Fast and Secure Communication Resume Protocol for Wireless Networks

Kihong Kim¹, Jinkeun Hong², and Jongin Lim¹

¹ Graduate School of Information Security, Korea University,
1, 5-Ka, Anam-dong, Sungbuk-ku, Seoul, 136-701, South Korea
hong0612@hanmir.com, jilim@korea.ac.kr

² Division of Information and Communication, Cheonan University,
115 Anse-dong, Cheonan-si, Chungnam, 330-740, South Korea
jkhong@cheonan.ac.kr

Abstract. There are important performance issues in secure wireless networks, such as power, bandwidth, and bit error rate (BER), that must be considered when designing a communication resume protocol. The efficiency of a secure communication resume for a fast resume of secure communication is a key point in secure connection development. In this paper, a fast secure communication resume protocol for a wireless network is presented and evaluated against the efficiency of conventional resume protocols. Our proposed resume protocol is found to achieve better performance, in terms of transmission traffic, consumed time, and BER, than conventional resume protocols.

1 Introduction

The wireless transport layer security (WTLS) provides privacy, authentication, and integrity in wireless application protocol (WAP) [1]. As the use of wireless networks becomes more widespread, the necessity of security for these networks is of increasing importance. However, in order to solve security issues in wireless networks, the efficiency of security services must be taken into account. From the point of view of wireless environmental characteristics, research on optimizing the security considerations of WTLS, such as low bandwidth, limited consumed power energy and memory processing capacity, and cryptography restrictions, has been presented [2][3][4][5]. Secure session exchange key protocol and security in wireless communications have been researched by Mohamad Badra and Ahmed Serhrouchni [6], and by Mohammad Ghulam Rahman and Hideki Imai [7]. Hea Suk Jo and Hee Yong Youn [8] examined a synchronization protocol for authentication in wireless LANs, while Min Shiang Hwang et al. [9] proposed an enhanced authentication key exchange protocol. However, in terms of efficiency, the performance considerations for secure wireless networks, such as power, bandwidth, and BER, are very important. Of particular importance for a secure connection point is the efficiency of the secure communication resume for the fast resume of secure communication. In this paper, a protocol for fast secure communication resume using IV count in wireless networks is presented and its performance is evaluated against that of conventional resume protocols.

Results shows that the proposed protocol achieves better performance in terms of transmission traffic, consumed time, and BER than conventional protocols.

The remainder of this paper is organized as follows. In the next section, detailed descriptions of the full handshaking and communication resume protocol are given. In section 3, the proposed secure communication resume protocol is illustrated. Some performance considerations are presented in section 4, and concluding remarks are provided in section 5.

2 Key Handshaking Protocol in WTLS

The WTLS protocol determines the session key handshaking mechanism for secure services and transactions in wireless networks, and consists of the following phases: the handshaking phase, the change cipher spec phase, and the record protocol phase (RP) [3][4][5]. In the handshaking phase, all the key techniques and security parameters, such as protocol version, cryptographic algorithms, and the method of authentication, are established between the client and the server. After the key handshaking phase is complete, the change cipher spec phase is initiated. The change cipher spec phase handles the changing of the cipher [6][7]. The RP phase is a layered protocol phase that accepts raw data to be transmitted from the upper layer protocols.

Secure communication necessitates the encryption of communication channels. To achieve this, a key handshaking protocol allows two or more users to share a key or an IV. A key handshaking protocol is illustrated in Fig. 1. The client sends a client hello message that includes information such as the version, acceptable cipher suites, and client random. When the server receives the client hello message, it responds with a hello message to the client and it also sends its certificate, key exchange, certificate request, and server hello done message. After receiving the server hello done message, the client responds by authenticating itself and sending its certificate. Then, the client generates the premaster secret and sends its encryption data $E_{KUS}[Premaster\ Secret]$ encrypted with the server's public key to the server. The premaster secret is used to generate a master secret that is shared between the client and the server. The client then generates the master secret using the premaster secret, client random, and server random. It also generates a sufficiently long key block using the master secret, client random, and server random [1]. The generated key block is hashed into a sequence of secure bytes, which are assigned to the message authentication code (MAC) keys, session keys, and IVs. This is represented as follows in Eq. (1).

