IOP Conference Series: Materials Science and Engineering

PAPER • **OPEN ACCESS**

Implementation of salsa20 stream cipher algorithm as an alternative cipher suite SSL-VPN for VOIP security

Implementation of salsa20 stream cipher algorithm as an alternative cipher suite SSL-VPN for VoIP security

Aghnia Luthfy Nugrahtama and Yogha Restu Pramadi*

Laboratory of Cryptographic Software Engineering Sekolah Tinggi Sandi Negara/National Crypto Institute, Indonesia

*yogha.restu@stsn-nci.ac.id

Abstract. Voice over Internet Protocol (VoIP) is one of the emerging communication technologies. The use of VoIP has the threat of tapping. Therefore, it is necessary to implement security services for VoIP, one of them using SSL- VPN. SSL-VPN uses cipher suites that contain a collection of cryptographic algorithms to provide various cryptographic services, starting from authentication, key exchange, encryption and message digest. In this study, modification of cipher suites was carried out in SSL-VPN, by implementing the Salsa20 stream cipher algorithm as an alternative to TLS cipher suites. Furthermore, the results of the alternative implementation of TLS cipher suites are used and tested to secure VoIP communication. Tests carried out are vectors test, data transfer speed performance test, and measurement of QoS dan MOS of VoIP communication secured with the alternative Salsa20 TLS cipher suite.

1. Introduction

Voice over Internet Protocol (VoIP) is a protocol for transmitting real-time voice data using the internet protocol network [1], The principle of VoIP work is simply to change the analog sound (in the form of a human voice) obtained from the speaker sensor into digital data packages, then the data packets are sent via the internet network to the VoIP server, then VoIP server forwards the packets to the recipient, and vice-versa so that voice communication occurs between the two parties [2],

The use of VoIP has several advantages, but on the other hand, also has a vulnerability in the form of wiretapping by unauthorized parties [3]. Tapping can cause conversation confidentiality services not guaranteed [4]. Therefore, it is necessary to implement security for VoIP, using VPN [5] [6]. In general, the use of SSL-VPN can meet the security aspects needed by VoIP.

SSL-VPN provides several TLS cipher suites containing cryptographic algorithms. AES algorithm used as the default cipher suites in Open VPN for VoIP security has a considerable influence on VoIP performance according to QoS (Quality of Service) [6]. QoS parameters that get influence are a delay (latency), jitter, packet loss, and throughput.

SSL-VPN uses a symmetric algorithm as an encryption algorithm that already exists in the default TLS cipher suites. There are still other symmetric algorithms that in theory have better computing speeds. One of them is the Salsa20 stream cipher algorithm which has better computational speed compared to the default TLS cipher suites, AES [7]. Therefore, this research will try to modify an SSL-VPN implementation by implementing the Salsa20 stream cipher algorithm and measure its performance as an alternative choice of TLS cipher suites. Furthermore, this research is related with previous research, such as [4] [6] [8] [9] [10].

2. Related Works

In 2010, a study was carried out by Radmand & Talevski [6] that conducted a research related to YPN implementation to overcome vulnerabilities in VoIP. The research measured QoS with several different encryption algorithms used in the VPN. VPN is a security mechanism using a tunnel to secure communication. In this study, OpenVPN was used as the VPN software.

Furthermore, in 2012, another VoIP-related research was carried out [8]. The purpose of the research conducted by Dherik Barison et al was to evaluate the quality of encrypted VoIP communication with several different encryption algorithms using OpenVPN. Evaluation is done using MOS parameters.

Then in 2015, a research on VPN-SSL Implementation on OpenBTS and VoIP [9], aimed to provide a safe exchange of information on blank spot areas and add security services to data transmission sent between BTS and VoIP. This research is useful to secure data transmission in areas that have minimal telecommunications network facilities. In the same year, [10] also aims to build a prototype personal VPN gateways that work on the layer 4 OSI layer model (SSL-VPN), the OpenVPN is modified by implementing ATHS3 stream cipher algorithm as an alternative choice of TLS cipher suites.

