Journal of the Korea Academia-Industrial cooperation Society
Vol. 15, No. 3 pp. 1691-1699, 2014

http://dx.doi.org/10.5762/KAIS.2014.15.3.1691
ISSN 1975-4701 / eISSN 2288-4688

Software Implementation of WAVE Security Algorithms

Jung-Ha Kang[1], Sung-Jin Ok[1], Jae Young Kim[2] and Eun-Gi Kim[1*]

[1]**Dept. of Information and Communication Engineering, Hanbat National University**

[2]**IT Convergence Technology Research Lab., Electronics and Telecommunications Research Institute**

WAVE 보안 알고리즘의 소프트웨어 구현

강정하[1], 옥성진[1], 김재영[2], 김은기[1*]

[1]한밭대학교 정보통신학과, [2]한국전자통신원구원 융합기술연구부문

Abstract IEEE developing WAVE specifications are able to support V2V and V2I wireless communications, and these functionalities can be used to enhance vehicle operational safety. To overcome any security weaknesses that are inherent in wireless communications, WAVE specification should support message encryption and authentication functions. In this study, we have implemented WAVE security algorithms in IEEE P1609.2 with openssl library and C language. We have verified the normal operation of implemented software, using the test vectors of related specifications, and measured their performance. Our software is platform independent, and can be used for the full implementation of WAVE specification.

요 약 IEEE에서는 V2I, V2V 등의 무선 통신 기능을 제공하여 차량 운행의 안전을 증대 시킬 수 있는 WAVE 규격을 정의하고 있다. WAVE 규격에서는 무선 통신이 갖는 보안 취약성을 극복할 수 있도록 메시지의 암호화 및 인증 기능을 지원하고 있다. 본 논문에서는 WAVE 규격에서 지원하고 있는 보안 알고리즘들을 openssl 라이브러리와 C 언어로 구현하였으며, 구현된 알고리즘들은 관련 규격들에서 제시하고 있는 테스트 벡터를 이용하여 정상 동작을 확인하고 성능을 측정하였다. 본 논문에서 구현된 보안 알고리즘들은 플랫폼에 독립적으로 구현되어, WAVE 보안 규격의 구현에 활용될 수 있을 것으로 생각된다.

Key Words : AES-CCM, ECDSA, ECIES, ITS Security, WAVE

1. Introduction

These days, the integration of IT and vehicle technologies is rapidly progressing.

Various applications have already been applied to vehicle communication systems. However, the problem of attacks - such as eavesdropping, spoofing and alterations on networking - must be resolved in order to safely activate these vehicle communication services. Users and designers of ITS also expect a continuity of confidence, integrity, and privacy protection for networking data and services.

IEEE has defined WAVE (Wireless Access in Vehicular Environments) standard, with regard to ITS technology. The purpose of WAVE standard is to improve vehicle safety, to reduce traffic congestion, to enable services for vehicle maintenance, and to provide the potential for new commercial services[1-4].

This research was financially supported by the Ministry of Education (MOE) and National Research Foundation of Korea(NRF) through the Human Resource Training Project for Regional Innovation (No. 201301590001) and a grant (12-TI-C01) from Advanced Water Management Research Program funded by Ministry of Land, Infrastructure and Transport of Korean government.

*Corresponding Author : Eun-Gi Kim(Hanbat National Univ.)
Tel: +82-42-821-1215 email: egkim@hanbat.ac.kr

Received December 27, 2013 Revised January 8, 2014 Accepted March 6, 2014

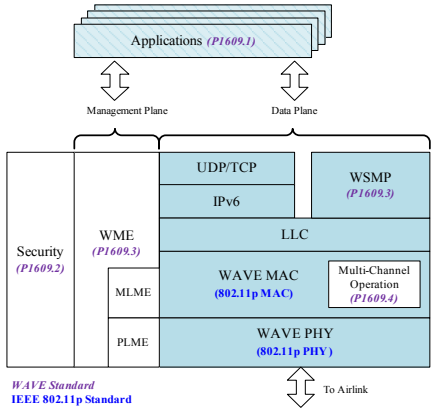
The WAVE system, proposed by IEEE standard as automotive networking technologies, supports V2V(Vehicle to Vehicle) and V2I(Vehicle to Infra-structure) communications. This WAVE system is intended for ITS services and applications for automotive safety at a high level within these network environments.

