A. M. Balamurugan*, A. Sivasubramanian and B. Parvathavarthini

Secured Hash Based Burst Header Authentication Design for Optical Burst Switched Networks

DOI 10.1515/joc-2015-0097
Received November 14, 2015; accepted June 8, 2016

Abstract: The optical burst switching (OBS) is a promising technology that could meet the fast growing network demand. They are featured with the ability to meet the bandwidth requirement of applications that demand intensive bandwidth. OBS proves to be a satisfactory technology to tackle the huge bandwidth constraints, but suffers from security vulnerabilities. The objective of this proposed work is to design a faster and efficient burst header authentication algorithm for core nodes. There are two important key features in this work, viz., header encryption and authentication. Since the burst header is an important in optical burst switched network, it has to be encrypted; otherwise it is be prone to attack. The proposed MD5&RC4-4S based burst header authentication algorithm runs 20.75 ns faster than the conventional algorithms. The modification suggested in the proposed RC4-4S algorithm gives a better security and solves the correlation problems between the publicly known outputs during key generation phase. The modified MD5 recommended in this work provides 7.81% better avalanche effect than the conventional algorithm. The device utilization result also shows the suitability of the proposed algorithm for header authentication in real time applications.

Keywords: optical burst switching, core router, symmetric key encryption, RC4-4S, MD5, avalanche effect

1 Introduction

The development of optical burst switching relies on the successful development of several key technologies, including all-optical switches and optical wavelength converters. An optical burst switched network consists of switching nodes that are interconnected via fiber links [1]. Each fiber link is capable of supporting multiple wavelength channels using WDM [2]. Nodes in optical burst switched networks are classified into edge router and core router. Edge routers are responsible for assembling packets into bursts, and scheduling the bursts for transmission on outgoing wavelength channels. Edge router can also be called as an ingress node or an egress node. Ingress node is responsible for combining packets into a burst. The assembled bursts are transmitted optically over OBS core routers without any storage at intermediate nodes within the core. Egress node is responsible for disassembling the burst into packets. The edge router performs the functions of pre-sorting packets, buffering packets, assembling packets into burst, and disassembling bursts into its constituent packets [3].

The core router is referred to as an intermediate node. Typically, an OBS core node (switch) consists of two layers which are shown in Figure 1. The upper layer is responsible for processing control packets and configuring the switching fabric. Control packets are processed in this layer, switching resources are reserved, and switching resources are freed after the burst cut-through the switch [4]. The switch matrix control unit and the port forwarding table, which is a lookup table and link scheduling module are maintained. The lower layer is responsible for all-optical burst transport functionality. The lower layer consists of optical ports, wavelengths, and optical-to-optical connections [5].

The assembly of burst can be done by either setting a predefined maximum burst size or by setting a time out value. Once a burst is formed, the ingress node generates a burst header. This burst header contains all the details of the burst. The average length of the burst header is 40 bytes. It contains information such as source address, destination address, and number of packets in the burst, payload, and length of the burst and offset time etc. There is a separate channel for the burst and the burst header to travel [4, 5]. The channel used for transmission of the burst header is specific and different from the channels for transmission of transmit the bursts.

A distinctive characteristic of OBS is that the burst header undergoes O/E/O at each core router. During this

*Corresponding author: A. M. Balamurugan, St. Joseph's College of Engineering, Chennai, Tamil Nadu, India,
E-mail: bala_am2000@yahoo.com
A. Sivasubramanian, School of Electronics, VIT University, Chennai, Tamil Nadu, India, E-mail: shiva_31@yahoo.com
B. Parvathavarthini, St. Joseph's College of Engineering, Chennai, Tamil Nadu, India, E-mail: parvathavarthini@gmail.com

