A Secure and Efficient Communication Resume Protocol for Secure Wireless Networks

Kihong Kim[1], Jinkeun Hong[2], and Jongin Lim[3]

[1] National Security Research Institute,
161 Gajeong-dong, Yuseong-gu, Daejeon, 305-350, South Korea
hong0612@etri.re.kr
[2] Division of Information and Communication, Cheonan University,
115 Anse-dong, Cheonan-si, Chungnam, 330-740, South Korea
jkhong@cheonan.ac.kr
[3] Graduate School of Information Security, Korea University,
1, 5-Ka, Anam-dong, Sungbuk-ku, Seoul, 136-701, South Korea
jilim@korea.ac.kr

Abstract. There are important performance issues in secure wireless networks, such as mobility, power, bandwidth, and bit error rate (BER), that must be considered when designing a communication resume protocol. The efficiency of a secure session 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 using the initialization vector (IV) count for a secure wireless network is presented and evaluated against the efficiency of conventional resume protocols. Our proposed secure session resume protocol is found to achieve better performance, in terms of transmission traffic, consumed time, and BER, than conventional resume protocols with the same security capabilities.

1 Introduction

The wireless transport layer security (WTLS) provides privacy, authentication, and integrity in wireless application protocol (WAP) [1]. As the use of secure wireless networks becomes more widespread, the necessity of security for these networks is of increasing importance. However, in order to solve security issues in secure 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,

L. Kutvonen and N. Alonistioti (Eds.): DAIS 2005, LNCS 3543, pp. 157–168, 2005.

such as mobility, power, bandwidth, and BER, are very important. Of particular importance for a secure connection point is the efficiency of the secure session resume for the fast resume of secure communication.

In this paper, a protocol for fast secure session resume using IV count in secure 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 resume protocols with the same security capabilities. Of particular note is that the proposed protocol reduces consumed time by up to 60.6 %, compared with conventional protocols in a wireless network environment.

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

2 Conventional Key Handshaking Protocol in WTLS

The WTLS protocol determines the session key handshaking mechanism for secure services and transactions in secure 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. Through the change cipher spec phase, both the client and the server send the finished message, which is protected by a RP data unit that is applied by the negotiated security suites [6] [7]. The RP phase is a layered protocol phase that accepts raw data to be transmitted from the upper layer protocols. RP compresses and encrypts the data to ensure data integrity and authentication. It is also responsible for decrypting and decompressing data it receives and verifying that it has not been altered. In terms of secure wireless networks, WTLS requires fewer cryptographic computations, fewer resources, and less processing time than the secure sockets layer (SSL) protocol [1] [3].

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 conventional key handshaking protocol is illustrated in Fig. 1. The client sends a client hello message that includes information such as the version, session ID, 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.