WAVE technology uses IEEE P 1609.2 with security which defines both a secure message format and secure processing procedures, for the use of WAVE devices.

1.1 WAVE system technologies

The WAVE system uses wireless communication technologies in high mobility environments in order to rapidly exchange frames for vehicle to vehicle or vehicle to infra-structure communication. The MAC (Media Access Control) and PHY(Physical) layers of the WAVE system are an amended version of the IEEE 802.11 standard, which supports wireless access in vehicular environments.

Vehicles need to be equipped with an end device platform, a networking module, and an antenna for WAVE communications. Also, devices must be installed on the road, with road side communication modules and antennas. The WAVE standard consists of IEEE 802.11p and IEEE 1609.4 as lower layers, IEEE 1609.3 for networking services, and IEEE 1609.2 for security services as an upper layer. The protocol layers of the WAVE system are illustrated in [Fig. 1].



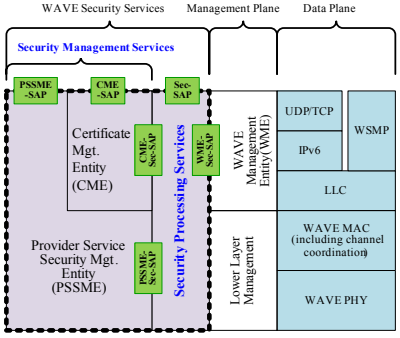
[Fig. 1] WAVE protocol stack

WAVE systems with this protocol architecture provide wireless communication technologies such as integrated type V2V and V2I. The WAVE system, using these functions, provides services for vehicle safety in emergency situations, as well as a wide range of other applications within the transportation environment. IEEE 802.11p, as a lower layer of the WAVE system, is based on IEEE 802.11 WLAN (Wireless Local Area Network). Thus, in a wireless communication environment, the WAVE system has security vulnerabilities. The IEEE 1609.2 standard has been defined to overcome these security weaknesses.

IEEE P1609.2 standard specifies a presentation language in order to define message formats, contents, and so on. This standard also defines message encryption, decryption and message authentication methods[5,6].

1.2 IEEE P1609.2 in WAVE system

IEEE P1609.2 specifies security services for applications and management messages in WAVE systems. The WAVE security services support confidentiality, authentication, authorization and integrity.



[Fig. 2] The structure of WAVE security services

The services and entities within WAVE security are illustrated in [Fig. 2][6]. The WAVE security services consist of Security Processing Services and Security Management Services. The Security Processing Services provide secure communications for data and WSAs(WAVE Services Advertisements). The Security Management Services provide Certificate Management Services and Provider Service Security Management Services. The Certificate Management Service manages information related to the validity of all certificates by CME (Certificate Management Entity). The Provider Service Security Management service manages

information related to certificates and private keys by PSSME (Provider Service Security Management Entity).

In order to provide WAVE Security Services, the cryptographic mechanisms (supported by P1609.2) include ECDSA (Elliptic Curve Digital Signature Algorithm), ECIES (Elliptic Curve Integrated Encryption Scheme) and AES-CCM (Advanced Encryption Standard - Counter with CBC-MAC) [6].

In WAVE security system, these security algorithms provide the following functions:

- Signature algorithms: ECDSA
- Public key encryption algorithms: ECIES
- Symmetric algorithms: AES-CCM

In this study, we have implemented security algorithms, as specified in IEEE P1609.2, using Openssl library and C language for application to automotive networking security. Our implemented security algorithms have been verified using the values of test vectors, as defined by the related specifications. We have also successfully accomplished performance testing with the implemented software.

This paper is organized as follows: Section 2 describes the implementation of ECDSA and the performance of the implemented ECDSA software. ECIES implementations and performance results are described in section 3, and AES-CCM implementations and performance results are described in section 4. Finally, section 5 concludes this paper.

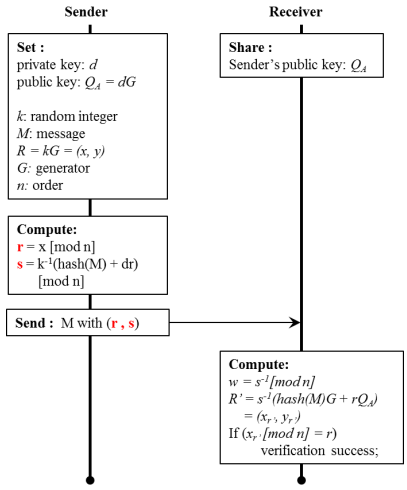
2. Implementation of ECDSA in IEEE P1609.2

ECDSA is the most widely standardized elliptic curve-based signature scheme, and appears in international standards such ANSI X9.62, FIPS 186-2, IEEE 1363-2000, ISO/IEC 15946-2 and SECG.