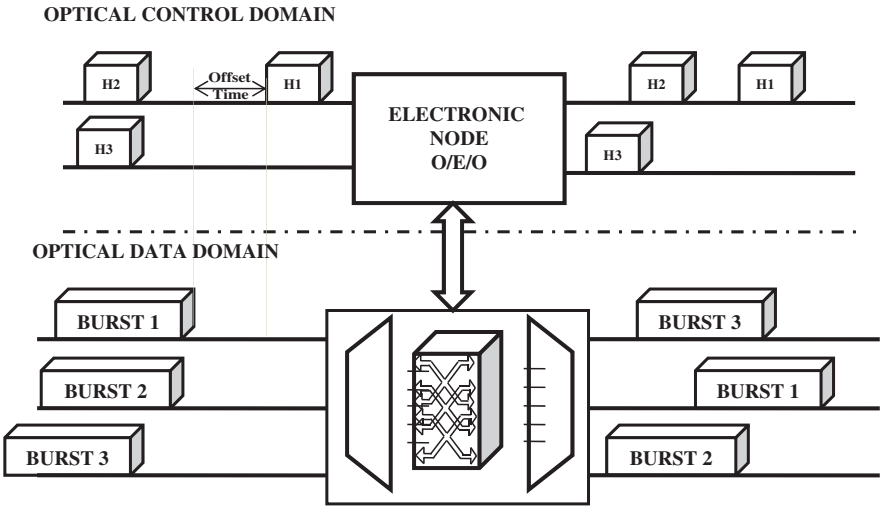

Figure 1: Core router architecture.

conversion process the burst header has greater possibility of being compromised. Due to importance in an optical burst switched network, the burst header has to be authenticated and encrypted otherwise it will be more prone to attack. Therefore per hop header authentication is an of essential security service for OBS network. This work proposes a modified MD5& RC4-4S based header authentication algorithm.

2 Attacks and vulnerabilities in core router

Certain type of attacks can occur, when a burst header is not encrypted and authenticated. These are detailed below.

Burst header flooding attack occurs when the optical node is encountered by intruders; the node creates many copies of the same burst header and therefore congests the next node. This leads to the flooding of the intermediate node with duplicate copies [6, 7]. So the next intermediate node makes reservations for the generated fake burst headers. This leads to a situation of overflow of buffers at the intermediate core node. Thus the reservation of uncompromised nodes is not possible until it receives a valid burst header. The other type of attack faced by the header is Fake burst header attack [8, 9]. In this threat, the attacker loads the intermediate node with a harmful fake burst header that redirects the incoming burst to a fake destination. After an offset time the burst approaches the compromised node from where it is sent to the fake destination. The malicious burst header is inserted in malicious burst header injection into the network by another party at the right time, to misguide the burst to an unauthorized node [10, 11].

3 Proposed burst header authentication design

Burst header has to be authenticated and encrypted for obtaining a secured OBS network. Such authentication is done by using a modified MD5 algorithm. When compared to the other algorithm, modified MD5 algorithm is salient due to its wider availability and relatively shorter length (128 bits). The burst header generated for each burst header is 320 bits and sent through the modified MD5 algorithm. A digest of 128 bit generated from the modified algorithm along with 320 bit header is concatenated to form 448 bits. The burst header is encrypted along with the message digest using the proposed RC4-4S algorithm and sent along the WDM channel.

The encrypted burst header has to be decrypted at the core router. This decryption of the burst header along with message digest is done using proposed RC4-4S algorithm. The RC4-4S decryption algorithm separates the burst header and the message digest. The decrypted burst header is processed again by the modified MD5 algorithm and a 128 bit message digest is generated. The decrypted message digests and the former generated message digest is compared to conclude whether a burst header has been modified. When the decrypted message digest and the former generated digest are the same, it can be concluded that the burst header is not modified.