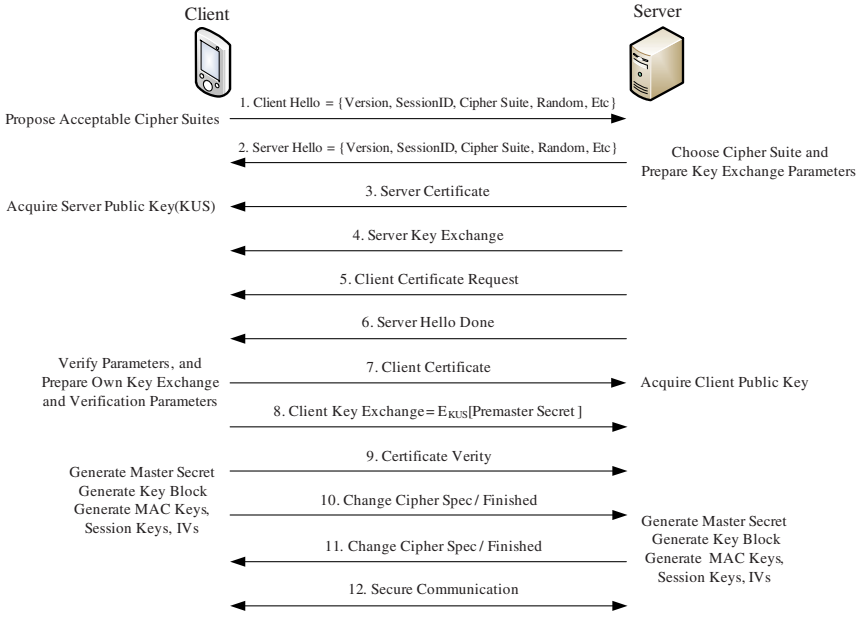

Fig. 1. Full handshaking process in WTLS protocol

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) and Eq. (2).

$$Master\ \ Secret\ =\ Pseudo\ Random\ Function(Premaster\ Secret, \quad (1)$$
$$''Master\ Secret'',\ Client\ Random\ +\ Server\ Random)$$

$$Key\ \ Block\ =\ Pseudo\ Random\ Function(Master\ Secret, \quad (2)$$
$$''Key\ Expansion'',\ Client\ Random\ +\ Server\ Random)$$

The client sends a change cipher spec message and proceeds directly to the finished message in order to verify that the key exchange and authentication process were successful. The server also generates MAC secrets, session keys, and IVs using the key block. Then it sends the finished message to the client. Finally, secure communication over the secure connection is established using session keys and IVs.

2.1 Protocol for Secure Session Resume Using Premaster Secret

After completion of the conventional full handshaking protocol shown in Fig. 1, a secure communication between client and server 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 session resume will be required. The aim of the session 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, which results in a new master secret and new key block, or 2) random regeneration and retransmission, in which random is only used to change the key block in each secure session resume.

Fig. 2 shows a protocol for a secure session resume using premaster secret regeneration and retransmission. In this protocol, a new premaster secret is generated and sent in each session 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 session 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 (including channel delay), and BER. This protocol is executed as follows. First, secure communication between the client and server is performed for time Δt, and then data

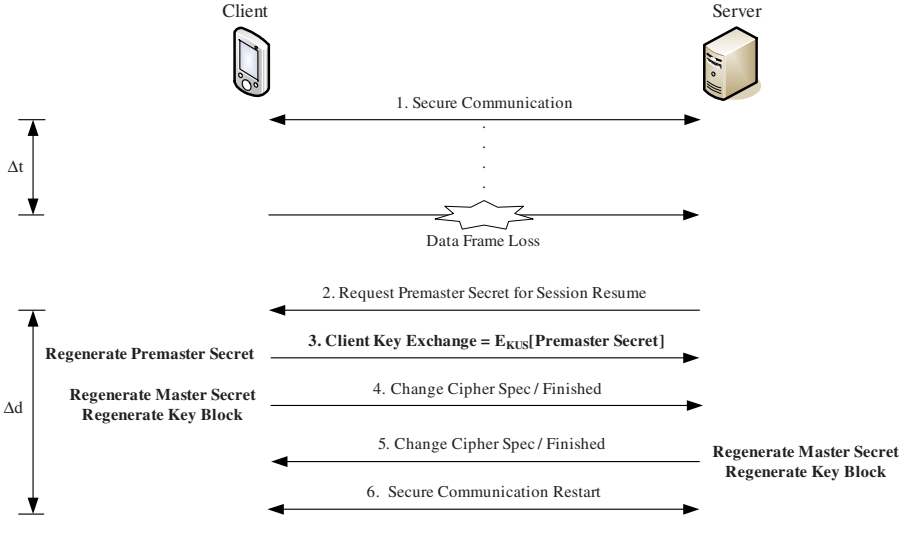

Fig. 2. Conventional protocol for secure session resume using premaster secret

frame loss occurs. After the server realizes the data frame loss, it requests a new premaster secret for session resume. The client generates a new premaster secret and sends $E_{KUS}[Premaster\ Secret]$. 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.

$$New\ Master\ Secret = Pseudo\ Random\ Function(New\ Premaster\ Secret, ''Master\ Secret'', Original\ Client\ Random + Original\ Server\ Random) \quad (3)$$

$$New\ Key\ Block = Pseudo\ Random\ Function(New\ Master\ Secret, ''Key\ Expansion'', Original\ Client\ Random + Original\ Server\ Random) \quad (4)$$

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 session resume time Δd shown in Fig. 2, secure communication is reinitiated.

