International Conference on Recent Advances in Management, Architecture, Technology and Engineering
(ICRAMATE'16)
Organized by
International Journal of Advanced Research in Management, Architecture, Technology and Engineering
(IJARMATE)
21st March 2016

Parallel RC4 Key Searching System Based on FPGA

Muthamil Jothi.M[1], M.Priya[2], V.Manjula[3], Christo Ananth[4]
U.G. Scholars, Department of ECE, Francis Xavier Engineering College, Tirunelveli[1, 2, 3]
Associate Professor, Department of ECE, Francis Xavier Engineering College, Tirunelveli[4]

Abstract—this paper presents an effective field-programmable gate array (FPGA)-based hardware implementation of a parallel key searching system for the brute-force attack on RC4 encryption. The design employs several novel key scheduling techniques to minimize the total number of cycles for each key search and uses on-chip memories of the FPGA to maximize the number of key searching units per chip. Based on the design, a total of 176 RC4 key searching units can be implemented in a single Xilinx XC2VP20-5 FPGA chip. Operating at a 47-MHz clock rate, the design can achieve a key searching speed of 1.07 x 10^7 keys per second. Breaking a 40-bit RC4 encryption only requires around 28.5 h.

Keywords—Brute-force attack, field-programmable gate array (FPGA), RC4 encryption.

I. INTRODUCTION

RC4 has been the most popular and powerful stream cipher since its invention in 1987 . To break an RC4 encryption requires either guessing the internal state of the cipher or brute-force search of the whole key space . In an RC4 encryption using an 8-bit word, the total number of possible internal state is 256!.

Tews *et al.* demonstrated a practical key recovery attack on WEP which is based on partial key exposure vulnerability in the RC4 stream cipher used in the WEP protocol. They presented a WEP attack implemented in software which can recover a 104-bit WEP key in less than 60 s. However, this kind of technique cannot be applied to other RC4 encryptions which do not use the same flawed session key generation scheme as that in the WEP protocol. Christo Ananth et al. [1] proposed a system which can achieve a higher throughput and higher energy efficiency. The S-BOX is designed by using Advanced Encryption Standard (AES). The AES is a symmetric key standard for encryption and decryption of blocks of data. In encryption, the AES accepts a plaintext input, which is limited to 128 bits, and a key that can be specified to be 128 bits to generate the Cipher text.

In decryption, the cipher text is converted to original one. By using this AES technique the original text is highly secured and the information is not broken by the intruder. From that, the design of S-BOX is used to protect the message and also achieve a high throughput, high energy efficiency and occupy less area. Christo Ananth et al. [2] proposed a system which contributes the complex parallelism mechanism to protect the information by using Advanced Encryption Standard (AES) Technique. AES is an encryption algorithm which uses 128 bit as a data and generates a secured data. In Encryption, when cipher key is inserted, the plain text is converted into cipher text by using complex parallelism. Similarly, in decryption, the cipher text is converted into original one by removing a cipher

All Rights Reserved © 2015 ICRAMATE16

key. The complex parallelism technique involves the process of Substitution Byte, Shift Row, Mix Column and Add Round Key. The above four techniques are used to involve the process of shuffling the message. The complex parallelism is highly secured and the information is not broken by any other intruder. Christo Ananth et al. [3] proposed a system in which the complex parallelism technique is used to involve the processing of Substitution Byte, Shift Row, Mix Column and Add Round Key. Using S- Box complex parallelism, the original text is converted into cipher text. From that, we have achieved a 96% energy efficiency in Complex Parallelism Encryption technique and recovering the delay 232 ns. The complex parallelism that merge with parallel mix column and the one task one processor techniques are used. In future, Complex Parallelism single loop technique is used for recovering the original message.

In this paper, we present a highly optimized FPGA-based massively parallel RC4 key searching system, integrating 176 key searching units on a single Xilinx XC2VP20-5 chip. Our proposed implementation scheme achieves a significant speedup in the key searching process because of the following novel implementation techniques.