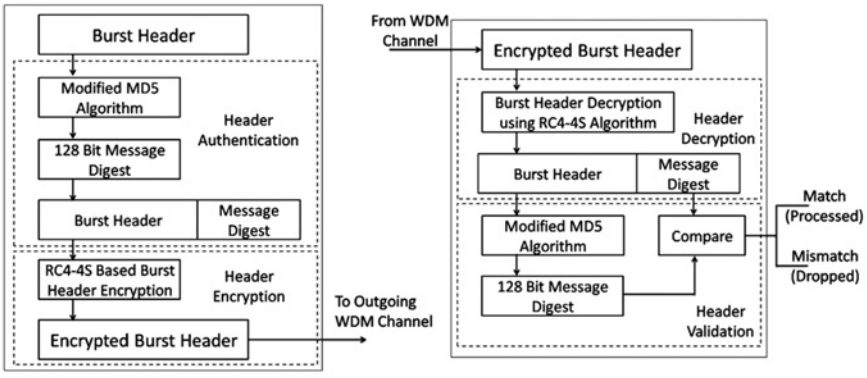

Figure 2: Proposed burst header authentication design.

The core router drops the burst header when any mismatch is found from the comparison. The proposed approach which is shown in the Figure 2 enhances the security service of the core router.

4 Symmetric key based burst header encryption algorithm design

Based on the encryption key, the ciphers are classified into two types, viz., symmetric key and asymmetric key approach. The symmetric key approach uses the same key for encryption and decryption. This proposed work uses symmetric key based burst header encryption. The asymmetric key approach will reduce the speed of the encryption/decryption. So the high speed OBS will rule this option out [7].

Apart from the key, and based on the input data, the ciphers can classified into two types called block ciphers and stream ciphers. The block ciphers encrypts block of data with a fixed size. Modification, insertion or deletion of the blocks is easy in the block cipher approach as the identical block of plain text which always gives identical block of cipher text. The stream ciphers encrypt continuous streams of data.

Each plaintext digit in the stream cipher is encrypted one at a time to produce the cipher output. The advantages of stream cipher are that they have lower hardware complexity and execute at higher speed. The stream ciphers are best where the amount of data is not known or when they are continuous. Stream ciphers are less vulnerable and mathematically easily analyzed. The very popular and widely used stream cipher algorithm is RC4.

4.1 Conventional RC4 realization

The conventional RC4 algorithm works in two phases: Key scheduling phase (KSA) and Pseudo random generation phase (PRGA). A 256 byte state table is initialized, using a variable length key from 1 to 256 bytes, in the RC4 realization. The pseudo-random bytes are generated with the help of the state table. Further the cipher text is obtained by the XOR operation with the plain text. The swapping operation is performed on every element in the state table, at least once. The first and the most difficult phase of the above mentioned algorithm is the key scheduling setup [12, 13].

The encryption variable uses two arrays, state [S] and key [K], and N-number of mixing operations for generating the encryption key. The encrypted message is created when the plain text is XORed with the encrypting variable, which is produced from the pseudo random generation phase. By XORing the encrypted message with the same encrypting variable, the receiver can decrypt the plaintext [14].

4.2 RC4 -2S algorithm

Hammood et al. [15] propose a RC4-2S algorithm to reduce the problem between the public known outputs correlation. Two important shortcomings of the conventional RC4 algorithms are 1) KSA weakness and 2) The S box relation weakness [15, 16]. This algorithm consists of two stable state vectors named S1 & S2. In the first phase of KSA, S1 is computed and swapped between 0 and (N/2)-1.In a similar manner, the swapping of S2 is only between N/2 and N-1. It gets two different secret random inputs S1 & S2 due to this second phase. From this new S1 & S2 the sequence of key will be generated

which is XORed with the plain text for producing cipher text. The PRGA of RC4-4S algorithm requires two swaps and five modulo functions to generate two bytes of key per iteration, so the RC4-2S is faster than conventional RC4 while RC4 requires one swap and three modulo functions to generate a key [15].

4.3 Proposed RC4-4S algorithm

In the proposed RC4-4S algorithm, there are four states [S1, S2, S3 and S4]. The first phase of the KSA, shows that S1 is filled from (N/4)-1, S2 is filled from N/4 to (N/2)-1, S3 is filled from N/2 to (3N/4)-1 and the state S4 gets the remaining N/4 numbers from 3N/4 to N-1.These four states S1, S2, S3 and S4 get seeded with the help of the input secret key, K. The permutation and swapping of the elements of S1, S2, S3 and S4 are done by the use of this key, K. Hence, the four secret random inputs for the second phase are S1, S2, S3 and S4.