The ECDSA algorithm in WAVE standard provides assurance that any message from the sender is unaltered within WAVE networking.

2.1 The operation of ECDSA algorithm

Fig. 3 shows the operations of the ECDSA algorithm[7]. As shown in Fig. 3, the sender generates a random integer k, and computes r using k, private key, d, and ECDSA parameters (G, n). The calculated r is used to compute s with a message hash value, private key and k. The receiver computes using the r and s received from the sender, and confirms whether the $x_{r'} [\mathrm{mod}\, n]$, calculated by the receiver, matches up with r from the sender. Providing that $x_{r'} [\mathrm{mod}\, n]$ is the same as r, the message is unaltered within the network.



[Fig. 3] Operations of the ECDSA algorithm

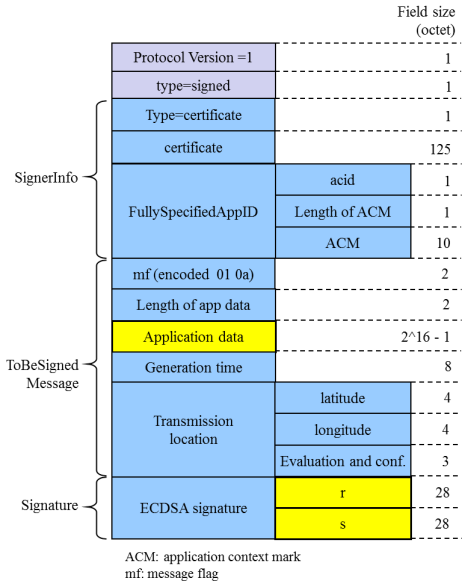
Detailed descriptions are as follows:

- Suppose the Sender wants to send a message M
  - Sender has a private key $d (d \in [1, n-1])$ and a public key $Q_A = dG$
  - G is a point on the curve (G is a generator) : $G \times \mathrm{order} = 0$ (point to infinity)
- To sign M, Sender computes each parameter in order
  - Compute $R = kG = (x, y)$
  - Compute $r \equiv x [\mathrm{mod}\, n]$
  - Compute $s \equiv k^{-1}(hash[M] + dr) [\mathrm{mod}\, n]$
  - The signature on M is the pair (r, s)
- Receiver's verification
  - Compute $w = s^{-1} [\mathrm{mod}\, n]$
  - Compute $R' = s^{-1}[hash(M)\, G + rQ_A] \rightarrow (x_{r'}, y_{r'})$
  - If "$x_{r'} [\mathrm{mod}\, n] = r$", the signature is accepted as valid. Otherwise, it is rejected.

2.2 The Implementation of ECDSA algorithm

ECDAS algorithm in IEEE P1609.2 standard follows the FIPS 186-2 specification, stating that the signature algorithm follows the ANSI X9.62 specification[7,8].

The public key algorithms used in the WAVE system include ecdsa_nistp224_with_sha224, ecdsa_nistp256_with_sha256, and so on. These algorithms' names relate to the applied elliptic curve and hash function. For instance, ecdsa_nistp224_with_sha224 uses the parameter specified in "ECDSA over a 224-bit prime field" in an X9.62 and 224 SHA scheme, as a hash algorithm. Fig. 4 illustrates the signed message structure that is exchanged in the WAVE communication.



[Fig. 4] Message structure of a ToBeSignedMessage

As described in Fig. 4, the sender sets the "Application data" of a ToBeSignedMessage structure to 'sending message', and transmits the message with ECDSA signature values, which are related to the message. The core functions of ECDSA implemented in the paper are the function for signature computing from sender-side, and the function for signature verification from receiver-side.

The functional prototypes are as follow:

■ The function to **compute an ECDSA signature value**:

- ECDSA_SIG ***ECDSA_p1609_do_sign** (const char *dgst, const EC_KEY *eckey, const BN_CTX *ctx);
- Return Value: ECDSA_SIG structure including r and s in [Fig. 3] and [Fig. 4].