• All the key searching units are synchronized so that many of the system components can be shared among all the units, reducing the total resource usage.

• An innovative architecture for implementing the key scheduling process is used so that the total number of clock cycles for searching one key can be reduced to 772, compared to 792.

• The dual-port simultaneous read/write feature of the Block RAMs in the FPGA chip is fully exploited so that one Block RAM can be used to implement two key searching units, which outperforms all other proposed implementations. Christo Ananth et al. [5] proposed a system which contributes the complex parallelism mechanism to protect the information by using Advanced Encryption Standard (AES) Technique. AES is an encryption algorithm which uses 128 bit as a data and generates a secured data. In Encryption, when cipher key is inserted, the plain text is converted into cipher text by using complex parallelism. Similarly, in decryption, the cipher text is converted into original one by removing a cipher key. The complex parallelism technique involves the process of Substitution Byte, Shift Row, Mix Column and Add Round Key. The above four techniques are used to involve the process of shuffling the message. The complex parallelism is highly secured and the information is not broken by any other intruder.

We implement a massively parallel RC4 key searching system that breaks a 40-bit RC4 encryption in 28.5 h. The rest of the paper is organized as follows. The RC4 encryption algorithm is described in Section II, followed by our RC4 key searching system architecture in Section III. Section IV describes the implementation details and results. We discuss and conclude our findings in Sections V and VI.

II. RC4 ENCRYPTION

A. Algorithm

RC4 is a binary additive stream cipher. It uses a variable-sized key ranging from 8 to 2048 bits in multiples of 8 bits. It is actually a class of algorithms parameterized on the size of its block. The parameter, *n*,is the word size for the algorithm. In most applications, *n* is chosen to be 8. The internal state of RC4 consists of a table, S-box (known as S), of size 2^n words and two word-sized counters *I* and *j*.

All Rights Reserved © 2015 ICRAMATE16

The core of RC4 encryption consists of two functions: key schedule algorithm (KSA), which is responsible for the initialization of a key dependent permutation in S, and key stream generator which generates a sequence of bits that can be XOR with the plaintext for encryption and with the cipher text for decryption.

1) Key Schedule Algorithm: It consists of two phases: initialization phase in which S is set to the identity permutation, and scrambling phase in which it uses a key (K) with L bytes long to continuously swap values of S to produce new unknown key-dependent permutations. As the only action on S is value swapping, the fact that S contains a permutation is always maintained. Algorithm details are as follows.

Initialization phase:

j=0,s[i]=i A i [0,2^n-1]

Scrambling phase:

j=j+s[i]+k[i mod L],swap(s[i],s[j] A i €[0,2^n-1]

2) Key stream Generator: It continuously shuffles the permutation stored in S and picks up a different value from the S permutation as output. One round of the cipher outputs an n-bit word as the key stream. Algorithm details are:

Initialization phase:

i=0,j=0

Key stream generation loop:

i=i+1,j=j+S[i],swap(S[i],S[j])
t=s[i]+s[j],output s[t]

B. Characteristics

Based on the above key scheduling and key stream generation algorithms, RC4 has the following characteristics.

• S starts as an identity permutation of the 2^n possible words and remains a permutation throughout.

• Since S is always a permutation, the state just holds $\log_2 (2^n!)+2n \approx 1700$ bits of information .

• Knowing the internal state of the cipher at a given time is sufficient to predict all the key stream bits in the future and so to break the cipher.

• As the permutation of S depends solely on K, knowing the latter can break the cipher.

• The period of the output key stream depends only on K, which is normally very long and hard to predict.

• In many commercial applications, n=8 and L=5,(40 bits).Thus, to break an RC4 encryption, it is more effective to find K by brute-force attack on the entire key space instead of finding the internal state of the cipher.

III. PROPOSED TECHNIQUES

A. RC4 Key Searching System

Our proposed RC4 key searching system consists of multiple key searching engines which can be implemented in a single FPGA chip to carry out massively parallel key search of an RC4 encryption. Each key searching engine consists of multiple key searching units.

