry bit is calculated alongside the sum bit, and each bit must wait until the previous carry has been calculated to begin calculating its own result and carry bits . The carry-look ahead adder calculates one or more carry bits before the sum, which reduces the wait time to calculate the result of the larger value bits. Carry look ahead depends on two things mainly,
Calculating, for each digit position, whether that position is going to propagate a carry if one comes in from the right. And combining these calculated values to be able to deduce quickly whether, for each group of digits, that group is going to propagate a carry that comes in from the right. Supposing that groups of 4 digits are chosen.

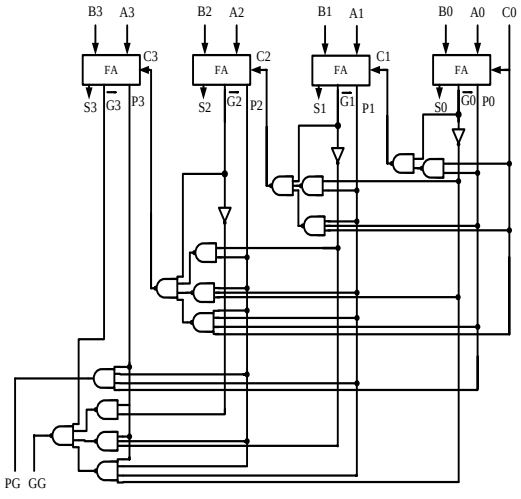

Fig.6. 4Bit CLA Adder Architecture

Phelix consist of number of addition operation. With existing modulo 232 adder it has long critical path delay. To reduce a critical path delay a new CLA Adder have been proposed. It is modified and fastest adder among the days.

It has improved Efficiency & cost. Basically CLA is 4 bit Adder as shown in fig. 5 and by cascading it will expand up to the requirement.

5. Comparison between existing & proposed architecture

Ripple carry adder delay is given by 2n+1 where n is no of bit in addition, phelix work on 32 bit platform so the total gate delay is 65 and typical delay of one gate is 10ns, so total gate delay is 650ns, in phelix one H-block have 13 number of adder, so total delay due to adder of h-block is 8450ns.
Carry look ahead adder has 4 gate delay only so total gate delay is 12 by cascading for 32 bit, similarly one gate has 10ns of delay, so total delay is 120ns. One h-block have 13 adder, thus total delay of h-block due to adder is 1560ns.

Phelix algorithm execute number of h-block operation for encryption as well decryption so with the help of proposed architecture we can reduce time delay and increased the speed of algorithm.

Table 1 Comparison between existing & proposed architecture

Ripple Carry Adder	Carry Look A Head Adder
Delay of adder is given by (2n+1) where n= no of bit in addition.	For each layer there is 4 gate delay.
for 32 bit (2x32+1)=65 gate delay	Total delay = 12 Gate delay (4bit+16+1bit+32bit)
Typically one gate has a delay of 10 ns.	One gate has a delay of 10 ns.
Total Delay 65 x 10 =650 ns.	Total delay = 12 x 10 =120 ns.
One H-block consist of total no of 13 adder.	One h-block consist of 13 adder.
Total delay of one block due to adder is 650 x 13 = 8450 ns.	Total delay of one block due to adder is 12 0 x 13 = 1560 ns.

Conclusions

Proposed Phelix architecture replaces existing modulo ripple carry adder by modulo CLA which is Fastest Adder now a days with improved efficiency and cost. Proposed architecture uses more area as compared to the existing architecture. The use of CLA minimizes the critical path delay and in the implementations and increases throughput of proposed architecture as compared with previous implementations.

References

D. Whiting, B. Schneier, and S. Lucks (May 2005) Phelix - Fast Encryption and Authentication in a Single Cryptographic Primitive presented at Symmetric Key Encryption Workshop, Aarhus, Denmark.

D. Whiting, Niels Ferguson, B. Schneier, and S. Lucks (May 2003) Helix - Fast Encryption and Authentication in a Single Cryptographic Primitive presented at Symmetric Key Encryption Workshop, Aarhus, Denmark.

Junjie Yan and Howard M. Heys Hardware Implementation of the Salsa20 and Phelix Stream Ciphers

Meltem S¨onmez Turan, Ali Do˘ganaksoy, C¸ a˘gda¸s C¸ alık Statistical Analysis of Synchronous Stream Ciphers

Yu-Ting Pai and Yu-Kumg Chen The Fastest Carry Look ahead Adder Proceedings of the Second IEEE International Workshop on Electronic Design, Test and Applications (DELTA'04)

Khaled M. Suwais Comprehensive survey on constructional design of existing stream cipher