■ The function to **verify an ECDSA signature value**

- int **ECDSA_p1609_do_verify**(const unsigned char *dgst, const int dgstLen, const ECDSA_SIG *sig, const EC_KEY *ecKey);
- Return Value: 1(Correct signature), 0(Incorrect signature), -1(Error)

Each function was implemented by using the openssl library according to the operation explaining in 2.1. In this study, we have verified the implemented functions, using parameters and test vectors from L.6.3.3 of ANSI X9.62 specification.

2.3 The performance of ECDSA

In this study, we have measured the performance of the ECDSA algorithms implemented. The test environments are as follows:

■ CPU: Intel(R) Core2 Duo CPU L9400 1.86GHz

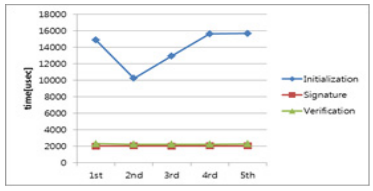
■ OS: Linux 3.4.2-1.fc16.i686.PAE

■ Openssl: Openssl-1.0.1

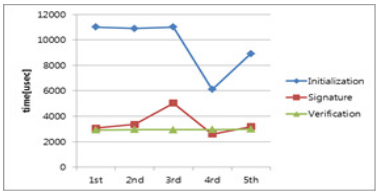
The test items for performance measurements consist of initialization time, signature time and verification time.

Initialization time is the interval of time taken to set up the values of init parameters. The signature time is the time interval taken to compute the signature value. The verification time is the time interval taken to calculate both the signature value, using the received parameters, and to check the signature's validation. The parameters and test vectors in L.6.3.3 of ANSI X9.62 specification, as mentioned in 2.2, were used for the tests. Using ECDSA_SHA224, testing was carried out 5 times for each item.

Fig. 5 shows the test results for performance measurements.



(a)



(b)

[Fig. 5] Test results for performance measurements of ECDSA
(a) The performance of ECDSA_SHA224
(b) The performance of ECDSA_SHA256

Fig. 5(a) shows the initialization time for setting up the values used in ECDSA_SHA224. The minimum time is 10,235usec. The maximum time is 15,695usec. Therefore, as a result of the initialization test for ECDSA_SHA224, the average time is calculated as 13,863usec.

Fig. 5(a) shows the signature time for computing r and s values, using ECDSA_SHA224. The minimum time is 2,009usec. The maximum time is 2,051usec. Therefore, as a result of the test of the signature of ECDSA_SHA224, the average time is 2,032usec.

Fig. 5(a) shows the signature verification time. The minimum time is 2,226usec. The maximum time is 2,299usec. Therefore, as a result of the verification of ECDSA_SHA224, the average time is 2,256usec We also carried out testing using ECDSA_SHA256, using the same methods as with ECDSA_SHA224. Each item's time graph is described in [Fig. 5](b).

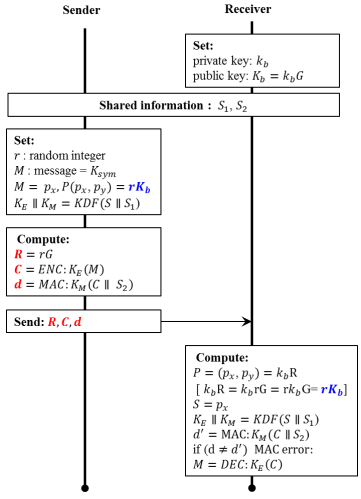
Fig. 5(b) shows the initialization time of ECDSA_SHA256 with microsecond units. The minimum time is 6,098usec. The maximum time is 11,007usec. Therefore, as a result of initialization test of ECDSA_SHA256, the average time is 9,582usec. Fig. 5(b) illustrates the signature time of ECDSA_SHA256, and Fig. 5(b) shows the verification time of ECDSA_SHA256. In Fig. 5(b), the minimum time is 2,589usec, the maximum time is 5,035usec, and the average time is 3,448usec.

In Fig. 5(b), the minimum time is 2,915usec, the maximum time is 2,978usec, and as a result of verification test , the average time is 2,946usec.

3. Implementation of ECIES in IEEE P1609.2

The ECIES algorithm in IEEE P1609.2 specification is used to send symmetric keys which are needed for the AES-CCM scheme.