B. Operations for Testing One Key

To test a new key, key scheduling and key stream generation must be performed. For the former, 256 clock cycles are required to initialize the S-box, and then 256 iterations are required to scramble it. In each iteration, four clock cycles are required (reading S[i], calculating j and reading

All Rights Reserved © 2015 ICRAMATE16

S[j], writing S[i] and writing S[j]). For the latter, generating a byte of key stream requires five clock cycles (reading S[i], calculating j and reading S[j], writing S[i], writing S[j], and calculating t, reading S[t] and then outputting the result). After getting the first byte from the key searching unit, the output data is checked against the first byte of the chosen cipher text. If they do not match, the key being tested is incorrect and the next key testing should start. Otherwise, the above iteration is repeated until the first five bytes output from the key stream generator match with the corresponding bytes of the chosen cipher text, in which the key of the RC4 encryption is found. If the key being tested is incorrect,the expected value of the number of iterations in the key stream generation process is approximately one. Thus, the total number of cycles required for testing one key is 256 + 256 x 4 + 5 x 1=1285 cycles.

C. Reducing the Number of Clock Cycles for Testing One Key

1) Using "Read-Before-Write" Access Mode: As both Distributed RAM and Block RAM in modern FPGA chips support "read-beforewrite" access mode, we can use it to read S[j] and write S[i] in a single cycle, reducing the total number of clock cycles for testing one key to256 +256 x 3 +4 x1=1028 cycles.

2) Using 16-Bit Wide On-Chip Block RAM: A S-Box can be implemented using a 256 block of 8-bit wide Distribution RAM or Block RAM. However, if 16-bit wide RAM is used instead, the S-box initialization can be embedded in the S-box scrambling process. The mechanism is to divide the 16-bit wide RAM into two halves (S_H and S_L) via the high-order byte and the low-order byte of the data bus. For testing a new key, only one half of the RAM is used for the key scheduling and key stream generation, the other half is initialized

to the identity permutation for the next key testing. The detailed scheme is as follows.

• In key scheduling: scramble $S_H[i]$ and write $S_L[i]$ with i.

In key stream generation: calculate and output $S_H[t]$ using $S_H[i]$ and $S_H[j]$and write $S_L[i]$ with i.

• In testing the next key: swap the use of S_H and S_L.

As a result, 256 clock cycles used for the initialization of the S-box can be saved, reducing the total number of clock cycles for testing one key to: 256 x 3 + 4x 1=772 cycles.

D. One Key Searching Unit Tests Two Keys Simultaneously

The Block RAM can be used to implement a true dual-port RAM. If we use 512-byte Block RAM and divide it into two halves ($S_A$and S_B) via the most significant bit of the address, one key searching unit can use these two halves as two independent S-boxes for testing two keys simultaneously. For example, a single key searching unit can use S_A to test a key, K, and use S_B to test another key, K+1, simultaneously.

E. Key Search

40-bit test keys in the whole 40-bit key space are sequentially fitted into the key searching units to search for the correct key. To generate the test keys, one key searching engine uses a 40-bit counter, K, shared among all the key searching units inside the engine. In each unit, an 8-bit register K5 is used to store the last byte of the test key. The total number of key searching units in a key searching engine, n, is then restricted to be of form of 2^m where $m \le 8$.

All Rights Reserved © 2015 ICRAMATE16

Thus, only one 40-bit counter and n 8-bit registers are required. In comparison, the straightforward key generation approach requires n 40-bit counters.

IV. IMPLEMENTATION DETAILS AND RESULTS

The proposed RC4 key searching systems using different techniques described earlier were synthesized and implemented in VHDL codes using Xilinx ISE 8.1i software development tool. The FPGA platform used was a Xilinx HW-AFX-FF1152-300 FPGA Development Board which consists of a Virtex-II Pro XC2VP20-5 FPGA. A Chip- ScopePro 8.1.03i software tool was used for probing signals in the FPGA platform during testing.

