ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

Reduction of Area Consumption in FPGA for Blake Algorithm with Side Channel Attack Resistant

Mr. M.Jothi Kumar[1] (M.E)., Mrs.Chitravalavan[2] M.tech(Ph.D).,
PG student,Applied Electronics, AVC College of Engineering, Mayiladuthurai, India [1]
Head of the Department, ECE, AVC College of Engineering, Mayiladuthurai, India [2]

Abstract: This paper proposes the Pipelined SHA-3 BLAKE algorithm with Side-Channel Attack resistant, running on an FPGA with the intention of developing the optimization in FPGA for BLAKE algorithm. The National Institute of Standards and Technology (NIST) published Secured hash algorithm-3(SHA-3) BLAKE algorithm which replaces SHA-0, SHA-1, SHA-2.To implement BLAKE algorithm we have utilized VHDL, where we introduce the pipelining. Pipelining improves the efficiency of the algorithm by executing single task per clock cycle. Side Channel Attack is any attack based on information gained from the timing, power, etc to decode key bits of the algorithm. The simulation is done with the usage of modelsim and synthesis it on FPGA Spartan 3E using Xilinx software. Our VHDL implementation executed on BLAKE algorithm occupies 30% percentage of FPGA's area.

Keywords: Cryptography; SHA-3 finalist; BLAKE algorithm; Pipelining; Side-Channel Attack; FPGA; VHDL

I. INTRODUCTION

Cryptography is the practice and study of techniques for secure communication in the presence of third parties (i.e.) encrypts the message using the secret key, that message is known only by sender and receiver not by the third party. Encryption is the transformation of plaintext to cipher text based on encryption algorithm using the secret key. If both sender and receiver use the same key, then it is called as symmetric key encryption and if they use the different key is called as asymmetric key encryption. All encryption algorithms are based on substitution and transposition. In plain text each element is mapped into another element is called as substitution and elements in the plaintext are rearranged in the transposition.

The National Institute of Standards and Technology (NIST) announced a global competition to find a new Secured Hash Algorithm (SHA) function to replace SHA-0, SHA-1 and SHA-2 in 2007. BLAKE is a cryptographic hash function submitted to the NIST hash function competition by Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and RaphaelC.W.Phan. BLAKE was chosen as one of the five finalists in SHA-3 competition announced by NIST in 2010. It is implemented using pipelining which executes single task per clock cycle instead all task in one clock cycle..BLAKE algorithm has a family of four hash functions, for instance: BLAKE-224, BLAKE-256, BLAKE-384 and BLAKE-512, with different sizes of output. Here, we have implemented BLAKE-256 using VHDL in FPGA Spartan 3E.

VHDL (VHSIC Hardware Description Language) is a language for describing hardware from abstract to concrete level. Firstly it describes the structure of a design that is how that structure of a design can be decomposed into sub-design and how they are interconnected. Secondly, it allows describing the function specifications of a design using programming language forms. Thirdly, as a result, it allows simulating the design before manufacturing. So the designer can quickly compare the alternatives and test for correctness.

FPGA (Field Programmable Gate array) is an integrated circuit which can be reconfigured by a customer or designer after manufacturing, hence it is named as Field Programmable. FPGA contains Configurable Logic Blocks (CLB) with Programmable Interconnect. Configurable Logic Blocks are called as slices and each slice contains 2 Flip Flops (FF), and 2 four input Lookup Tables (LUTs).

II. EXISTING SYSTEM

The existing system presents the analysis of assembly instructions from SHA-3 BLAKE algorithm, running on an ARM processor, with the intention of developing a specific processor in FPGA for BLAKE Algorithm. Moreover,

All Rights Reserved © 2015 IJARTET

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

VHDL has been utilized here. The Synthesis and simulations of the ALU and UC were done with the usage of Altera Quartus II 9.1. The instruction set for the ALU for BLAKE-256 algorithm is based on the instructions from the ARM processor which is not reconfigurable.

The implementation of a control unit and an ALU, based on set of assembly instructions from the ARM (Advanced RISC Machine) processor for BLAKE algorithm (256 version), which will be implemented using VHDL (Very High Speed Integrated Hardware Description Languages) and FPGA (Field Programmable Devices). The assembly instructions for BLAKE algorithm can be obtained from the simple scalar simulation tool. This VHDL implementation executed on BLAKE algorithm occupied 43% of the FPGA's area.