3.1 The operation of ECIES



[Fig. 6] Operations of the ECIES algorithm

The operation of the ECIES algorithm is shown in [Fig. 6]. As described in [Fig. 6], the transmitter sends R, C and d, as the result of the ECIES encryption, to the receiver.

The detailed operations of the ECIES encryption and decryption scheme are as follows:

■ The receiver's private key k_b and public key $K_b = k_bG$

- Optional shared information: S_1 and S_2

■ To encrypt message M, the sender computes each parameter in the following order:

- Generate a random secret number $r (r \in [1, n-1])$ and calculates $R = rG$.
- Derive a shared secret $S = P_x$, where $P = (P_x, P_y) = rK_b$ (and $P \neq 0$)
- Use KDF (Key Derivation Function) to derive a symmetric encryption key, K_E, and MAC key K_M: $KDF(S \| S_1) = K_E \| K_M$
- Encrypt the message : $C = E(K_E; M)$
- Calculate the tag of the encrypted message and S_2 : $d = MAC(K_M; C \| S_2)$

- Output : $R \| C \| d$

■ To decrypt the cipher text, the receiver proceeds as follows:

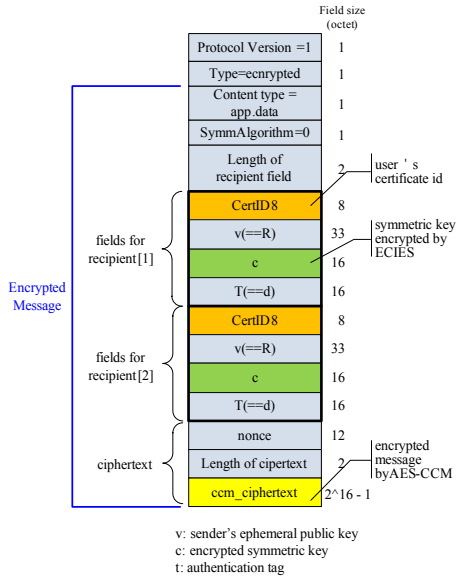
- Derive a shared secret: $S = P_x$, where $P = (P_x, P_y) = k_b R$ (the value, S, on the receiver side is the same as S derived on the sender side)
- Derive keys (K_E, K_M) in the same way as the sender : $KDF(S \| S_1) = K_E \| K_M$
- Use MAC to verify the tag and outputs: Failed if $d \neq MAC(K_M; C \| S_2)$
- Use the symmetric encryption scheme to decrypt the message $M = E^{-1}(k_E; C)$

3.2 The Implementation of the ECIES algorithm

The operation of ECIES, as specified in IEEE P1609.2, follows the IEEE std 1363a-2004 standard, where ECSVDP-DHC is used for the creation of a temporary secret key[2,9].

Message encryption uses the KDF2 scheme (non DHAES mode), based on a SHA-256 hash type. Message authentication is composed of a MAC1 scheme, based on SHA-256 hashing.

Fig. 7 shows part of an encrypted message structure.



[Fig. 7] Message structure of Encrypted Message

As described in Fig. 7, as the message is sent, it is encrypted by an AES-CCM scheme, using a symmetric key. The symmetric key is encrypted, according to the ECIES scheme, and transmitted as an encrypted message. In this paper, we have implemented the functions to provide the encryption and decryption of the data and key by ECIES. The procedure is as follows:

■ int **ecies_p1609_encrypt** (EC_POINT *partnerPubKey, EC_KEY *eckeySender, unsigned char *data, size_t length, unsigned char *p1, size_t p1Length, unsigned char *p2, size_t p2Length, struct vct *out);

■ int **ecies_p1609_decrypt** (EC_KEY *eckeyReceiver, unsigned char *p1, size_t p1Length, unsigned char *p2, size_t p2Length, struct vct *rcvdVct, unsigned char *out);

The ecies_p1609_encrypt function contains the input parameters for the operations of the ECIES algorithm and the output parameters of $v(=R)$, c and $T(=d)$, which are results of the ECIES algorithm.

The ecies_p1609_decrypt function checks the integrity of a message, using the values of v, c and T, from the received message and, provided it is not proved false, then performs the decryption of the message.