In the second phase, the four states S1, S2, S3 and S4 produce four keys in each loop cycle, instead of one, as with the standard RC4 and two, as with the RC4-2S algorithm. In this RC4-4S algorithm, there are more elements to be swapped between the four states, by making use of the five pointers: i, $j_1 = j_1 + s_1[i]$, $j_2 = j_2 + s_2[i]$, $j_3 = j_3 + s_3[i]$ and $j_4 = j_4 + s_4[i]$, in the S-box. The sequence of output stream which is XORed with plain text to generate cipher text is produced by the four states S1, S2, S3 and S4 in PRGA.

5 Hash algorithm

Hash function is the method of compressing the strings. It takes an input of arbitrary length called message and produces an output of fixed length called digest. The main features of using hash algorithm are due to its short and fixed length. The hash algorithms are used for data authentication. This proposed work uses MD5 algorithm for header validation.

5.1 MD5 algorithm

MD5 processes a variable length message into a fixed length output of 128 bits. The input message is broken up into chunks of 512 bit blocks; the message is padded to ensure its length is divisible by 512. In the processing of padding a single bit "1" is appended at the end of the message. This is followed by as many as zeros such that the length of the message is 64 bits [17].

The main MD5 algorithm operates on a 128 bit state which is divided into four 32 bit words. The four 32 bits are stored in buffer and are designated as A, B, C & D. Initially they are fixed constants which are shown in eq. (1). Since the message is 512 bits, a total of 16 rounds are required to update these four 32 bit values. The buffers can be updated in each round with the help of nonlinear functions. This is shown in eq. (2).
The buffer values are initialized as follows:

$$\begin{aligned} A &= 0x67452301 \\ B &= 0xefcdab89 \\ C &= 0x98badcfe \\ D &= 0x10325476 \end{aligned} \tag{1}$$

The non linear functions are defined as:

$$\begin{aligned} F(B, C, D) &= (B \wedge C) \vee (\neg B \wedge D) \\ G(B, C, D) &= (B \wedge D) \vee (C \wedge \neg D) \\ H(B, C, D) &= B \oplus C \oplus D \\ I(B, C, D) &= C \oplus (B \vee \neg D) \end{aligned} \tag{2}$$

$\oplus$, $\wedge$, $\vee$, $\neg$ denote the XOR, AND, OR and NOT operations respectively.

5.2 Modified MD5 algorithm

For increasing the security level of the MD5 algorithm few modifications has been suggested in the conventional round function which is shown in Figure 3.

The suggested modifications are:

- The 32 bit value of buffer A and the output of the non linear function F is XORed instead of modulo 2^{32} additions.
- The updated 32 bit value of M_i is XORed with K_i instead of modulo 2^{32} additions.

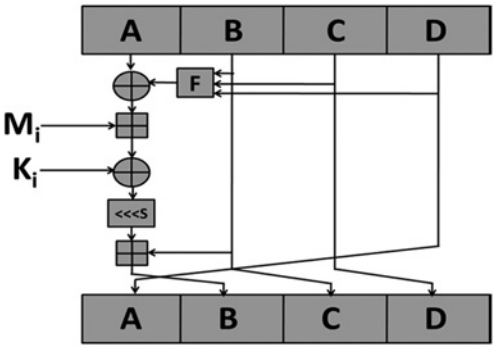

Figure 3: Modified MD5 algorithm.

– The modulo 2^{32} additions are replaced by XOR differentials because XOR differentials are difficult to analyze [18].

6 Performance comparison of the proposed burst header authentication algorithm

The proposed RC4-4S algorithm has been successfully synthesized using Xilinx ISE 14.6. The HDL used for this simulation is verilog. The performance analysis of time complexity and device utilization of the RC4-4S algorithm are presented in Tables 1 and 2.