2.2 Protocol for Secure Session Resume 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 session resume, which results in the generation of a new key block in each session resume. As with premaster secret regeneration and retransmission, this protocol also suffers from time delay, including channel 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 for session 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 session 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 connection starts up with different key materials: new session keys and new IVs.

$$New\ Key\ Block = Pseudo\ Random\ Function(Original\ Master\ Secret, ''Key\ Expansion'', New\ Client\ Random + New\ Server\ Random) \quad (5)$$

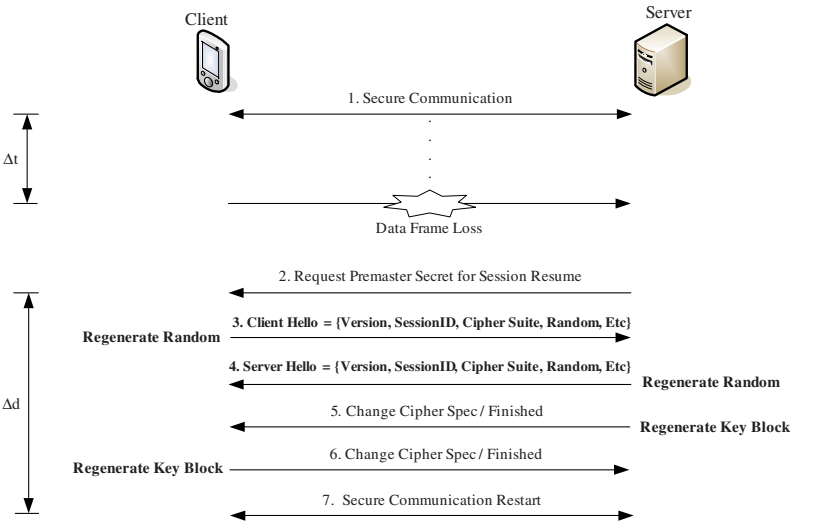

Fig. 3. Conventional protocol for secure session resume using random value

Finally, the server sends the finished message to the client. The server generates a new key block, resulting in new session keys and new IVs, and then it also sends the finished message to the client. After session resume time Δd shown in Fig. 3, secure communication is reinitiated.

3 Proposed Secure Communication Resume Protocol Using IV Count

3.1 Protocol for Proposed Secure Session Resume 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 new, efficient, and secure communication resume protocol that uses an IV count value.

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 session. After realizing the data frame loss, the server requests a new count value of IV for session resume. The client sends a new count value IV and 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 session will use the same session keys as the previous session. 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