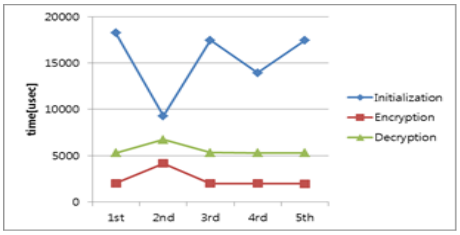
IEEE P1609.2 and related specifications do not provide any specific test vectors which would enable us to confirm the proper operations of ECIES.

Therefore, in this study, we have carried out testing to encrypt and decrypt the message using our own test vector. As a result, we found that our test vectors function normally with ECIES functions.

3.3 The performance of ECIES

We have tested the initialization time, encryption time and decryption time, in order to measure the performance of ECIES. The test environment is the same as described in 2.3.

There were no specified test vectors for the ECIES algorithms in IEEE P1609.2, so the test vectors as described in 3.2 were used. The testing was carried out 5 times using ECIES, and each item's time graph is described in [Fig. 8]. The initialization, encryption and decryption times of the ECIES are shown in Fig. 8 respectively.



[Fig. 8] Test results for ECIES performance measurements

As a result of the initialization test of ECIES, the minimum time is 9,270usec, the maximum time is 18,234usec, and the average time is 15,266usec.

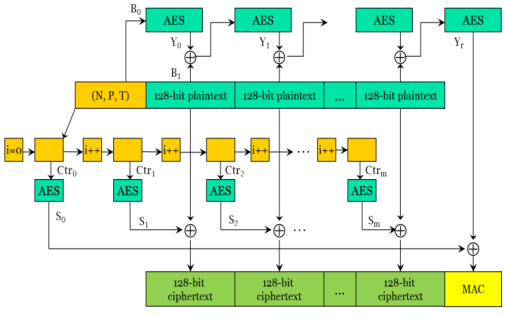
As a result of the encryption test of ECIES, the minimum time is 1,976usec, the maximum time is 4,165usec, and the average time is 2,430usec.

As a result of the decryption test of ECIES, the minimum time is 5,289usec, the maximum time is 6,724usec, and the average time is 5,584usec.

4. Implementation of AES-CCM in IEEE P1609.2

4.1 The operation of AES-CCM

The sender transmits the encrypted message to the receiver using an AES-CCM scheme to the receiver in IEEE P1609.2. Fig. 9 shows the operations of AES-CCM algorithms.



[Fig. 9] Operation of AES-CCM algorithm

The AES-CCM scheme divides user data into 128 bit units and then performs the encryption [10]. AES-CCM provides not only the message encryption function, but also a message authentication function. The parameters needed for the encryption of the AES-CCM scheme include the sending of the message P(plain text), nonce N, associated data A, and so on.

4.2 The Implementation of AES-CCM

The AES-CCM algorithm specified in IEEE P1609.2 follows NIST SP 800-38C standard [11].

When several sent messages are encrypted by the same symmetric key, each nonce value used for each message encryption must be different.

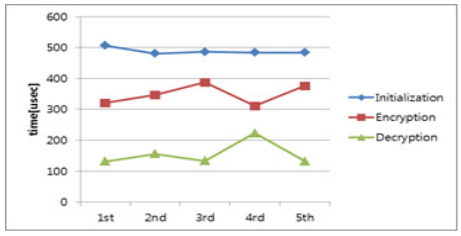
In this paper, we have implemented the encryption and decryption functions of AES-CCM scheme to be used with IEEE P1609.2 specification. The implemented functions are as follows.

- int **AES_CCM_p1609_Encrypt**(char *K, char *N, char *A, char *P, char *out, unsigned long *outlen);
- int **AES_CCM_p1609_Decrypt**(char *K, char *N, char *A, char *C, unsigned long ctlen, char *out, unsigned long *outlen);

Our testing has confirmed that the implemented functions work properly, using the test vectors in NIST SP 800-38C "Appendix C".

4.3 The performance of AES-CCM

We have measured the performance of AES-CCM implementation in a test environment, as shown in 2.3. C.3 Example 3 values in NIST SP 800-38C "Appendix C" were used as test vectors, and the test was carried out 5 times for each item. The test results are shown in [Fig. 10].



[Fig. 10] Test results for performance of AES-CCM

As a result of the initialization test of AES-CCM, the minimum time is 481usec, the maximum time is 507usec, and the average time is 488usec.