III. PROPOSED SYSTEM

The Proposed system presents the implementation of SHA-3 BLAKE algorithm (version 256) in FPGA using pipelining. This Pipelined SHA-3 BLAKE algorithm simulated using modelsim and synthesis using Xilinx on FPGA Spartan 3E.

A. *Blake-256 Algorithm*

BLAKE-256 algorithm is one among the family of four hash functions and it is a 32 bit version. BLAKE-256 produces a 256-bit hash which is treated as eight 32-bit word. Table I represents the some important terminologies involved in the BLAKE-256 algorithm.

TABLE I
BLAKE-256 ALGORITHM TERMINOLOGIES

Symbol	Meaning
←	Variable assignment
+	Sum(module)
>>>k	K-bit rotation to the right
<<<k	K-bit rotation to the left
⊕	Exclusive-OR(Ex-or)

BLAKE algorithm performs both encryption and compression function. The compression function of the BLAKE-256 algorithm involves three steps: i) Initialization ii) Round Function iii) Finalization.

B. *Initialization*

The initial state of the compression function is treated as a 4x4 matrix as shown in Fig. 1. This state is initialized by the current hash (h), salt value(s), timer value (t), and a 256-bit constant (c).

$$\begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 & v_7 \\ v_8 & v_9 & v_{10} & v_{11} \\ v_{12} & v_{13} & v_{14} & v_{15} \end{pmatrix} \leftarrow \begin{pmatrix} h_0 & h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 & h_7 \\ s_0 \oplus c_0 & s_1 \oplus c_1 & s_2 \oplus c_2 & s_3 \oplus c_3 \\ t_0 \oplus c_4 & t_0 \oplus c_5 & t_1 \oplus c_6 & t_1 \oplus c_7 \end{pmatrix}$$

Fig. 1. Initial state of the compression function

C. *Round Function*

Once the matrix is initialized, next step is round function which is iteratively processed through 14 rounds. Each round consists of eight state transformations labelled as G0 through G7. Each operates on and modifies only 4 of the 16 state words, generalized as a, b, c, and d. The visual representation of G transformation is shown in Fig. 2. The transformations consisting of addition, bit-rotations, exclusive or (XOR) operations.

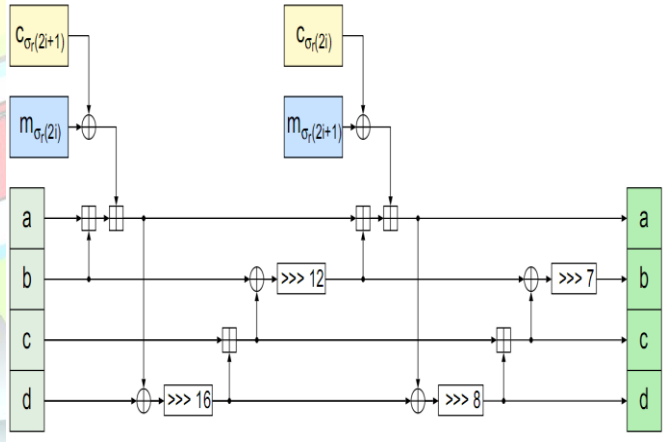

Fig. 2. Visualized G transformation

All Rights Reserved © 2015 IJARTET

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

$$
\begin{aligned}
a &\leftarrow a + b + (m_{\sigma_r(2i)} \oplus c_{\sigma_r(2i+1)}) \\
d &\leftarrow (d \oplus a) \ggg 16 \\
c &\leftarrow c + d \\
b &\leftarrow (b \oplus c) \ggg 12 \\
a &\leftarrow a + b + (m_{\sigma_r(2i+1)} \oplus c_{\sigma_r(2i)}) \\
d &\leftarrow (d \oplus a) \ggg 8 \\
c &\leftarrow c + d \\
b &\leftarrow (b \oplus c) \ggg 7
\end{aligned}
$$

Fig. 3. Generalized G transformation

The generalized form of G transformation is shown in Fig. 3. The first four steps are performed on independent columns in parallel and last four steps are performed on independent diagonals in parallel as well. The parallelized

