An Efficient Verifiably Encrypted Signature from Weil Pairing

Jian Shen[1,2,3], Wenying Zheng[4], Jin Wang[1], Yuhui Zheng[1], Xingming Sun[1,3], Sungyoung Lee[5]
[1]Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, China
[2]Jiangsu Technology & Engineering Center of Meteorological Sensor Network, Nanjing University of Information Science & Technology, China
[3]School of Computer & Software, Nanjing University of Information Science & Technology, China
[4]School of Applied Meteorology, Nanjing University of Information Science & Technology, China
[5]Department of Computer Engineering, Kyung Hee University, Korea
{s_shenjian, zwy_sj}@126.com, wangjin@nuist.edu.cn, zhengyh@vip.126.com, sunnudt@163.com, sylee@oslab.khu.ac.kr

Abstract

A verifiably encrypted signature (VES) is a kind of very useful signature in online optimistic contract signing protocols. VES enables fair exchange between participants in signing protocols, which can allow the verifier to test that a given ciphertext is the encryption of a signature on a given message. In this paper, a novel efficient VES scheme which makes use of Weil pairing is proposed. The proposed signature scheme can provide good security properties such as validity, unforgeability, and opacity. We analyze the security and efficiency of the proposed scheme. Compared with the previous schemes, the proposed VES schemes is more efficient in terms of computational cost. In particular, the total computational cost of the three important phases of VES signing, VES verification, and adjudication in our scheme is decreased by at least 4*M*, where *M* indicates a scalar point multiplication. In addition, the signature size in our scheme is reduced to half of the size used in the previous schemes.

Keywords: Verifiably encrypted signature (VES), Weil pairing, Validity, Unforgeability, Opacity.

1 Introduction

A verifiably encrypted signature (VES) is used in optimistic contract signing protocols over Internet to provide fair exchange [1-3], which can protect the security of the exchange and keep well the fairness of the trading. A VES can allow the verifier to test that a given ciphertext is the encryption of a signature on a given message [10-11]. The realization of VES relies on a trusted third party, named adjudicator, which needs not to join the exchange protocol in on-line mode. For example, when a user Alice wants to sign a message for Bob but does not want Bob to possess her signature on the message immediately, Alice can encrypt her signature using the public key of an adjudicator and send the result to Bob along with a proof that she has given him a valid encryption of her signature. Bob can verify that Alice has signed the message but cannot deduce any information about her signature. At a later stage, Bob can either obtain the ordinary signature from Alice or resort to the adjudicator who can reveal Alice's signature.

A VES scheme gives rise to a new way to encrypt a signature with a designated public key and subsequently verify that the resulting ciphertext indeed contains such a signature. There are many applications of VES schemes, such as online contract signing, e-payment, and other electronic commerce.

Boneh et al. [4] first proposed a practical VES scheme as an application of their aggregate signature, which is based on a short signature structure. In [4], a trusted third party (adjudicator) is used to generate a pair of public key and private key. The public key of the adjudicator serves as a public parameter of the system and the corresponding private key kept secretly by the adjudicator is used to resolve any possible dispute between the two trading parties. It is worth noting that the VES verification in [4] requires three pairing computations. It is a large burden for the verifier. Recently, Ming and Wang [6] proposed an efficient VES scheme, where a user's public key and an adjudicator's public key both consist of three points of the cyclic group G_1. The computational cost in [6] is still a little high. Note that each public key generation requires three scalar point multiplications. In addition, the processes of the VES signing and the VES verification are complex, where a user needs to perform three scalar point multiplications and a verifier requires one pairing computation and three scalar point multiplications. In accordance with [7], the computations of the pairing and the point multiplication are both time-consuming. Hence, in the design of the VES scheme, we should reduce the scalar point multiplication and the pairing computation as many as possible.

In this paper we develop a more efficient VES scheme from Weil pairing. The security design of our protocol is based on the discrete logarithm problem (DLP). Our VES scheme provides good security properties, such as validity, unforgeability, and opacity. Moreover, our proposed VES signature is composed of only one point over the elliptic curves and has some improvement in terms of computational cost. Compared with the previous schemes,

*Corresponding author: Sungyoung Lee; E-mail: sylee@oslab.khu.ac.kr
DOI: 10.6138/JIT.2013.14.6.09

our scheme requires less computational complexity and less signature size. It is worth noting that the total computational cost of the three important phases of VES signing, VES verification, and adjudication in our scheme is decreased by at least 4*M*, where *M* indicates a scalar point multiplication. In addition, the signature size in our scheme can be reduced to half of the size used in the previous schemes.