$$\begin{aligned} MS &= PRF(PS, "MS", CR + SR) \\ KB &= PRF(MS, "KE", CR + SR) \end{aligned} \quad (1)$$

Here, MS is the master secret, KB is the key block, PS is the premaster secret, CR is the client random, SR is the server random, PRF is the pseudo random function, and KE is the key expansion. The client sends a change cipher spec message and finished message. The server also generates MAC secrets, session keys, and IVs. Then it sends the finished message to the client. Finally, secure communication is established using session keys and IVs.

Fig. 1. Full handshaking protocol in WTLS protocol

2.1 Communication Resume Protocol Using Premaster Secret

After completion of the full handshaking protocol shown in Fig. 1, a secure communication is established. However, data frame loss occurs because of bit slips, channel loss, reflection, and diffraction in the communication channel. If a data frame is lost, the output of the decryptor will be unintelligible for the receiver and a communication resume will be required. The aim of the communication resume is to ensure that the encryptor and decryptor have the same internal state at a certain time. An internal state different from all previous sessions has to be chosen to prevent the reuse of session keys or IVs [10][11][12]. To overcome the problems caused by these data frame losses, resume protocols for secure communication have been suggested. Such protocols can be achieved by one of two methods: 1) premaster secret regeneration and retransmission, or 2) random regeneration and retransmission.

Fig. 2 shows a protocol for a secure communication resume using premaster secret regeneration and retransmission. In this protocol, a new premaster secret is generated and sent in each communication resume, and thus it results in the generation of a new master secret and new key block. Therefore, new session keys and new IVs are generated for every communication resume. However, since a new premaster secret is generated and sent in each secure communication resume, this method has disadvantages such as a large computation load, time delay, and BER.

This protocol is executed as follows. First, secure communication is performed for time Δt , and then data frame loss occurs. After the server realizes the data

Fig. 2. Conventional protocol using premaster secret

frame loss, it requests a new premaster secret for communication resume. The client generates a new premaster secret and sends $E_{KUS}[\text{Premaster Secret}]$ to the server. The client then generates a new master secret using the new premaster secret and the original random cached in the initial hello message stage, and generates a new key block using the new master secret and original random. Thus, the result is the generation of new session keys and new IVs.

$$\text{New MS} = \text{PRF}(\text{New PS}, \text{"MS"}, \text{Original CR} + \text{Original SR}) \quad (2)$$

$$\text{New KB} = \text{PRF}(\text{New MS}, \text{"KE"}, \text{Original CR} + \text{Original SR})$$

The client then sends the finished message to the server. The server generates a new master secret and a new key block, and then also sends the finished message to the client. After the communication resume time Δd , secure communication is reinitiated.

2.2 Communication Resume Protocol Using Random Value

On the other hand, the protocol for a secure communication resume using random regeneration and retransmission is shown in Fig. 3. In this protocol, a new random is generated and sent in each secure communication resume, which results in the generation of a new key block in each communication resume. As with premaster secret regeneration and retransmission, this protocol also suffers from time delay, and a large BER.

Secure communication is performed for time Δt , and then data frame loss occurs. After realizing the data frame loss, the server requests a new random

Fig. 3. Conventional protocol using random value

for communication resume. The client generates a new random and includes it in a hello message. After the server receives the hello message from the client, it sends its own hello message that includes its new random. The server also generates a new key block using the new random and cached original master secret, and then generates new session keys and new IVs. This means that a resumed communication will use the same master secret as the previous one. Note that, although the same master secret is used, new random values are exchanged in the secure communication resume. These new randoms are taken into account in the new key block generation, which means that each secure communication starts up with different key materials: new session keys and new IVs.

$$New\ KB = PRF(Original\ MS, "KE", New\ CR + New\ SR) \quad (3)$$

Finally, the server sends the finished message to the client. The server generates a new key block, and then it also sends the finished message to the client. After communication resume time Δd , secure communication is reinitiated.