Table 1. List of Additional Source Code and Modification

AdditionalSourceCode	**Modified Source Code**
openssl/crypto/salsa20/salsa20. c	openssl/config
openssl/crypto/salsa20/salsa20 locl.h	openssl/configure
openssl/crypto/salsa20/build.info	openssl/crypto/evp/build. info
openssl/crypto/include/internal/salsa20.h	openssl/crypto/evp/c_allc.c
	openssl/crypto/objects/objects .txt
	openssl/include/openssl/evp.h
	openssl/include/openssl/tls1.h
	openssl/include/openssl/ssl.h
	openssl/include/openssl/objmac.h
	openssl/ssl/s3_lib.c
	openssl/ssl/ssl_ciph.c
	openssl/ssl/sslinit. c
	openssl/ssl/ssllocl.h
	openssl/ssl/tl_trce.c
	openssl/apps/openssl.c
	openssl/apps/progs.pl
	openssl/apps/speed.c
	openssl/util/libcrypto.num
	openssl/doc/apps/dsa.pod
	openssl/doc/apps/gendsa.pod
	openssl/doc/apps/genrsa.pod
	openssl/doc/apps/rsa.pod
	openssl/doc/apps/pkcsl2.pod
	openssl/doc/apps/ciphers.pod

3. Implementation, Testing, and Analysis

At this stage, the system is installed, configured, and implemented according to the design model that has been carried out. Implementation is carried out using C programming language because the source code of cipher suites SSL-VPN uses C language. There are several tests conducted in this study.

3.1. Implementation of Salsa20 Alternative Ciphersuite

At this stage, the system is installed, configured, and implemented according to the design model that has been carried out. Implementation is carried out using C programming language because the source code of cipher suites SSL-VPN uses C language. There are several tests conducted in this study.

To implement Salsa20 into the OpenSSL library, we need to develop a new code for salsa20. And for the code to be able to be chosen in the TLS handshake we need to modify the OpenSSL source code, so it is available as a ciphersuite. in Table 1 we list the code that is developed and modified to support Salsa20.

3.2. VoIP and SSL-VPN implementation on server-client side

After the implementation phase of alternative cipher suites, the next step is to configure both the server and client in order to function in accordance with the communication scenario that has been designed, as depicted in Figure 1.

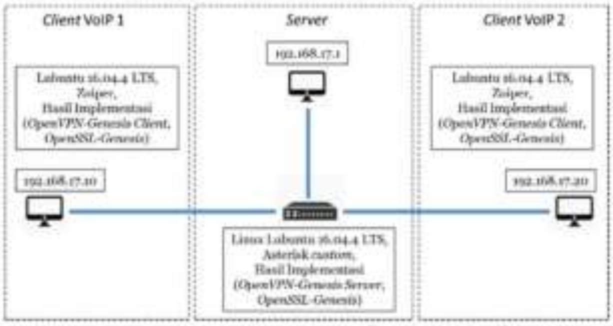
Client VoIP 1
Server
Client VoIP 2
192.168.17.1
Lubuntu 16.04.4 LTS, Zoiper, Hasil Implementasi (OpenVPN-Genesis Client, OpenSSL-Genesis)
Lubuntu 16.04.4 LTS, Zoiper, Hasil Implementasi (OpenVPN-Genesis Client, OpenSSL-Genesis)
192.168.17.10
192.168.17.20
Linux Lubuntu 16.04.4 LTS, Asterisk custom, Hasil Implementasi (OpenVPN-Genesis Server, OpenSSL-Genesis)

Figure 1. VoIP Communication Design

3.3. Vector test of Salsa20 stream cipher algorithm

This stage is a test that must be done after the implementation is completed. Vector test is the data provided in each cryptographic algorithm document to verify the correctness of the implementation of the algorithm. This test is done by comparing the results of vector test between Salsa20 stream cipher algorithm encryption on OpenSSL-l.l.Oh with Salsa20 stream cipher algorithm that contained in the original document, the result is displayed on Table 2. The purpose of this test is to ensure that the implementation of the Salsa20 stream cipher algorithm on OpenSSL-1.1 .Oh has been carried out correctly and the computation process is in accordance with the original algorithm design so that the same output is obtained with the Salsa20 stream cipher algorithm vector test issued by the ECRYPT.