A. *Main Framework*

The system consists of M key searching engines as shown in above Fig.. Each key searching engine consists of n key searching units. By passing test keys (start_keys) in different key ranges to the key searching engines, searching the whole key space can be carried out in parallel.

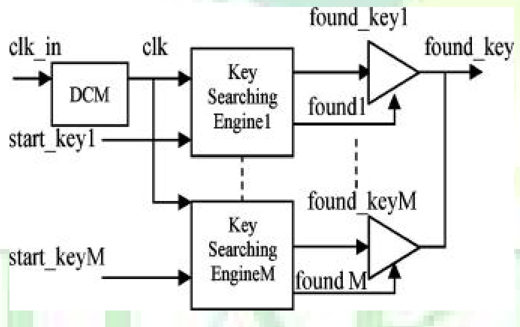

Fig1.Schematic of the proposed RC4 key searching system

B. *Implementation Schemes and Their Results*

We developed three different implementation schemes. The total amount of FPGA resources used and the maximum system clock frequency in the three schemes were compared to find out the most optimal and effective scheme for implementing an RC4 key searching system.

Scheme 1: Distributed RAM was used to implement the S-Boxes and the technique described in Section III-C-1 was used in the KSA process. However, the techniques described in Section III-C-2 and III-D cannot be used as Distributed RAM is not a true dual-port RAM. According to equation in Section IV-C, maximizing _ can reduce the overhead in the resource usage. Because of the limited slice resources in the FPGA chip, the largest size for the 1st key searching engine is 64. Following the same rule, we implemented three more key searching engines of size of 8, 4, and 1 to fully utilize all the on-chip resources.

Scheme 2: Block RAM with 8-bit bus width and 512 data depth was used to implement the S-Boxes and the technique described in Section III-D was employed to test two keys simultaneously by one key searching unit. We implemented three parallel RC4 key searching engines with sizes of 64, 16, and 8. As there were slices and TBufs left on the chip after implementing the three engines, we implemented three more key searching engines of sizes of 16, 8, and 4 using Distributed RAM.

Scheme 3: A Block RAM with 16-bit bus width and 512 data depth was used to implement two S-boxes for each key searching unit as described in Section III-D and the technique described in Section III-C2was employed to reduce the total clock cycles for testing one key to 772.

All Rights Reserved © 2015 ICRAMATE16

We implemented three key searching engines with size of 64, 16, and 8.

Table1
RC4 IMPLEMENTATIONS COMPARISON

Implementation scheme	**Scheme1**	**Scheme2**
No of key searching unit	90	204
CLB Slices Used	9107	9245
Block RAM used	0	88

Performance comparison among the three proposed implementation schemes is summarized in Table.