3 Proposed Secure Communication Resume Protocol

3.1 Proposed Protocol Using IV Count

To overcome the problems inherent in conventional secure communication resume protocols and to reinitiate secure communication much faster than they allow, we propose a fast, efficient, and secure communication resume protocol that uses an IV count value.

Fig. 4. Proposed protocol using IV count

Fig. 4 shows the proposed secure communication resume protocol, in which a count value of IV is sent to generate the new IVs in each secure communication. After realizing the data frame loss, the server requests a new count value of IV for communication resume. The client generates the new IV count value and sends its encryption data $E_{KUS}[\text{Count Value of IV}]$ encrypted with the server's public key to the server. It then generates new IVs using the count value. That is, the count value is used to generate new message protection materials, which means that each secure connection starts up with different IVs. Therefore, a resumed communication will use the same session keys as the previous communication. Note that, although the same session keys are used, new IVs are used in the secure communication resume. The client sends the change cipher spec and finished message to the server. The server generates new IVs using the received count value, and then sends the change cipher spec and finished message to the client. The client and server finally have the new IVs after communication resume time Δd .

$$I = I_0 + C, \quad 1 \leq C \leq 2^{IV \text{ Size}} - 1 \quad (4)$$

Here, I is the value of IV in each communication and I_0 represents the value of the original IV. C is a count value in each communication resume and is increased by a value of one for every communication resume.

3.2 Security Analysis

Security problems regarding attacks against the WAP WTLS were surveyed by Markku Juhani Saarinen [5], and it has been found that many of the changes that

were made by the WAP Forum have led to increased security problems [1]. In this paper, to determine the key refresh period for secure communication resume, the key refresh concept, which is referred by the WAP forum, was used and the condition of low bound was derived to avoid collisions from the birthday paradox [13]. By the birthday paradox, for strong collision resistance and a well-designed block cipher function with n bit input block size, it must hold that finding any pair $(x, y) \ni f(x) = f(y)$, takes $2^{n/2}$ trials. In the birthday bound, this means that even a perfect n bit block cipher function will start to exhibit collisions when the number of inputs nears the birthday bound $2^{n/2}$. Then, if coincidence exists, the problem of the plaintexts information issue occurs. Consequently, a new key is generated and updated before encrypting the $2^{n/4}$ input plaintext blocks.

$$T_{Key\ Refresh} = 2^{n/4} \quad (5)$$

On the other hand, IV resets after 2^{IVSize} . The probability of IV reset within the $2^{n/4}$ key refresh period is as small as the IV size is large, while the probability of its reset is as large as the IV size is small. For instance, if an IV size is 8 bytes, it resets after 2^{64} . This means that 8 bytes IV do not reset within the 2^{16} key refresh period, namely, $2^{64/4}$ input plaintext blocks in 64 bits block size. Therefore, if data frame loss occurs within the key refresh period, and if then a secure communication resume is required, we have only to generate and update a new IV for every communication resume instead of a new key generation. In addition, we have only to generate and update new keys at the time of key refreshing.

4 Performance Consideration

In this paper, to prove the efficiency of the proposed protocol, we compared and analyzed the transmission message size, the consumed time, and BER for communication resume of our proposed protocol with conventional protocols.

Table 1 shows a comparison of the transmission procedure and message sizes: CLT is the client, SVR is the server, the Change Cipher Spec/Finished message is CCS/F, V is WTLS version, and SID is session ID, R is random, and SI is a security association such as key exchange suit, cipher suit, compression method, etc. In the premaster secret protocol, the transmission messages sizes is about 46 bytes. In the case of random protocol, the transmission messages are about 86 bytes in size. However, in the proposed protocol, the transmission messages are composed only of the count value of IV, CLT CCS/F message, and SVR CCS/F message and their size is about 34 bytes. This Table shows that our proposed protocol allows the establishment of secure communication in an economic way, as it has fewer transmission message flows and smaller sizes than either the premaster secret or the random protocol.

