ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.4, April 2006, pp.907–914
DOI: 10.1360/jos170907

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于 Weil 对的成对密钥协商协议*

姚 刚+, 冯登国

(信息安全国家重点实验室(中国科学院 软件研究所),北京 100080)

Pairwise Key Agreement Protocols Based on the Weil Pairing

YAO Gang+, FENG Deng-Guo

(State Key Laboratory of Information Security (Institute of Software, The Chinese Academy of Sciences), Beijing 100080, China)

+ Corresponding author: Phn: +86-10-82612787, Fax: +86-10-62520469, E-mail: g.yao@is.iscas.ac.cn

Yao G, Feng DG. Pairwise key agreement protocols based on the weil pairing. *Journal of Software*, 2006,17(4): 907−914. http://www.jos.org.cn/1000-9825/17/907.htm

Abstract: To achieve security in the networks, it is important to be able to encrypt and authenticate messages sent between the users. Keys for encryption and authentication purposes must be agreed upon by the users in the networks. Three new pairwise key agreement protocols based on Weil pairing are proposed in this paper. In those protocols, all the users share common secret information. They may arrange the pairwise key and authenticate each other by fewer messages. The proposed protocols have the security attributes such as known session key security, perfect forward secrecy, no key-compromise impersonation, no unknown key-share and no key control.
Key words: key agreement; Weil pairing; session key; authentication

摘 要: 为了实现网络安全,一个重要的方法是网络用户传送加密和可鉴定的消息.此时,用来加密和鉴定的密钥应该由网络中的用户协商得到.提出了 3 个基于 Weil 对的成对密钥协商协议.在协议中,所有用户共享一个秘密信息,通过较少的步骤,同时实现密钥协商和用户认证.提出的协议满足如下的安全特性:部分密钥泄漏的安全性、完备的前向安全性、个人密钥泄漏的安全性、无不明的密钥共享和无法控制密钥等.
关键词: 密钥协商;Weil 对;会话密钥;鉴定
中图法分类号: TP309 文献标识码: A

1 Introduction

Key establishment protocol is process whereby a shared secret key becomes available to participating entities, typically for subsequent use as symmetric keys for a variety of cryptographic purposes including encryption, message authentication, and entity authentication. Key establishment may be broadly subdivided into key transport and key agreement protocols[1]. In a key transport protocol, one entity creates or otherwise obtains a secret value, and securely transfers it to the other entity. In key agreement, a shared secret is derived by all the parties in a group

* Supported by the National Natural Science Foundation of China under Grant No.60273027 (国家自然科学基金); the National Grand Fundamental Research 973 Program of China under Grant No.G1999035802 (国家重点基础研究发展规划(973))
Received 2004-04-06; Accepted 2005-06-22

as a function of information contributed by, or associated with, each of these, such that no party in the group can predetermine the resulting value.

As the first practical solution to key agreement problem, the Diffie-Hellman key agreement protocol[1] enables two parties, never having met in advance or shared keying material, to establish a shared secret by exchanging messages over an open channel. The security of this protocol is based on the assumption of the difficulty of the discrete logarithm arithmetic and the Diffie-Hellman Decision problem.

It is generally desired that each party in a key establishment protocol be able to determine the true identity of the others which could possibly gain access to the resulting key, implying preclusion of any unauthorized additional parties from deducing the same key. However, the Diffie-Hellman key agreement protocol does not authenticate the two communication parties, hence suffers from the "man-in-the-middle" attack. Key authentication is the property whereby one party is assured that no other party aside from a specifically identified second party may gain access to a particular secret key. The goal of any authenticated key establishment protocol is to establish keying data. Ideally, the established key should have precisely the same attributes as a key established face-to-face. However, it is not an easy task to identify the precise security requirements of authenticated key establishment.

A secure key agreement protocol is desired to have the following attributes[1,2]:

• *Known-Key security*: The protocol should still achieve its goal in the face of an adversary who has learned some other session keys—unique secret keys which each run of a key agreement protocol between A and B should produce.

• *Forward secrecy*: If the long-term keys or the secret key of some parties are compromised, the secrecy of past session keys are not compromised.

• *Key-Compromise impersonation*: When A's private key is compromised, it may be desirable that this event does not enable an adversary to impersonate other entities to A.

• *Unknown key-share*: Entity B cannot be coerced into sharing a key with entity A without B's knowledge.

• *No key control*: The secret session key between any two users is determined by both users taking part in, and none of the two users can influence the outcome of the secret session key, or enforce the session key to fall into a pre-determined interval.

In this paper, we propose three new pairwise key agreement protocols which are based on Weil pairing. The protocols have the attributes such as known session key security, perfect forward secrecy, no key-compromise impersonation, no unknown key-share and no key control. The remainder of this paper is organized as follows: Section 2 gives the properties of Weil pairing and reviews some key agreement protocols; Section 3 presents our pairwise key agreement protocols which are based on Weil pairing; Section 4 gives the performance and the security analysis of our protocols; Finally, Section 5 concludes this paper.