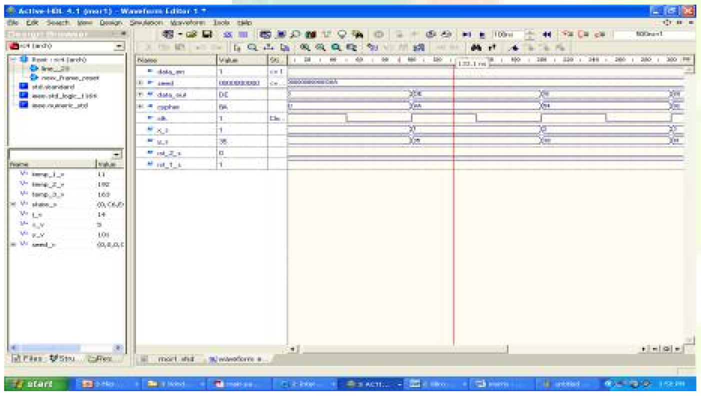

Fig2 Simulation output

V. DISCUSSION

A. Using Distributed RAM and Block RAM

We show from the implementation that S-boxes used in the key searching units can be built by using Distributed RAM or Block RAM.As both support "read-then-write" access mode , Christo Ananth et al. [4] proposed a system, Low Voltage Differential Signaling (LVDS) is a way to communicate data using a very low voltage swing (about 350mV) differentially over two PCB traces. It deals about the analysis and design of a low power, low noise and high speed comparator for a high performance Low Voltage Differential Signaling (LVDS) Receiver. The circuit of a Conventional Double Tail Latch Type Comparator is modified for the purpose of low-power and low noise operation even in small supply voltages. The circuit is simulated with 2V DC supply voltage, 350mV 500MHz sinusoidal input and 1GHz clock frequency. LVDS Receiver using comparator as its second stage is designed and simulated in Cadence Virtuoso Analog Design Environment using GPDK 180nm .By this design, the power dissipation, delay and noise can be reduced. The technique described in Section III-C-1 can be used to reduce the number of clock cycles per iteration in the key scheduling process to 3 compared to 4 which did not use this "read-then-write" access mode, although the FPGA chip used in their implementation also supports it. Meanwhile, employed the dual-port writing feature to write S[i] and S[j] simultaneously in a single clock cycle to achieve throughput of three cycles per key scheduling iteration. However, under this implementation scheme, a possible memory contention problem may exist in the last clock cycle of each iteration the key scheduling and key generation. Thus, extra resources such as comparators are required to solve the problem.

However, Distributed RAM requires resource of slices to implement but Block RAM does not, because it is a standard primitive in most Xilinx FPGA chips. In an X2VP20-5 chip, one Block RAM consists of 18-kb memory. As one S-Box requires only 256X8=2kb memory, one Block RAM can be used to implement two S-Boxes for

All Rights Reserved © 2015 ICRAMATE16

testing two keys simultaneously. FPGA chips used consist of Block RAM with 4 kb or above. However, because of the limitation in their implementation architecture, one Block RAM can only be used for implementing one S-Box, resulting in smaller density of key searching units per chip compared to our architecture. Meanwhile, Distributed RAM requires slices [12] (one Distributed RAM of size of 256x8 bits requires 16 slices), thus more slice resources are required if one Distributed RAM is used to implement two S-Boxes. Hence, it is not recommended to use one Distributed RAM to implement two S-Boxes at a time.

Choice of using Distributed RAM or Block RAM to implement S-Boxes depends on the resource of slices and Block RAM in the FPGA device. We should first exploit all the Block RAM resources. If there are slice resources left, we can then use Distributed RAM to implement other S-boxes.

C. *Number of Clock Cycles in Key Scheduling Process*

By using the technique described in Section III-C1, the total number of clock cycles in the key scheduling process can be reduced from 1285 to 1028. In addition, we proved that by applying the technique described in Section III-C2, we can embed the S-Box initialization process into the S-Box scrambling process and so the total number of clock cycles in the key scheduling process can be further reduced to 772, which is the smallest compared to other proposed implementations. Christo Ananth et al. [7] proposed a secure hash message authentication code. A secure hash message authentication code to avoid certificate revocation list checking is proposed for vehicular ad hoc networks (VANETs). The group signature scheme is widely used in VANETs for secure communication, the existing systems based on group signature scheme provides verification delay in certificate revocation list checking. In order to overcome this delay this paper uses a Hash message authentication code (HMAC). It is used to avoid time consuming CRL checking and it also ensures the integrity of messages. The Hash message authentication code and digital signature algorithm are used to make it more secure . In this scheme the group private keys are distributed by the roadside units (RSUs) and it also manages the vehicles in a localized manner. Finally, cooperative message authentication is used among entities, in which each vehicle only needs to verify a small number of messages, thus greatly alleviating the authentication burden.

D. *System Performance Considerations*

The time required to crack an RC4 encryption with brute-force depends on the total number of parallel key-searching units in the FPGA chip, the maximum system clock frequency and the total number of clock cycles for testing one key.