To evaluate the efficiency of secure communication resume protocol, the BER in each protocol must also be considered. The results of BER in 3G at 384Kbps according to the number of communication resume are shown in Table 2. Here, T1 are the transmission bits at each 1 iteration, TC1 are the transmission bits

Table 1. Comparison of Δd and transmission message size

Protocol for Resume	Steps	Δd	Transmission Message Size
Premaster Secret	3	Premaster Secret	20 bytes
		CLT CCS/F	13 bytes
		SVR CCS/F	13 bytes
Random	4	[V, R, SID, SI] CLT Hello	30 bytes
		[V, R, SID, SI] SVR Hello	30 bytes
		CLT CCS/F	13 bytes
		SVR CCS/F	13 bytes
Proposed	3	IV Count Value	8 bytes
		CLT CCS/F	13 bytes
		SVR CCS/F	13 bytes

Table 2. BER in 3G according to the number of communication resumes (per hour)

Protocol for Resume		#1	#2	#4	#6	#8
Premaster Secret	T1	2.78×10^{-7}	5.56×10^{-7}	1.11×10^{-6}	1.67×10^{-6}	2.22×10^{-6}
	TC1	5.03×10^{-7}	1.11×10^{-6}	2.22×10^{-6}	3.33×10^{-6}	4.44×10^{-6}
	TC3	1.26×10^{-6}	2.78×10^{-6}	5.55×10^{-6}	8.33×10^{-6}	1.11×10^{-5}
Random	T1	4.72×10^{-7}	9.44×10^{-7}	1.89×10^{-6}	2.83×10^{-6}	3.78×10^{-6}
	TC1	9.72×10^{-7}	1.94×10^{-6}	3.89×10^{-6}	5.83×10^{-6}	7.78×10^{-6}
	TC3	2.97×10^{-6}	5.94×10^{-6}	1.19×10^{-5}	1.78×10^{-5}	2.38×10^{-5}
Proposed	T1	1.94×10^{-7}	3.89×10^{-7}	7.78×10^{-7}	1.16×10^{-6}	1.56×10^{-6}
	TC1	2.78×10^{-7}	5.56×10^{-7}	1.11×10^{-6}	1.67×10^{-6}	2.22×10^{-6}
	TC3	1.17×10^{-6}	2.33×10^{-6}	4.67×10^{-6}	7.00×10^{-6}	9.33×10^{-6}

at each 1 iteration with 50 % redundancy channel coding, and TC3 are the transmission bits at 3 iterations with 50 % redundancy channel coding. When computing the BER for 1 communication resume number per hour in the TC1 environment, the BERs in each protocol are provided: 5.03×10^{-7} in premaster secret protocol, 9.72×10^{-7} in random protocol, and 2.78×10^{-7} in the proposed protocol. This means that the proposed protocol reduces BER by over 45 % when compared with the premaster secret protocol, and by about 72 % when compared with the random protocol.

The consumed time and key refresh iteration is shown in Table 3. If a data frame loss occurs within the key refresh period, the proposed communication resume protocol is performed to reopen secure communication and a new key is generated and updated at the key refresh period. For example, at 100bps in 2G, CT is needed to 699 min and CTavg results in a key refresh of 2 iterations per day. Here, CT is consumed time for key refresh and CTavg is the average refresh iterations during one day.

Table 3. Consumed time and key refresh iteration in 2G, 3G bearer service environment (at cryptor input/output block size = 64 bits)

Consumed Time	2G at 100bps	2G at 9.6Kbps	3G at 14.4Kbps	3G at 384Kbps
CT	699 min	7.28 min	4.85 min	0.18 min
CTavg	2 iterations	198 iterations	297 iterations	7910 iterations

In the TC1 case in Table 4, the total consumed time using the proposed protocol for one day, 10.8 sec, is considerably less than the total consumed time using either the premaster secret protocol (14.7 sec) or the random protocol (27.4 sec).