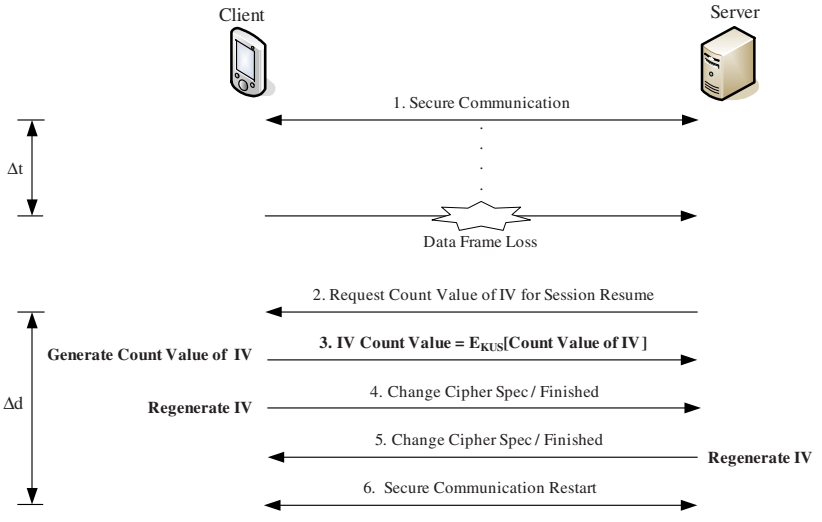

Fig. 4. Proposed protocol for secure session resume using IV count

to the client. The client and server finally have the new IVs after session resume time Δd as shown in Fig. 4 and as represented in Eq. (6).

$$\alpha = \alpha_0 + \nu, \quad 1 \leq \nu \leq 2^{IV\ Size} - 1 \tag{6}$$

Here, α is the value of IV in each session and α_0 represents the value of the original IV. ν is a count value in each session resume and is increased by a value of one for every session 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 session 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].

An internal state different from all previous sessions has to be chosen, to prevent the reuse of session keys or IVs. If two n bit ciphertexts, C_i and C_j, are arbitrarily chosen from ciphertext block $C = (C_1, C_2, \ldots, C_M)$, and provided that the input plaintext $P = (P_1, P_2, \ldots, P_M)$ in the cipher block chaining (CBC) mode [13] are equal, the following Eq. (7) is given.

$$\begin{aligned} C_i &= E_K(P_i \oplus C_{i-1}) = E_K(P_j \oplus C_{j-1}) = C_j \\ P_i \oplus C_{i-1} &= P_j \oplus C_{j-1}, \quad P_i \oplus P_j = C_{i-1} \oplus C_{j-1} \end{aligned} \tag{7}$$

Also, if the two ciphertexts, C_i and C_j, in cipher feed back (CFB) mode [13] are equal and if the two key stream blocks, O_i and O_j, in output feed back (OFB) mode [13] are also equal, these are represented as follows in Eq. (8).

$$\begin{aligned} E_K(C_i) \;=\; P_{i+1} \oplus C_{i+1} \;&=\; P_{j+1} \oplus C_{j+1} \;=\; E_K(C_j) \\ P_{i+1} \oplus P_{j+1} \;&=\; C_{i+1} \oplus C_{j+1} \\ O_i \;=\; E_K(O_{i-1}) \;=\; P_i \oplus C_i \;&=\; P_j \oplus C_j \;=\; E_K(O_{j-1}) \;=\; O_j \\ P_i \oplus P_j \;&=\; C_i \oplus C_j \end{aligned} \tag{8}$$

In other words, we acquire the information of plaintexts from the known ciphertexts. Therefore, the new keys are calculated and updated after a proper period to overcome the problematic plaintexts information issue. This key refresh period is computed using the birthday paradox. The number of pairs generated with ciphertext block $C = (C_1, C_2, \ldots, C_M)$ is $M(M-1)/2$, and the probability of at least one coincidence is shown in Eq. (9).

$$\begin{aligned} P \;&=\; 1 \;-\; \left(1 \;-\; \frac{1}{2^n}\right)^{\frac{M(M-1)}{2}} \;=\; 1 \;-\; \left(1 \;-\; \frac{1}{2^n}\right)^{-2^n \times \frac{M(M-1)}{2^{n+1}}} \\ &\approx 1 \;-\; e^{\frac{M(M-1)}{2^{n+1}}} \;\approx\; \frac{M(M-1)}{2^{n+1}} \end{aligned} \tag{9}$$

where n bit is the block size used in the block cipher. If M is about $2^{n/2}$, then at least one coincidence is found. 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. For example, in the case of DES, the maximum key refresh period is $2^{16}(2^{64/4})$ plaintext blocks.