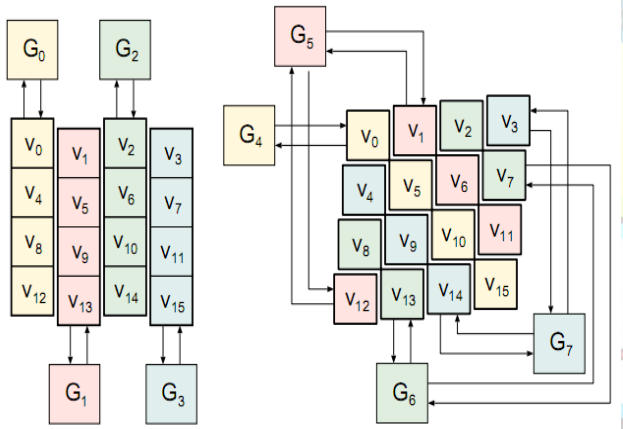

Fig. 4. Parallelized column and diagonal step

column and diagonal step is shown in Fig. 4. For one round, eight (4 column step [G0 to G3] and 4[G4 to G7] diagonal step) G function can be called and totally 112 G function can be called for 14 rounds.

D. Finalization

After fourteen rounds of G transformation,compression function performs finalization step. The final step in compression function is shown in Fig. 5. It produces new hash values (h'0 to h'7) by performing EX-OR operation on

$$
\begin{aligned}
h'_0 &\leftarrow h_0 \oplus s_0 \oplus v_0 \oplus v_8 \\
h'_1 &\leftarrow h_1 \oplus s_1 \oplus v_1 \oplus v_9 \\
h'_2 &\leftarrow h_2 \oplus s_2 \oplus v_2 \oplus v_{10} \\
h'_3 &\leftarrow h_3 \oplus s_3 \oplus v_3 \oplus v_{11} \\
h'_4 &\leftarrow h_4 \oplus s_0 \oplus v_4 \oplus v_{12} \\
h'_5 &\leftarrow h_5 \oplus s_1 \oplus v_5 \oplus v_{13} \\
h'_6 &\leftarrow h_6 \oplus s_2 \oplus v_6 \oplus v_{14} \\
h'_7 &\leftarrow h_7 \oplus s_3 \oplus v_7 \oplus v_{15}
\end{aligned}
$$

Fig. 5. Final step in compression function

sequence of new values (v0 to v15) with initial hash value (h0 to h7) and salt value (s0 to s3).

IV. SIDE CHANNEL ATTACKS

"Side channel attacks" are attacks based on "side channel information" that can be retrieved from the encryption device to recover the secret key that device is using. Side Channel attacks can be classified as Timing attack (based on the time that operations of the Encryption device take), power analysis attack (based on analyzing power consumption of hardware during computation), Electromagnetic attack (based on leaked electromagnetic radiation) and more. In this paper, the technique to overcome the Power analysis attack among the Side channel attacks in FPGA for BLAKE algorithm is implemented.

A. Power Analysis Attack

The power analysis attack recovers the secret key of Encryption device by observing power consumption of that device. An Example for Power Analysis Attack of Smart card reader is shown in Fig. 6.Here, Cryptographic device (Smart card reader) consumes some power while swiping a card, then the Power consumption can be observed by the oscilloscope. This observed information can be decoded into

All Rights Reserved © 2015 IJARTET

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

Fig. 6.Power Analysis Attack

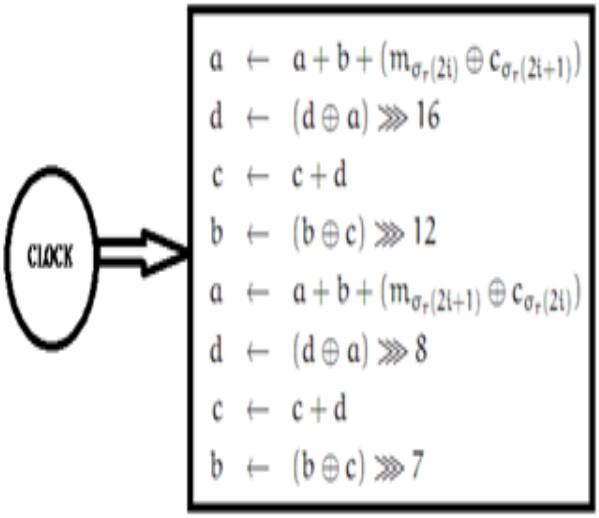

Fig. 7. BLAKE algorithm with Power Analysis Attack resistant

computer as binary value (0's and 1's) to recover the secret key of the Cryptographic device.