As a result of the encryption test of AES-CCM, the minimum time is 311usec, the maximum time is 387usec, and the average time is 348usec.

As a result of the decryption test of AES-CCM, the minimum time is 131usec, the maximum time is 223usec, and the average time is 155usec.

5. Conclusion

In this paper, we have implemented the security algorithms of IEEE P1609.2 standard, as specified by IEEE for the ITS system. The security algorithms implemented in this study have been verified using test vectors, defined in the related specifications. We have also carried out performance testing on the implemented software.

The security software implemented in this study follows IEEE P1609.2, defined in the year 2006[2]. IEEE currently defines IEEE P1609.2/D15pre as the most recent version [6].

The security algorithm of IEEE P1609.2/D15pre is the same as the algorithm of IEEE P1609.2/2006, so we can expect that the algorithms implemented in this paper can also be applied to the most recent version of IEEE P 1609.2 without any changes.

As a result of performance testing, we have shown that the average time of the ECDSA signature is approximately 3msec, the average time for ECIES encryption is approximately 2msec, and the average time for AES-CCM encryption is approximately 0.3msec. Network equipment or embedded systems can make use of security hardware modules because of this rapid performance. In the performance results of this study, we have found that the implemented security software module works competently with the wave system, providing that the hardware specification of the wave system corresponds to our test system, as described in 2.3. The security software module is more effective than the security hardware module in terms of cost, management, and so on. In the future, the study of adding service primitives, implementing functions of message encoding/decoding and building CA (Certificate Authority) of WAVE system needs to be carried out for the full implementation of IEEE P 1609.2.

References

[1] IEEE Std. 1609.1, IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)－ Resource Manager, 2006

[2] IEEE Std. 1609.2, “IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages”, 2006 DOI: http://dx.doi.org/10.1109/IEEESTD.2006.6636021

[3] IEEE Std. 1609.3, “IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)－ Networking Services”, 2007

[4] IEEE Std. 1609.4, “IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)－ Multi-channel Operation”, 2006

[5] EUN-GI KIM, HANBYEOG CHO, “SW Implementation of Security Algorithms in IEEE 1609.2”, *18’th ITS World Congress, Orlando, USA, 2011*

[6] IEEE P1609.2/D15pre, “Draft Standard for Wireless Access in Vehicular Environments – Security Services for Applications and Management Messages”, 2012

[7] NIST FIPS PUB 186-2, “DIGITAL SIGNATURE STANDARD (DSS)”, 2000

[8] ANSI X9.62, “Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)”, 2005

[9] IEEE Std. 1363a, “IEEE Standard Specifications for Public Key Cryptography: Additional Techniques”, 2004

[10] NIST FIPS 197, “Specification for the ADVANCED ENCRYPTION STANDARD(AES)”, 2001

[11] NIST Special Publication 800-38C, “Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality”, 2004

Jung-Ha Kang [Regular member]

- Aug. 2001 : Hanbat National Univ., Information and Communication Engineering, MS
- Jan. 2002 ~ Apr. 2012 : Fumate Co., Ltd. Principal Research Engineer
- May 2012 ~ current : Hanbat National Univ., Dept. of Information and Communication Engineering, Ph.D. candidate

<Research Interests>
Computer Network, Cryptography, Network Security

Sung-Jin Ok [Associate member]

- Feb. 2014 : Hanbat National Univ., Information and Communication Engineering, MS

<Research Interests>
Computer Network, Cryptography, Network Security

Jae Young Kim [Regular member]

- Feb. 1992 : Yonsei Univ., Electronic Engineering, MS
- Aug. 1996 : Yonsei Univ., Electronic Engineering, PhD
- Sep. 1996 ~ Feb. 1999 : Daewoo Electronics Ltd., Engineer
- Mar. 1999 ~ current : Electronics and Telecommunications Research Institute (ETRI), Principal Member of Research Staff

<Research Interests>
Energy IT, Security, Wireless Sensor Network

Eun-Gi Kim [Regular member]

- Feb. 1989 : Korea Univ., Electronic Engineering, MS
- Feb. 1994 : Korea Univ., Electronic Engineering, PhD
- Feb. 1995 ~ current : Hanbat National Univ., Dept. of Information and Communication Engineering, Professor

<Research Interests>
Computer Network, Embedded S/W, Cryptography, Network Security