The RC4-4S algorithm has been synthesized on commercial off the shelf FPGA for checking the physical feasibility in silicon. The FPGA model used for synthesizing the RC4-4S algorithm is VIRTEX 7. The propagation delay of the RC4-4S algorithm is compared with RC4 and RC4-2S and is presented in Table 1. This table shows that the RC4-4S algorithm is 6.2 ns faster than the RC4-2S and 17.4 ns faster than the conventional RC4 algorithm, as it requires four swaps and five modulo functions to generate four bytes of key.

Table 2 presents the hardware implementation for the RC4-4S algorithm on VIRTEX7. The slice utilization percentage for the RC4-4S algorithm is only 1% of the available resource while the LUT utilization varies between 9 and 10%. From the Table 2, it proves that the RC4-4S algorithm can be implementable on a single COTS FPGA.

The avalanche effect comparison of the modified MD5 algorithm is shown in Table 3. The avalanche effect is one of the desirable features to investigate the performance of any integrity algorithm. The average number of flipped bits is evaluated for different input combinations and it is compared with conventional MD5 algorithm. Table 3 helps concluding the avalanche effect of the modified MD5 algorithm as 7.81% better than the conventional MD5 algorithm.

In a similar manner, the computational time complexity comparison is also made for the modified MD5 with RC4-4S header authentication algorithm. This is shown in Table 4. It is clear that the proposed MD5 & RC4-4S algorithm is 20.75 ns faster than the conventional MD5 & RC4 conventional algorithm.

Table 3: Avalanche effect comparison.

Authentication algorithms	Avalanche effect comparison		
	Digest size (bits)	Average number of flipped bits	Avalanche effect (%)
Conventional MD5 algorithm	128	59	46.09
Modified MD5 algorithm	128	69	53.90

Table 1: Time complexity comparison of RC4-4S algorithm.

Header encryption algorithm	Time complexity (Model: VIRTEX7: Device:xc7vx1140t-2G-flg1930)				
	Total time in ns	Logic	Logic (%)	Route	Route (%)
Conventional RC4 algorithm	80.284 ns	21.572 ns	26.9	58.712 ns	73.1
RC4-2S algorithm	69.264 ns	19.855 ns	28.7	49.410 ns	71.3
Proposed RC4-4S algorithm	62.814 ns	16.281 ns	25.9	46.533 ns	74.1

Table 2: Device utilization comparison of RC4-4S algorithm.

Header encryption algorithm	Device utilization (Model: VIRTEX7: Device:xc7vx1140t-2G-flg1930)					
	Slice			LUTs		
	Available	Used	Percentage	Available	Used	Percentage
Conventional RC4 agorithm	712,000	8,015	1.1	8,267	883	10
RC4-2S algorithm	712,000	8,581	1.2	9,106	909	9.9
Proposed RC4-4S algorithm	712,000	11,511	1.6	11,881	1,071	9

Table 4: Time complexity comparison of proposed burst header authentication algorithm.

Burst header authentication algorithm	Time complexity (Model: VIRTEX7: Device:xc7vx1140t-2G-flg1930)				
	Total time in ns	Logic	Logic (%)	Route	Route (%)
MD5 & RC4 algorithm	84.026 ns	22.100 ns	26.3	61.927 ns	73.7
MD5 & RC4-2S algorithm	70.200 ns	20.485 ns	29.2	49.715 ns	70.8
Proposed MD5 & RC4-4S algorithm	63.270 ns	16.350 ns	25.8	46.920 ns	74.2

7 Conclusion

For providing security services to the core router, the header must be encrypted before it propagate to the next core router. This is because the burst header contains the details about the burst. The objective of this proposed work is to develop an efficient and fast per hop burst header authentication algorithm. This proposed work also recommends hash function associated with burst header for authentication. The modified MD5 algorithm used in this work is derived from the conventional MD5 algorithm due to its wide availability and relatively short length. The RC4-4S algorithm is 6.2 ns faster than the RC4-2S algorithm and 17.4 ns faster than the conventional algorithm. The increased speed is due to the proposed algorithm requiring four swaps and five modulo functions to generate four bytes of keys. The slice utilization percentage of the RC4-4S algorithm is only 1% and the LUT utilization varies between 9 to 10%. The modified MD5 algorithm has been designed with XOR differentials in the round function for better security and lesser computation time, also it produces a 7.81% better avalanche effect. The comparison results states that the proposed MD5&RC4-4S based approach is 20.75 ns faster than the conventional algorithms. The implementation results help the conclusion that the proposed burst header authentication algorithm is suitable for real time applications in optical burst switched networks.