2 Preliminaries

In this section, we briefly introduce Weil pairing which is necessary for a description of our signature scheme.

2.1 Weil Pairing

Let p be a prime number such that $p = 6q - 1$ for some prime number q and E a supersingular elliptic curve defined by the Weierstrass equation $y^2 = x^3 + 1$ over F_p. The set of rational points $E(F_p) = \{(x, y) \in F_p \times F_p : (x, y) \in E\}$ forms a cyclic group of order $p + 1$. Furthermore, because $p + 1 = 6q$ for some prime number q, the set of points of order q in $E(F_p)$ form a cyclic subgroup, denoted as G_1. Let $\mathcal{G}$ be the generator of G_1. Let G_2 be the subgroup of F_{p^2} containing all elements of order q. The modified Weil pairing [5] is a map: $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$, which has the following properties:

(1) Bilinear: For any $\mathcal{P}$, $\mathcal{Q} \in G_1$ and a, $b \in Z$, we have $\hat{e}(a\mathrm{P}, b\mathrm{Q}) = \hat{e}(\mathrm{P}, \mathrm{Q})^{ab}$.

(2) Non-degenerate: if $\mathcal{G}$ is a generator of G_1, then $\hat{e}(\mathcal{G}, \mathcal{G}) \in F_{p^2}^*$ is a generator of G_2.

(3) Computable: Given $\mathcal{P}$, $\mathcal{Q} \in G_1$, there is an efficient method to compute $\hat{e}(\mathrm{P}, \mathrm{Q}) \in G_2$.

3 Efficient Verifiably Encrypted Signature from Weil Pairing

There are three entities in our signature scheme: user, verifier and adjudicator. Our signature scheme runs in seven algorithms: *KeyGen*, *Sign*, *Verify*, *AdjKeyGen*, *VESSign*, *VESVerify*, and *Adjudication*. Note here that all the system parameters can be calculated by algorithm *gen* based on a security parameter input. Let $l \in Z^+$ be a security parameter. All algorithms run in probabilistic polynomial time with l as input. For concreteness, on input l, algorithm *gen* outputs a prime number q, the description of two groups G_1 and G_2 of order q, a generator $\mathcal{G}$ of G_1, and the description of a bilinear map $\hat{e}$: $G_1 \times G_1 \rightarrow G_2$. We denote the output of *gen* by $gen(1^l) = \{q, G_1, G_2, \mathcal{G}, \hat{e}\}$. The security parameter l is used to determine the size of q; for example, one could take q to be a random l-bit prime. We assume that the description of two groups G_1 and G_2 contains polynomial time algorithms for computing the group actions in G_1 and G_2. Similarly, we assume that the description of $\hat{e}$ contains a polynomial time algorithm for computing $\hat{e}$. In addition, we define a one-way hash function $H: \{0, 1\}^* \rightarrow Z_q^*$, where H maps its arbitrary length to a nonzero value in Z_q^*. The seven algorithms in our signature scheme are described in detail as follows:

- **KeyGen.** A user randomly picks $x \in Z_q^*$ as its private key. The user's public key is computed as: $x \cdot \mathcal{G} = P_{pub} = (x_p, y_p)$.
- **Sign.** Given the private key x of the user, the message $m \in (0, 1)^l$, the hash function H and the public key $P_{pub} = (x_p, y_p)$, the use signs a signature σ on m: $\sigma = \frac{H(m)}{x + x_p} \cdot \mathcal{G}$
- **Verify.** Given the public key $P_{pub} = (x_p, y_p)$, the message m and the signature σ, the verifier verifies whether $\hat{e}\left(\sigma, \frac{P_{pub} + x_p \cdot \mathcal{G}}{H(m)}\right) = \hat{e}(\mathcal{G}, \mathcal{G})$ can hold.
- **AdjKeyGen.** The adjudicator randomly picks $y \in Z_q^*$ as its private key, and computes the corresponding public key as $P_{pub}' = y \cdot \mathcal{G}$.
- **VESSign.** Given the user's private key x, the message m and the adjudicator's public key P_{pub}', the user computes the verifiably encrypted signature σ_{VES} as $\sigma_{VES} = \frac{H(m)}{x + x_p} \cdot P_{pub}'$.
- **VESVerify.** Given the verifiably encrypted signature σ_{VES}, the message m, the public key of user $P_{pub} = (x_p, y_p)$, and the public key of adjudicator P_{pub}', the verifier verifies whether $\hat{e}\left(\sigma_{VES}, \frac{P_{pub} + x_p \cdot \mathcal{G}}{H(m)}\right) = \hat{e}\left(P_{pub}', \mathcal{G}\right)$ can hold. If the above equation holds, the verifier can make sure that the user has given him a valid encryption of the original signature.
- **Adjudication.** Given the verifiably encrypted signature σ_{VES}, the message m, and the private key of adjudicator, the adjudicator extracts the original signature on the message m as: $\sigma = \frac{\sigma_{VES}}{y}$.