The Power Analysis Attack can be categorized as Simple Power Analysis (SPA) Attacks and Differential Power Analysis (DPA) Attacks. Simple power Analysis is based on observing visual representation of the power consumption of an Encryption Device whereas Differential Power Analysis involves not only visual representation but also statistical analysis and error-correction statistical methods, to obtain information about the secret key.

B.BLAKE algorithm with Power Analysis Attack resistant

This Paper presents the implementation of BLAKE algorithm in FPGA with Differential Power Analysis Attack resistant. This can be done by making use of single clock to run the operations of BLAKE algorithm in FPGA as shown in Fig. 7. The main operation of the BLAKE algorithm takes place in G transformation as explained in section III(C).In this G transformation, single clock for whole steps is used instead of single clock for each step of operation. It is difficult to analyze the power variations for whole operations in single clock cycle. The Differential power analysis attack can also overcome by hiding the hash values in registers called as register hiding. These make the attacker not to easily analyze the power consumption of BLAKE algorithm while executing in FPGA.

V. IMPLEMENTATION ARCHITECTURE

In this paper, BLAKE-256 algorithm is implemented using the pipelining. The basic structure of implementing BLAKE-256 algorithm in FPGA using VHDL is shown in Fig. 8. First, set the initial hash and constant value and using the iteration matrix send the A, B, C, D input for G function. Then, set the A, B, C, D output from G function (i.e.) values in the column and diagonal are updated by fourteen rounds of G function call.

The initial hash values H_A_in, H_B_in, H_C_in, H_D_in are set to the BLAKE and the output hash values H_A_out,H_B_out, H_C_out, H_D_out from the BLAKE are sent as the G Function input (A_in,B_in,C_in,D_in) as shown in Fig.8. After performing G function (column and diagonal step), output of the G function (A_out,B_out,C_out,D_out) are set as the new hash value

All Rights Reserved © 2015 IJARTET

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

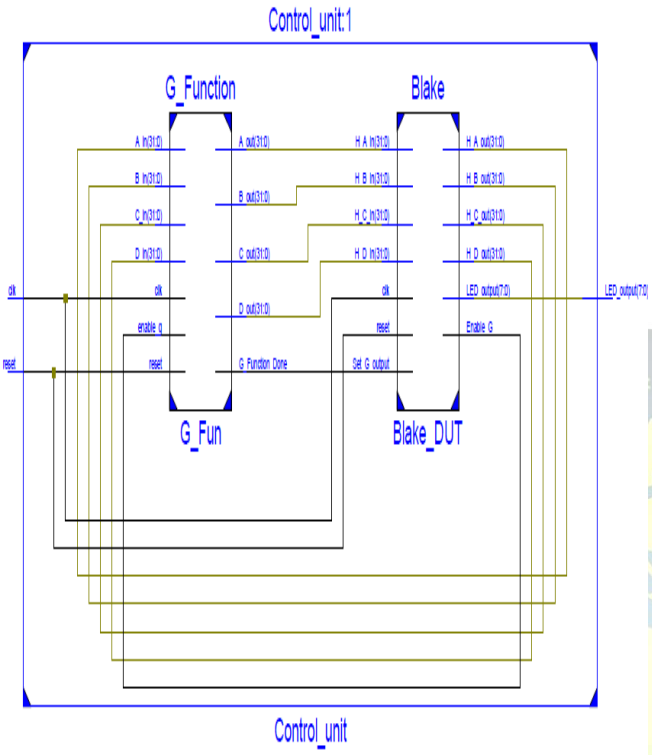

Fig. 8. Blake algorithm implementation

for BLAKE. This step is repeated continuously for 14 times (14 rounds) and finally gets the new hash value from the BLAKE.