References

1. Qiao C, Yoo M. Optical burst switching (OBS) – a new paradigm for an optical internet. J High Speed Networks 1999;8:69–84.
2. Chlamtac. I, Ganz. A, Karmi. G. Lightpath communications: an approach to high bandwidth optical WAN's. IEEE Trans Commun 1992;40:1171–1182.
3. Balamurugan AM, Sivasubramanian A. Optical burst switching issues and its features. Int J Emerg Trends Technol Comput Sci 2013;2:306–15.
4. Gauger C, Dolzer K, Spath J, Bodamer S. Service differentiation in optical burst switching networks. Photonic Networks 2001, March 2001, 124–132.
5. Haselton F. A PCM frame switching concept leading to burst switching network architecture. IEEE Commun Mag 1983;21:13–19.
6. Sreenath N, Muthuraj K, Vinoth G. Threats and vulnerabilities on TCP/OBS networks. International Conference on Computer Communication and Informatics (ICCCI) 2012, January 2012, 1–5.
7. Chen YH, Verma PK, Kak S. Embedded security framework for integrated classical and quantum cryptography services in optical burst switching networks. Secur Commun Networks 2009;2:546–54.
8. Balamurugan AM, Sivasubramanian A. Modeling the performance of DDoS attack in optical burst switched networks. Aust J Basic Appl Sci 2014;8:479–82. December 2014.
9. Balamurugan AM, Sivasubramanian A. A novel QKD based secure edge router architecture design for burst confidentiality in optical burst switched networks. J Opt Commun 2014;35:109–16. ISSN (Online) 2191–6322, ISSN (Print) 0173–4911, DOI: 10.1515/joc-2014-0003.
10. Fok MP, Wang ZX, Deng YH, Prucnal PR. Optical layer security in fiber-optic networks. IEEE Trans Inf Forensics Secur 2011;6:725–36.
11. Balamurugan AM, Sivasubramanian A. Quantum key based burst confidentiality in optical burst switched networks. Sci World J 2014;2014:7. Article ID 786493, DOI:10.1155/2014/786493.
12. Mousa A, Hamad A. Evaluation of the RC4 algorithm for data encryption. Int J Comp Sci Appl 2006;3:44–56.
13. Singhal, N, Raina JP. Comparative analysis of AES and RC4 algorithms for better utilization, Int J Comp Trends Technol 2011;6:177–81.
14. Seth SM, Mishra R. Comparative analysis of encryption algorithms for data communication. Int J Comp Sci Technol 2011;2:292–4.
15. Hammood MM, Yoshigoe K, Sagheer AM. RC4-2S: RC4 stream cipher with two stable tables, information technology convergence, lecture notes in electrical engineering 2013;253. DOI: 10.1007/978-94-007-6996-0_2, Springer science + Business media Dordrecht 2013.
16. Fluhrer S, Mantin I, Shamir A. Weakness in the key scheduling algorithm of RC4. In: Proceedings of annual workshop on selected areas in cryptography, vol. 2259. Toronto: Springer, 2009:1–24
17. Stallings W. Cryptography and network security: principles and practice, 5th ed. USA: Prentice Hall, 2011.
18. Balamurugan AM, Sivasubramanian A, Parvathavarthini B. QKD based secured burst integrity design for optical burst switched networks. J Opt Commun. ISSN (Online) 2191–6322, ISSN (Print) 0173–4911, March 2015, DOI: 10.1515/joc-2014-0093.