Table 2. The result of Vector Test 1

Key	80000000000000000000000000000000 00000000000000000000000000000000
Initialization Vector (IV)	0000000000000000
Plaintext (bit 0 - 63)	00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000 00000000000000000000000000000000
Ciphertext (bit 0 -63)	E3BE8FDD8BECA2E3EA8EF9475B29A6E7 003951 El 097A5C38D23B7A5FAD9F6844 B22C97559E2723C7CBBD3FE4FC8D9A07 44652A83E72A9C461876AF4D7EF1A117

3.4. *Functional Test of Server-Client Devices*

At this stage, the author will do a test related to the client laptop can communicate with the server laptop and can access the services provided. The minimum device needed to do this test is two laptops as clients, one laptop as a server, and one switch. The configuration of the IP address of each device can be seen in Table 3.

Table 3. IP Address Configuration

Device	Interface	Type	Configuration
Server laptop	enp0s3	Static	IP Address : 192.168.17.1 Subnet Mask: 255.255.255.0
Client laptop-1	enp0s3	Static	IP Address : 192.168.17.10 Subnet Mask:
Client laptop-2	enp0s3	Static	IP Address: 192.168.17.20 Subnet Mask:

3.5. *Performance test of Salsa20 TLS ciphersuites*

This test was carried out with the aim of benchmarking or comparing the data transfer rates of Salsa20 cipher suites on OpenVPN-Genesis with the default cipher suites in Open VPN-2.4.6. The selected cipher suites have the same authentication algorithm, key exchange, and message digest. This aims to equalize the cipher suites specifications of each algorithm. Data transfer will be done using netcat via a VPN tunnel that has been added to the alternative cipher suites Salsa20. One algorithm will be used to transfer data with size variants 10 MB, 20 MB, 50 MB, 100 MB, and 200 MB. The average data transfer speed in second units of each cipher suites is displayed in Table 4.

Table 4. Netcat Data Transfer Speed

Size of File	Ciphersuites Algorithms			
	SALSA20	CHACHA20 -	AES128-GCM	AES128-CBC
10 MB	0.79s	0.79s	0.81s	0.80s
20 MB	1.56s	1.58s	1.56s	1.59s
50 MB	4.13s	4.16s	4.15s	4.15s
100 MB	8.60s	8.60s	8.57s	8.59s
200 MB	17.42s	17.43s	17.45s	17.44s

3.6. *Quality of Service (QoS) testing*

QoS measurement is carried out in three communication scenarios: (i) normal VoIP communication without security; (ii) VoIP communication with OpenVPN security, default cipher suites; and (iii) VoIP communication with OpenVPN Salsa20 cipher suites security.

a. Delay

Delay testing measures the impact of additional processing to the time to transmit data packets caused by adding encryption to the packet processing. The results of delay measurement can be seen in Table 5. Table 5 shows the results of the average delay obtained from each communication scenario.

Table 5. Average delay results

VoIP Communication Delay Average (ms)	
Without Open VPN	19.98
Open VPN Default Ciphersuites	19.97
Open VPN Salsa20 Ciphersuites	19.98

b. *Jitter*

Jitter measurement results in the three scenarios that are tested can be seen in Table 6. Jitter is caused by delay variations that occur when VoIP communication is in progress. The measurement shows that VoIP communication with OpenVPN default cipher suites has the largest jitter average compared to the other two scenarios.

Table 6. Average jitter results

VoIP Communication	Jitter Average (ms)
Without OpenVPN	1.311
OpenVPN Default Ciphersuites	3.049
OpenVPN Salsa20 Ciphersuites	1.912

c. *Packet Loss*

Packet loss analysis is carried out to find out the size of the packet loss at the time of delivery. If the value of packet loss is large, then the quality of VoIP communication is not good. From the data obtained at the time of testing, the amount of packet loss VoIP communication with the three scenarios carried out is 0%, which means that there are no packages lost at all when communication is in progress. The result is expected as the testing condition is carried out in an internal network.

d. *Throughput*

The results of throughput measurements in the three scenarios tested can be seen in Table 7. The highest average throughput values are obtained when VoIP communication uses OpenVPN default cipher suites. The throughput value obtained in this test is inversely proportional to the delay obtained. It can be concluded that the throughput value is inversely proportional to the delay value. The value of throughput will correspond to the amount of bandwidth capacity in each codec used by VoIP. These results also show that Salsa20 alternative TLS cipher suites can be used as an alternative to the default AES default cipher suites because it has a significant amount of throughput.