$$T_{Key\ Refresh} \;=\; 2^{n/4} \tag{10}$$

Table 1. Result of collision probability and input/output block size

Block Size (n)	Number of Input Block (M)	Probability (P)
64 bits	2^8	1.77×10^{-15}
	$2^{16}(2^{64/4}$, key refresh period)	1.16×10^{-10}
	2^{32}	4.99×10^{-1}
128 bits	2^{16}	6.31×10^{-30}
	$2^{32}(2^{128/4}$, key kefresh keriod)	2.71×10^{-20}
	2^{64}	4.99×10^{-1}

Table 1 shows the relation of collision probability and block size in the bound of the birthday paradox.

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 session resume is required, we have only to generate and update a new IV for every session 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 and the consumed time for session resume of our proposed protocol with conventional protocols. The performance of the proposed session resume protocol has also been evaluated in terms of BER.

Table 2 shows a comparison of the transmission procedure and message sizes according to each protocol for secure session resume: 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.

As shown in Table 2, in the premaster secret protocol, the transmission messages consist of the premaster secret, client change cipher spec/finished message, and server change cipher spec/finished message. The transmission size of these messages is about 46 bytes. In the case of random protocol, the transmission messages are about 86 bytes in size and are composed of the client hello mes-

Table 2. Comparison of Δd and transmission message size according to each protocol

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 3. Consumed time to reopen secure session

Protocol for Resume		2G at 100bps	2G at 9.6Kbps	3G at 14.4Kbps	3G at 384Kbps
Premaster Secret	T1	3.7 sec	38 ms	25 ms	1 ms
	TC1	7.4 sec	76 ms	51 ms	2 ms
	TC3	22.1 sec	230 ms	153 ms	5 ms
Random	T1	6.9 sec	70 ms	47 ms	1.7 ms
	TC1	13.7 sec	140 ms	95 ms	3.5 ms
	TC3	41.3 sec	430 ms	280 ms	10.7 ms
Proposed	T1	2.7 sec	28 ms	18 ms	0.7 ms
	TC1	5.4 sec	56 ms	30 ms	1 ms
	TC3	16.3 sec	170 ms	113 ms	4.2 ms

sage, server hello message, client change cipher spec/finished message, and server change cipher spec/finished message. However, in the proposed protocol, the transmission messages are composed only of the count value of IV, client change cipher spec/finished message, and server change cipher spec/finished message and their size is about 34 bytes. This Table shows that our proposed session resume protocol allows the establishment of secure sessions in an economic way, as it has fewer transmission message flows and smaller sizes than either the premaster secret or the random protocol. This can be particularly advantageous in wireless networks where the radio bandwidth is bottlenecked.

To evaluate the efficiency of secure session resume protocol, the consume time needed for each protocol must also be considered. The results of the consumed time for session resume are shown in Table 3. For 2G and 3G in a bearer service environment, each protocol is serviced by 100bps (9.6Kbps), i.e., a minimum(maximum) bandwidth environment of 2G, and 14.4Kbps (384Kbps), i.e., a minimum(maximum) bandwidth environment of 3G. Here, T1 are the transmission bits at each 1 iteration, TC1 are the transmission bits at each 1 iteration with 50 % redundancy channel coding, and TC3 are the transmission bits at 3 iterations with 50 % redundancy channel coding.

In the TC1 environment, 2G at 100bps, if the protocol used for session resume is the premaster secret protocol, the consumed time for session resume is about 7.4 sec. In the case of random protocol, the consumed time is even higher about 13.7 sec. In the proposed protocol, however, the consumed time is only about 5.4 sec, proving that this protocol provides a faster secure session resume than the other resume protocols.

Table 4 shows results of BER in 3G according to the number of session resumes using the consumed time in Table 3. When computing the BER for 1 session 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.