VI. SIMULATION AND SYNTHESIS RESULTS

The simulation result of implementing BLAKE algorithm in VHDL using modelsim is shown in Fig. 9. The synthesis report was got by synthesizing pipelined BLAKE algorithm on FPGA Spartan 3E using Xilinx is shown in Table II

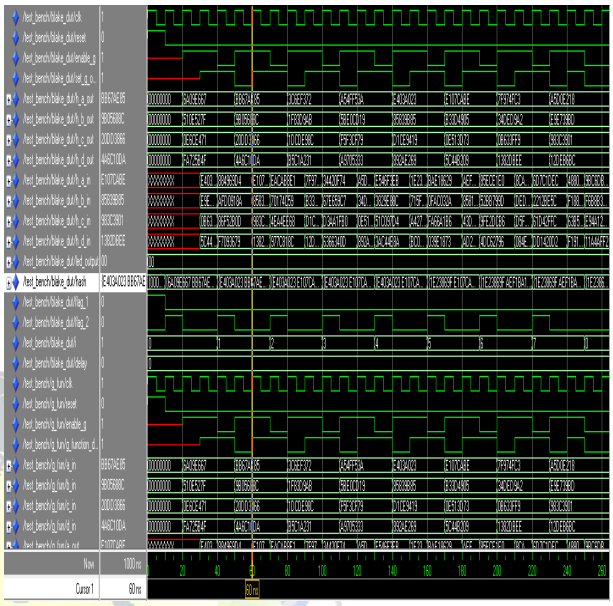

Fig. 9. Simulation result

The FPGA Spartan 3E consists of 4656 slices and each slice has 2 Flip Flops and 2 LUTs. BLAKE algorithm occupies 1018(10%) out of 9312 Flip Flops and 2779(29%) out of

Table II
SYNTHESIS REPORT

Number of Slices	1443 out of 4656	30%
Number of Slice Flip Flops	1018 out of 9312	10%
Number of 4 input LUTs	2779 out of 9312	29%
Number of IOs	10	-
Number of bonded IOBs	10 out of 232	4%
Number of BRAMs	2 out of 20	10%

9312 LUTs in FPGA. Therefore, it totally occupies 1443(30%) slices out of 4656 (i.e.) 30% of the FPGA's area.

All Rights Reserved © 2015 IJARTET

ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com

International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 2, Issue 4, April 2015

VII. CONCLUSION

BLAKE algorithm is one of the security algorithms, we considered BLAKE due to its accomplishment on being one of the 5 finalist algorithms of competition prompted by NIST. BLAKE algorithm is so simple, fast in both hardware and software. The implementation of this algorithm in FPGA Spartan 3E using pipelining occupied 30% of the FPGA's area.

Here, the BLAKE algorithm with side channel attack resistant in FPGA has been done up to the development level. For the future, I intend to test it using Side-channel Attack Standard Evaluation Board (SASEBO).

REFRENCES

[1]. J-P. Aumasson, L. Henzen, and R. C. W. Phan, "SHA-3 proposal BLAKE*" , *Report NIST*, December 16, 2010.

[2]. A. Regenscheid, R. Perlner, S-J. Chang, J. Kelsey, M. Nandi, and S.Paul, "Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Competition", NIST, September 2009.

[3]. B. Baldwin, and W. P. Marnane, "Yet Another SHA-3 Round 3 FPGA Results Paper". IACR Cryptology ePrint, v. 2012, p. 180.

[4]. S. Kerckhof, F. Durvaux, N. Veyrat-Charvillon, F. Regazzoni, G. M.de Dormale, and F-X. Standaert, "Compact FPGA Implementations of the Five SHA-3 Finalists". In Proceeding 10th IFIP WG 8.8/11.2 International Conference - Smart Card Research and Advanced Applications (CARDIS 2011), Leuven, Belgium, pages 217-233, September 2011.

[5]. J-P. Kaps, P. Yalla, K. K. Surapathi, B. Habib, S. Vadlamudi, S.Gurung, and J. Pham, "Lightweight Implementations of SHA-3 Candidateson FPGAs". In Proceeding 12th International Conference on Cryptology in India (INDOCRYPT 2011), Chennai, India, pages 270-289, December 2011.

[6]. N. Sklavos, and P. Kitsos, "BLAKE HASH Function Family on FPGA: From the Fastest to the Smallest". In Proceeding of IEEE Computer Society Annual Symposium on VLSI (ISVLSI'2010), Lixouri Kefalonia, Greece, pages139-142,July2010.

[7]. Cryptography and Network Security Principles and Practices,4th edition by William Stallings, November 16, 2005

[8]. Pereira V.F, Moreno E.D, Dias W.R.A , q D.O.D, "Specific processor in FPGA for Blake Algorithm" ,Circuits and Systems (LASCAS), March 2013.

All Rights Reserved © 2015 IJARTET