Table 7. Average throughput results

VoIP Communication	Throughput Average (Kbps)
Without OpenVPN	245.9
OpenVPN Default Ciphersuites	306.1
OpenVPN Salsa20 Ciphersuites	298.5

3.7. Mean opinion score (MOS) testing

The data gathered from MOS testing is presented in Table 8, the highest test result data is obtained when running the first scenario (normal VoIP communication without OpenVPN security). While the lowest result is when running a VoIP communication scenario with the default OpenVPN security of cipher suites. In this test, the MOS values obtained from the three scenarios carried out show results that are not much different.

Table 8. MOS testing results

VoIP Communication	MOS Average
Without OpenVPN	4.75
OpenVPN Default Ciphersuites	4.55
OpenVPN Salsa20 Ciphersuites	4.65

4. Conclusion

Based on the research that has been done, the following conclusions can be obtained:

1. Salsa20 stream cipher algorithm was successfully implemented in OpenVPN as a TLS cipher suites. The compilation process between OpenSSL source code and OpenVPN source code is done with a special script or command so that OpenVPN can recognize the new algorithm implemented as a TLS cipher suite in OpenSSL.
2. The QoS testing shows that the Salsa20 cipher suites performance is almost identical to the AES default cipher suites, with the only noticeable difference is in the average jitter test in which the Salsa20 is better. Therefore, the Salsa20 as an alternative cipher suites could give similar QoS performance as the default cipher suites.

5. References

[1] Hallock, J. (2004). *A brief history of VoIP* (Document One - The Past). *Evolution and trends in digital media technologies* (Washington DC: COM 538)

[2] Wu C Y, Wu K P, Shih J and Lee H M 2011 VoIPS: VoIP secure encryption VoIP solution *Communications in Computer and Information Science CCIS* **223** 84- 93

[3] Patil, H. K., Wing, D., & Chen, T. M. (2013) *VoIP Security* J R Vacca, Ed., *Computer and Information Security Handbook* (2nd ed., p. 1200) (Atlanta, GA: Elsevier Inc.)

[4] Mohamed K, Mohamed O, Hamoudi M and Masmoudi M 2015 QoS evaluation in VoIP software with and without Blowfish encryption module *Proc. 2015 3rd IEEE Int. Conf. on Control, Engineering & Information Technology (CEIT)* (Tlemcen, Algeria, May 25-27) 254-259

[5] Bates R J B 2015 *Securing VoIP keeping your VoIP network safe* S Elliot, Ed. Waltham (Atlanta, GA: Elsevier Inc.)

[6] Radmand P and Talevski A 2010 Impact of encryption on QoS in VoIP *Proc. SocialCom 2010: 2nd IEEE Int. Conf. on Social Computing, PASSAT 2010: 2nd IEEE Int. Conf. on Privacy, Security, Risk and Trust* 721-726

[7] Bernstein D J 2007 The Salsa20 family of stream ciphers *CR. YP. TO.*

[8] Barison, D., Miani, R. S., Zarpelao, B. B., Davis Breda, G., & De Souza Mendes, L. (2012). Evaluation of quality in encrypted VoIP calls. *Proceedings of the 2012 4th Int. Conf. on Computational Aspects of Social Networks CASoN 2012* 175-180

[9] Rino 2015 *Implementasi VPN-SSL pada OpenBTS sebagai Pengamanan Data yang Ditransmisikan Antar BTS maupun VoIP* (Jakarta, Indonesia: Sekolah Tinggi Sandi Negara)

[10] Pramudianto A D 2015 *Implementasi ATHS3 sebagai alternatif ciphersuites OpenVPN dalam pembuatan prototipe perangkat personal virtual private network gateway (AR-6000) berbasis single board computer (SBC) Raspberry Pi Model B+.*(Jakarta, Indonesia: Sekolah Tinggi Sandi Negara)