4 Security Analysis

Verifiably encrypted signatures require three important security properties: validity, unforgeability, and opacity. We show that our VES scheme satisfies the three security properties as follows:

4.1 Validity

Validity requires that verifiably encrypted signatures are able to be successfully verified as ordinary signatures and adjudicated verifiably encrypted signatures are also able to be successfully verified as ordinary signatures. This means $VESVerify(m, VESSign(m)) = 1$ and $Verify(m, Adjudication(\mathrm{V}ESSign(m))) = 1$ hold for all m and for all properly generated key pairs and adjudicator key pairs. Note here that "1" indicates the verification is

successful.

For a verifiably encrypted signature σ_{VES} on a message m, the validity is proven as follows:

$$\begin{aligned}\hat{e}\left(\sigma_{VES}, \frac{P_{pub}+x_P\cdot\mathcal{G}}{H(m)}\right) &= \hat{e}\left(\frac{H(m)}{x+x_p}\cdot P_{pub}, \frac{P_{pub}+x_P\cdot\mathcal{G}}{H(m)}\right)\\ &= \hat{e}\left(\frac{H(m)}{x+x_p}\cdot P_{pub}, \frac{x\cdot\mathcal{G}+x_p\cdot\mathcal{G}}{H(m)}\right)\\ &= \hat{e}\left(\frac{H(m)}{x+x_p}\cdot P_{pub}, \frac{x+x_p}{H(m)}\cdot\mathcal{G}\right)\\ &= \hat{e}\left(P_{pub}', \mathcal{G}\right)\end{aligned}$$

Hence, $VESVerify(m, VESSign(m)) = 1$ holds. On the other hand,

$$\begin{aligned}\hat{e}\left(\sigma, \frac{P_{pub}+x_p\cdot\mathcal{G}}{H(m)}\right) &= \hat{e}\left(\frac{H(m)}{x+x_p}\cdot\mathcal{G}, \frac{P_{pub}+x_p\cdot\mathcal{G}}{H(m)}\right)\\ &= \hat{e}\left(\frac{H(m)}{x+x_p}\cdot\mathcal{G}, \frac{x\cdot P+x_p\cdot\mathcal{G}}{H(m)}\right)\\ &= \hat{e}\left(\frac{H(m)}{x+x_p}\cdot\mathcal{G}, \frac{x+x_p}{H(m)}\cdot\mathcal{G}\right)\\ &= \hat{e}\left(\mathcal{G},\mathcal{G}\right)\end{aligned}$$

Therefore, $Verify(m, Adjudication(VESSign(m))) = 1$ holds.

4.2 Unforgeability

Unforgeability requires that it is hard for an attacker to forge a valid VES.

Definition 1. Given access to a VES signing oracle S, an adjudication oracle A, and a hash oracle H, the advantage of an algorithm F in forging a VES is:

$$\mathbf{Adv\ VESForge}_F \underline{\underline{def}} \mathbf{Pr}\left[\begin{array}{l} VESVerify\left(P_{pub}, P_{pub}', m, \sigma_{VES}\right) = \mathbf{valid}:\\ \left(P_{pub}, x\right) \underline{R}\ \ KeyGen\\ \left(P_{pub}', y\right) \underline{R}\ \ AdjKeyGen\\ \left(m, \sigma_{VES}\right) \underline{R}\ \ F^{H,S,A}\left(P_{pub}, P_{pub}'\right)\end{array}\right]$$

The probability is taken over the coin tosses of the key generation algorithms, of the oracles, and of the forger. In addition, the forger must not previously have queried either oracle at m.