1) Total Number of Parallel Key Search: Resource usage per key searching unit determines the maximum number possible in a FPGA chip: the higher the resource usage, the less the number of units. In Scheme 1, as it uses Distributed RAM which consumes slice resource to implement S-Boxes, resource usage per RC4 key searching unit is higher than that in Scheme 2 which uses Block RAM to implement S-Boxes. In addition, in Scheme 1, only one key searching unit can test a key at a time, whereas in Schemes 2 or 3, one key searching unit can test two keys simultaneously. Thus, the total number of key searching units using Schemes 2 and 3 is much larger than that using Scheme 1.

All Rights Reserved © 2015 ICRAMATE16

2) Maximum Clock Frequency: Complexity of each key searching unit determines the maximum clock frequency: the more complex the unit, the slower the maximum clock frequency can be used. Among the three schemes, Scheme 1 is simplest and Scheme 3 is the most complicated. Thus, the maximum clock frequency is the highest in Scheme 1 and the lowest in Scheme 3.

VI. CONCLUSION

In this paper, a highly parallelized RC4 key searching system based on a FPGA device is presented. Three different implementation schemes were proposed and tested. Their implementation results indicate that Scheme 3 can be used for an effective hardware implementation of a parallel key searching system in a Xilinx XC2VP20-5 chip. Such system can search more than 10^7 keys per second, which is the fastest among all single-FPGA-chip RC4 key searching systems proposed previously in the literature. Besides, the slice resource per key searching unit is 51 slices, which is also the smallest among other previously proposed implementations. Using this implementation, a complete 40-bit RC4 key space can be tested in 28.5 h. The design is modularized and can be easily adapted to other hardware platforms containing multiple FPGA chips. When bigger or faster FPGA devices are used, the time for cracking an RC4 encryption using brute-force can be further reduced. For example, if an XC2VP50-5 chip is used, we simulated that an RC4 key searching system consisting of 458 key searching units can be implemented in the chip. Such system can complete a 40-bit RC4 key space search in 11 h. Similar techniques can be applied to hardware implementations of other ciphers or cryptographic algorithms with a view to carrying out brute-force attack on the ciphers or constructing high-speed processing engines for the algorithms.

REFERENCES

[1]Christo Ananth, H. Anusuya Baby, “S-Box using AES Technique”, International Journal of Engineering Research & Technology (IJERT), Vol. 3 Issue 3, March – 2014, pp 285-290
[2] Christo Ananth, H.Anusuya Baby, “High Efficient Complex Parallelism for Cryptography”, IOSR Journal of Computer Engineering (IOSR-JCE), Volume 16, Issue 2, Ver. III (Mar-Apr. 2014), PP 01-07
[3] Christo Ananth, H.Anusuya Baby, “Encryption and Decryption in Complex Parallelism”, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), Volume 3, Issue 3, March 2014,pp 790-795
[4] Christo Ananth, M.Muthamil Jothi, A.Nancy, V.Manjula, R.Muthu Veni, S.Kavya, “Efficient message forwarding in MANETs”, International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE), Vol. 1, Issue 1, August 2015, pp:6-9
[5] Christo Ananth, Bincy P Chacko, “Analysis and Design of Low Voltage Low Noise LVDS Receiver”, IOSR Journal of Computer Engineering (IOSR-JCE), Volume 9, Issue 2, Ver. V (Mar - Apr. 2014), PP 10-18
[6] “Virtex-II Pro and Virtex-II Pro X FPGA User Guide” ver. 4.0, March 23, 2005
[7] Christo Ananth, M.Danya Priyadharshini, “A Secure Hash Message Authentication Code to avoid Certificate Revocation list Checking in Vehicular Adhoc networks”, International Journal of Applied Engineering Research (IJAER), Volume 10, Special Issue 2, 2015,(1250-1254)
[8] Christo Ananth, S.Esakki Rajavel, S.Allwin Devaraj, M.Suresh Chinnathampy. "RF and Microwave Engineering (Microwave Engineering)." (2014): 300,ACES Publishers.
[9] Christo Ananth, S.Esakki Rajavel, S.Allwin Devaraj, P.Kannan. "Electronic Devices." (2014): 300, ACES Publishers.

All Rights Reserved © 2015 ICRAMATE16