Table 4. Total consumed time for communication resume for one day (at crypto input/output block size = 64 bits)

Protocol for Resume		2G at 100bps	2G at 9.6Kbps	3G at 14.4Kbps	3G at 384Kbps
Premaster Secret	T1	7.4 sec	7.5 sec	7.4 sec	7.9 sec
	TC1	14.7 sec	15.0 sec	15.1 sec	15.8 sec
	TC3	44.2 sec	45.5 sec	45.4 sec	39.6 sec
Random	T1	13.8 sec	13.9 sec	14.0 sec	13.5 sec
	TC1	27.4 sec	27.7 sec	28.1 sec	27.7 sec
	TC3	82.6 sec	85.1 sec	83.2 sec	84.6 sec
Proposed	T1	5.4 sec	5.5 sec	5.4 sec	5.5 sec
	TC1	10.8 sec	11.1 sec	10.7 sec	7.9 sec
	TC3	32.6 sec	33.7 sec	33.6 sec	33.2 sec

5 Conclusion

Most security research in wireless networks is focused on secured routing and transmitting in the network. However, because of the security issues in wireless networks, we suggest that the efficiency of security services is also an important issue. In this paper, a fast and secure communication resume protocol using IV count for wireless networks is presented and evaluated against the efficiency of conventional resume protocols. During the secure communication resume phases, we manage to reduce transferring traffic and thus also reduce the bandwidth on wireless networks. Moreover, our enhanced proposed protocol is able to reduce the consumed time or cryptographic load and the computations in order to reopen secure communication quickly.

Therefore, this proposed communication resume protocol provides a fast resume of secure communications, while reducing the transferring traffic, consumed time, and BER in a WTLS protocol environment.

References

1. WAP Forum. Wireless Transport Layer Security Spec. <http://www.wapforum.org>.
2. S. Jormalainen and J. Laine. Security in the WTLS. <http://www.tml.hut.fi/Opinnot/Tik-110.501/1999/papers/wtls.htm>.
3. R. Karri and P. Mishra. Optimizing the Energy Consumed by Secure Wireless Sessions-WTLS Case Study. *Mobile Networks and Applications*, No. 8, Kluwer Academic Publishers, 2003.
4. P. Mikal. WTLS : The Good and Bad of WAP Security. <http://www.advisor.com/Articles.nsf/aid/MIKAP001>, 2001.
5. M. J. Saarinen. Attacks against the WAP WTLS Protocol. <http://www.freeprotocols.org/harm0fWap/wtls.pdf>, 1999.
6. M. Badra et al.. A New Secure Session Exchange Key Protocol for Wireless Communication. *IEEE International Symposium on Personal, Indoor and Mobile Radio Communication*, 2003.
7. M. G. Rahman and H. Imai. Security in Wireless Communications. *Wireless Personal Communications*, No. 22, Kluwer Academic Publishers, 2002.
8. H. S. Jo and H. Y. Youn. A New Synchronization Protocol for Authentication in Wireless LAN Environment. *ICCSA '04*, LNCS publishers, 2002.
9. M. S. Hwang et al.. On the Security of an Enhanced Authentication Key Exchange Protocol. *AINA '04*, LNCS publishers, 2004.
10. J. Daemen, R. Govaerts, and J. Vandewalle. Resynchronization Weakness in Synchronous Stream Ciphers. *Pre-proceeding of EUROCRYPT'93*, 1993.
11. R. K. Nichols and P. C. Lekks. Wireless Security - Models, Threats, and Solutions. *McGraw-Hill Telecom*, 2002.
12. E. Amoroso. Fundamentals of Computer Security Technology. *PTR Prentice Hall*, Englewood Cliffs, New Jersey, 1993.
13. B. Schneier. *Applied Cryptography*, 2nd ed, John Wiley and Sons Inc., 1996.