Definition 2. A VES forger $F(t, q_H, q_S, q_A, \varepsilon)$-forges a VES if F runs in time at most t, makes at most q_H queries to the hash function, at most q_S queries to the VES signing oracle S, at most q_A queries to the adjudication oracle A, and $\mathbf{Adv\ VESForge}_F$ is at least ε. A VES scheme is $(t, q_H, q_S, q_A, \varepsilon)$-secure against existential forgery if no forger $(t, q_H, q_S, q_A, \varepsilon)$-breaks it.

In order to proof that the proposed VES scheme is secure against existential forgery, we need to define a basic signature scheme, which is only composed of *KeyGen*, *Sign*, and *Verify*. The basic signature scheme is similar to the signature scheme proposed in [9], which is provably secure against existential forgery. Hence, in this paper, we claim that our basic scheme is secure against existential forgery.

Theorem 1. Suppose that the basic signature scheme is $(t', q_H', q_S', q_A', \varepsilon')$-secure against existential forgery. Then the proposed VES scheme is $(t, q_H, q_S, q_A, \varepsilon)$-secure against existential forgery for all $q_H \le q_H'$, $q_S \le q_S'$, $\varepsilon \ge \varepsilon'$, and $t \le t' - (q_S + q_A + 1) \cdot t_1 - (q_A + 1) \cdot t_2$, where t_1 indicates the time for point multiplication in G_1 and t_2 implies the time for inversion in Z_q^*.

Proof of Theorem 1. Given a VES forger algorithm F, we construct a forger algorithm F' for the basic signature scheme. The basic scheme forger F' is given a public key P_{pub}', and has access to a signing oracle for P_{pub} and a hash oracle. It simulates the challenger and interacts with F as follows.

- **Setup.** Algorithm F' randomly picks $y \in Z_q^*$ and computes $P_{pub}' = y \cdot \mathcal{G}$. Note here that (y, P_{pub}') serves as the adjudicator's key pair. Now F' runs F, providing as input the public keys P_{pub} and P_{pub}'.
- **Hash Queries.** When algorithm F requests a hash on a message m, algorithm F' makes a query on m to its own hash oracle and receives a value $h \in Z_q^*$. Then F' responds to F with h.
- **VESSign Queries.** When algorithm F requests a VES signature on a message m, algorithm F' queries its own signing oracle (for P_{pub}) on m to obtain an ordinary signature $\sigma \in G_1$, and returns to F with $y \cdot \sigma$.
- **Adjudication Queries.** When algorithm F requests an adjudication for a VES σ_{VES} on a message m under the public keys P_{pub} and P_{pub}', algorithm F' checks that the VES σ_{VES} is valid and returns $\frac{\sigma_{VES}}{y}$ to F.
- **Output.** Finally, algorithm F outputs a forged and valid VES σ_{VES}^* on a message m^* with non-negligible probability. Note here that F must not have made any query to the VES signing oracle at m^*. In addition, F' computes $\sigma^* = \frac{\sigma_{VES}^*}{y}$ as a valid basic signature on .

Algorithm F' succeeds whenever algorithm F does, that is, with probability at least ε. The running time of algorithm F' is equivalent to the running time of algorithm F plus the time it takes to respond to q_H hash queries, q_S VES signing queries, q_A adjudication queries, and the time

to transform the final VES forgery of algorithm F into a basic signature forgery. Hash queries impose no overhead. Each VES signing query requires F' to perform one point multiplication in G_1. Each adjudication query requires F' to perform one inversion in Z_q^* and one point multiplication in G_1. The output phase also requires one inversion in Z_q^* and one point multiplication in G_1. We assume that one point multiplication in G_1 takes time t_1 and one inversion in Z_q^* takes time t_2. Hence, the total running time of F' is at most $t + (q_S + q_A + 1) \cdot t_1 + (q_A + 1) \cdot t_2$. If $F(t, q_H, q_S, q_A, \varepsilon)$-forges a VES, then $F'(t + (q_S + q_A + 1) \cdot t_1 + (q_A + 1) \cdot t_2, q_H, q_S, \varepsilon))$-forges a basic signature. Conversely, if the basic signature scheme is $(t', q_H', q_S', \varepsilon')$-secure against existential forgery, then the proposed VES scheme is $(t' - (q_S + q_A + 1) \cdot t_1 - (q_A + 1) \cdot t_2, q_H', q_S', q_A, \varepsilon')$-secure against existential forgery.