Table 4. BER according to the number of session 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}

We can also see that BERs increases as the number of session resumes increases, and also that when the proposed protocol is used, the BERs are small than when the premaster secret or random protocol is used.

Fig. 5 shows a BER at the T1 environment shown in Table 4, demonstrating that the BERs from the proposed protocol are smaller than those from the premaster secret and random protocols.

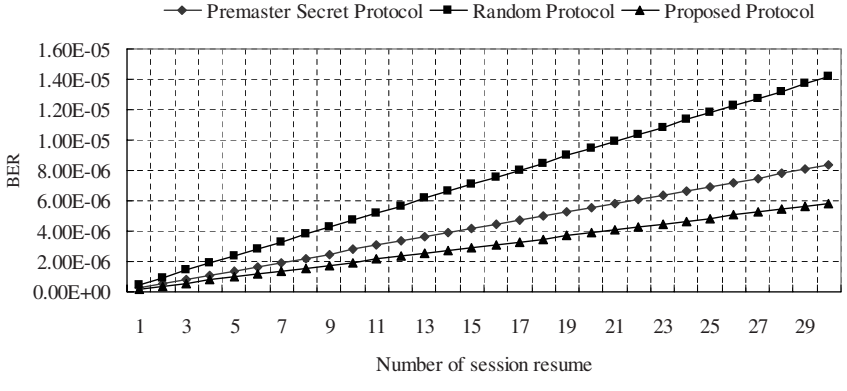

Fig. 5. BER at T1 environment in 384Kbps according to the number of session resumes

5 Conclusion

Most security research in secure wireless networks is focused on secured routing and transmitting in the network. However, because of the security issues in secure 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 session 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 sessions quickly.

Therefore, this proposed secure session resume protocol provides a fast resume of secure communications, while having the same security capabilities as other protocols and reducing the transferring traffic, consumed time, and BER in a WTLS protocol environment. In particular, this protocol reduces consumed time by up to 60.6 % when compared with conventional protocols.

References

1. WAP Forum. Wireless Transport Layer Security Spec. http://www.wapforum.org.
2. Sami Jormalainen, Jouni Laine. Security in the WTLS. http://www.tml.hut.fi/Opinnot/Tik-110.501/1999/papers/wtls.htm.
3. Ramesh Karri, Piyush Mishra. Optimizing the Energy Consumed by Secure Wireless Sessions-WTLS Case Study. *Mobile Networks and Applications*, No. 8, Kluwer Academic Publishers, 2003.
4. Philip Mikal. WTLS : The Good and Bad of WAP Security. http://www.advisor.com/Articles.nsf/aid/MIKAP001, 2001.
5. Markku-Juhani Saarinen. Attacks against the WAP WTLS Protocol. http://www.freeprotocols.org/harm0fWap/wtls.pdf, 1999.
6. Mohmad 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. Mohammad Ghulam Rahman, Hideki Imai. Security in Wireless Communications. *Wireless Personal Communications*, No. 22, Kluwer Academic Publishers, 2002.
8. Hea Suk Jo, Hee Young Youn. A New Synchronization Protocol for Authentication in Wireless LAN Environment. *ICCSA'04*, LNCS publishers, 2002.
9. Min Shiang Hwang et al.. On the Security of an Enhanced Authentication Key Exchange Protocol. *AINA'04*, LNCS publishers, 2004.
10. Joan Daemen, Rene Govaerts, Joos Vandewalle. Resynchronization Weakness in Synchronous Stream Ciphers. *Pre-proceeding of EUROCRYPT'93*, 1993.
11. Randall K. Nichols, Panos C. Lekks. Wireless Security - Models, Threats, and Solutions. *McGraw-Hill Telecom*, 2002.
12. Amoroso E. Fundamentals of Computer Security Technology. *PTR Prentice Hall*, Englewood Cliffs, New Jersey, 1993.
13. Bruce Schneier. *Applied Cryptography*, 2nd ed, John Wiley and Sons Inc., 1996.