4.3 Opacity

Opacity requires that it is hard for an attacker, given a VES, to **extract** an ordinary signature on the same message.

Theorem 2. Suppose that the basic signature scheme is secure against existential forgery and the DLP is hard in G_1. Then the proposed VES is secure against extraction.

Proof of Theorem 2. Given a VES σ_{VES} for a message m, if an adversary E wants to compute the ordinary signature σ on the message m, then E either directly forges the signature σ on the message m under the signer's public key P_{pub} or extracts the signature σ from the VES σ_{VES}. From Theorem 1, we know that the basic signature scheme is $(t', q_H', q_S', \varepsilon')$-secure against existential forgery. Hence, no attacker can $(t', q_H', q_S', \varepsilon')$-forges the signature σ on the message m. In addition, since the DLP is hard in G_1, it is impossible for an attacker to obtain y given P_{pub}'. We have $\sigma = \frac{\sigma_{VES}}{y}$. Therefore, no attacker can extract the signature σ from the VES σ_{VES}. Finally, we claim that the proposed VES is secure against extraction.

5 Performance Analysis

The proposed scheme has some improvement in the computational efficiency. In accordance with [7], we know that the computations of the pairing and the point multiplication are both time-consuming. Hence, in the design of the VES scheme, we should reduce the scalar point multiplication and the pairing computation as many as possible in order to improve the computational efficiency. We compare our proposed signature scheme with previously used schemes [4][6] in Table 1. Note here that M means a scalar point multiplication and e indicates a pairing computation. Suppose that the size of one point in G_1 of order q is 160 bits [8], it is clear that the size of our VES is only 160 bits, while the sizes of the VESs in [4][6] are both 320 bits. As described above, there are seven algorithms in our scheme, including *KeyGen*, *Sign*, *Verify*, *AdjKeyGen*, *VESSign*, *VESVerify*, and *Adjudication*. The costs of the first three algorithms in our scheme are almost the same as in the previous schemes. Hence, we focus only on comparing the cost of *VESSign*, *VESVerify*, and *Adjudication*. It is worth noting that, in our scheme, we can pre-compute $\hat{e}(P_{pub}', \mathcal{G})$ in *VESVerify* phase. Therefore, there is only one pairing computation. From Table 1, we can conclude that our scheme is more efficient than the previous schemes.

Table 1 Performance Comparison

	Boneh's scheme	Ming's scheme	Our scheme
Size	320 bits	320 bits	160 bits
VESSign	3*M*	3*M*	1*M*
VESVerify	3*e*	1*e* + 3M	1*e* + 1*M*
Adjudication	1*M*	1*M*	1v

6 Conclusion

A VES is a very important and useful cryptographic primitive, which can convince the verifier that a given ciphertext is the encryption of a signature on a given message. In online optimistic contract signing protocols, a VES can provide fair exchange between participants. Thus, it is usually used as a building block to construct an optimistic fair exchange. In this paper, we proposed a novel VES scheme based on Weil pairing. We have shown that our scheme can provide good security properties like validity, unforgeability, and opacity. In addition, compared with the previous schemes, the proposed VES scheme has some improvement in terms of computational cost.

Acknowledgments

This work is supported by the research fund from Nanjing University of Information Science and Technology under Grant No.S8113003001, the National Science Foundation of China under Grant No.61300237, No.61300238 and No.61232016, the National Basic Research Program 973 under Grant No.2011CB311808, the research fund from Jiangsu Technology & Engineering Center of Meteorological Sensor Network in NUIST under Grant No.KDXG1301, the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) under Grant No.2011-0030823, the Natural Science Foundation of Jiangsu Province under Grant No.BK2012461, the 2013 Nanjing Project of Science and

Technology Activities for Returning from Overseas, and the PAPD fund.

References

[1] N. Asokan, Victor Shoup and Michael Waidner, *Optimistic Fair Exchange of Digital Signature*, *IEEE Journal on Selected Areas in Communications*, Vol.18, No.4, 2000, pp.593-610.

[2] Feng Bao, Robert H. Deng and Wenbo Mao, *Efficient and Practical Fair Exchange Protocols with Offline TTP*, *Proceedings of the 1998 IEEE Symposium on Security and Privacy*, Oakland, CA, May, 1998, pp.77-85.

[3] Xiao-Long Xu, Jing-Yi Xiong and Chun-Ling Cheng, *Fair Exchange Mechanism for P2P Systems*, *Proceedings of the 2010 Ninth International Conference on Grid and Cloud Computing*, Nanjing, China, November, 2010, pp.235-239.

[4] Dan Boneh, Craig Gentry, Ben Lynn and Hovav Shacham, *Aggregate and Verifiably Encrypted Signatures from Bilinear Maps*, *Proceedings of the 22nd International Conference on Theory and Applications of Cryptographic Techniques (Eurocrypt' 03)*, Warsaw, Poland, May, 2003, pp.416-432.

[5] Dan Boneh and Matt Franklin, *Identity-Based Encryption from the Weil Pairing*, *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology (Crypto' 01)*, Santa Barbara, CA, August, 2001, pp.213-229.

[6] Yang Ming and Yumin Wang, *An Efficient Verifiably Encrypted Signature Scheme without Random Oracle*, *International Journal of Network Security*, Vol.8, No.2, 2009, pp.125-130.

[7] Paulo S. L. M. Barreto, Hae Y. Kim, Ben Lynn and Michael Scott, *Efficient Algorithms for Pairing-Based Cryptosystems*, *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology (Crypto' 02)*, Santa Barbara, CA, August, 2002, pp.354-368.

[8] Paulo S. L. M. Barreto and Michael Naehrig, *Pairing-Friendly Elliptic Curve of Prime Order*, *Proceedings of the 20th Annual ACM Symposium on Applied Computing, LNCS 3897*, Santa Fe, NM, March, 2005, pp.319-331.

[9] Fangguo Zhang, Reihaneh Safavi-Naini and Willy Susilo, *An Efficient Signature Scheme from Bilinear Pairings and Its Applications*, *Proceedings of the 7th International Workshop on Theory and Practice in Public Key Cryptography, LNCS 2947*, Singapore, March, 2004, pp.77-290.

[10] Wenzhong Guo, Naixue Xiong, Han-Chieh Chao, Sajid Hussain and Guolong Chen, *Design and Analysis of Self-adapted Task Scheduling Strategy in Wireless Sensor Networks*, *Sensors*, Vol.11, No.7, 2011, pp.6533-6554.

[11] Lei Shu, Manfred Hauswirth, Han-Chieh Chao, Min Chen and Yan Zhang, *NetTopo: A Framework of Simulation and Visualization for Wireless Sensor Networks*, *Ad Hoc Networks*, Vol.9, No.5, 2011, pp.799-820.

Biographies

Jian Shen received the PhD degree in Computer Science from Chosun University, Gwangju, Korea, in 2012. Since late 2012, he has been a Professor in the School of Computer and Software at Nanjing University of Information Science and Technology, Nanjing, China. His research interests include computer networking and security systems.

Wenying Zheng received the ME degree in Electronic Engineering from Chosun University, Gwangju, Korea, in 2009. Since late 2012, she has been a faculty member in the School of Applied Meteorology at Nanjing University of Information Science and Technology. Her research interests include image security, image recognition, and security systems.

Jin Wang received the PhD degree in Computer Engineering from Kyung Hee University, Korea, in 2010. Now, he is a Professor in the School of Computer and Software at Nanjing University of Information Science and Technology. His research interests include routing protocol and algorithm design.

Yuhui Zheng is an Associate Professor in the School of Computer and Software, Nanjing University of Information Science and Technology. His research interest covers image processing, pattern recognition, and remote sensing image restoration.

Xingming Sun is the Dean of the School of Computer and Software, Nanjing University of Information Science and Technology and the Director of Jiangsu Provincial Center for Network Security Engineering. His research interests include network and information security, cloud computing, wireless sensor networks, and information hiding.

Sungyoung Lee is a Professor in the Department of Computer Engineering, Kyung Hee University, Korea, since 1993. His current research focuses on Ubiquitous Computing and Applications, Wireless Ad-hoc and Sensor Networks, Context-aware Middleware, Sensor Operating Systems, Real-Time Systems and Embedded Systems.