2 Related Work

In this section, we briefly describe the basic definition and properties of the Weil pairing, and then review the research on key agreement protocols.

2.1 Weil pairing

Let E be an elliptic curve over a base field F. Let G_1 be a cyclic additive group generated by P whose odder is a prime q, and G_2 be a multiplicative cyclic group of the same order q. We assume that the discrete logarithm problems in both G_1 and G_2 are hard. The Weil pairing[3−5] is defined by a bilinear map e: $G_1 \times G_1 \to G_2$, where G_1 corresponds to the additive group of points of $E(F)$, and G_2 corresponds to the multiplicative group of an extension field of F. Let P, Q, R in G_1. The Weil pairing e has the following properties:

1. *Non-Degenerate*: There exists a P in G_1, such that $e(P,P)\neq 1$.
2. *Bilinearity*: For all P, Q, R in G_1, $e(P+Q,R)=e(P,R)e(Q,R)$, $e(P,Q+R)=e(P,Q)e(P,R)$.
3. *Non-Degeneracy*: If $e(P,Q)=1$ for all Q in G_1, then $P=O$, where O is a point at infinity.
4. *Computability*: There is an efficient algorithm to compute $e(P,Q)$ for all P, Q in G_1.

The *bilinear Diffie-Hellman* (*BDH*) *problem* for a bilinear pairing e: $G_1\times G_1\rightarrow G_2$ is defined as follows: given P, aP, bP, cP in G_1, compute $e(P,P)^{abc}$, where a, b and c are randomly chosen from Z_q^*. An algorithm A is said to solve the BDH with an advantage ε if $\Pr[A(P,aP,bP,cP)=e(P,P)^{abc}]\geq\varepsilon$.

BDH Assumption: We assume that the BDH problem is hard, which means that there is no polynomial time algorithm to solve the BDH problem with non-negligible probability.

2.2 Key agreement protocol

Diffie-Hellman key agreement is a fundamental technique providing unauthenticated key agreement. After Diffie-Hellman key agreement protocol is put forward, lots of key agreement protocols are proposed[1], such as Encrypted Key Exchange (EKE) protocol[6], JV multi-party key agreement protocol[7], two pass authenticated key agreement protocol[2], MTI/A0 key agreement protocol[8], tripartite key agreement protocol[9], ID-based authenticated two pass key agreement protocol[5], etc.

The Encrypted Key Exchange (EKE) protocol is designed by Bellovin and Merritt[6]. It provides security and authentication on computer networks, using both symmetric and public-key cryptography in a novel way: A shared secret key is used to encrypt a randomly generated public key. This protocol is secure against active attacks, and has the property that the password is protected against off-line “dictionary” attacks. A multi-party key agreement protocol is provided by Just and Vaudenay[7]. The protocols provide key authentication, key confirmation and forward secrecy. Three two-pass authenticated key agreement protocols are proposed by Song and Kim[2]. One of these protocols is presented in asymmetric setting, which is based on Diffie-Hellman key agreement working over an elliptic curve group, and the other two protocols are modifications of this protocol. The MTI/A0 key agreement protocol is presented by Matsumoto et al. in 1986[8]. It is designed to provide implicit key authentication. The security of this protocol is based on the Diffie-Hellman problem in elliptic curve group. In 2000, Joux shows how to implement an elegant tripartite key agreement protocol using pairing[9]: Only one broadcast is required for each party. This protocol has found very good applications to broadcast network. However, just like the Diffie-Hellman protocol, Joux’s protocol does not attempt to authenticate the three communicating parties, and is vulnerable to “man-in-the-middle” attacks. Smart describes an ID-based authenticated two-pass key agreement protocol which makes use of the Weil pairing[5]. The protocol requires a trusted key generation center, and has a property that the key generation center is able to recover the agreed session keys from the message flows and its secret key.

3 Key Agreement Protocols Based on Weil pairing

This section describes the pairwise key agreement protocols which is based on Weil pairing. The security of these protocols is based on the bilinear Diffie-Hellman problem in elliptic curve group. All protocols in this paper have been described in the setting of the group of the points on an elliptic curve defined over a finite field.

Let E be an elliptic curve over a base field F and G_1 be a cyclic additive group generated by P in $E(F)$ whose odder is a prime q. Let G_2 be a multiplicative cyclic group of the same order q. The Weil pairing is defined by a bilinear map e: $G_1\times G_1\rightarrow G_2$. Let H: $\{0,1\}^*\rightarrow Z_q^*$ be a cryptographic hash function. Suppose that all the users in the networks agree on a symmetric key algorithm, such as the Advanced Encryption Standard (AES). And denote the symmetric encryption and decryption with respect to a secret key k by $E_k(\cdot)$ and $D_k(\cdot)$, respectively.

3.1 Protocol 1

We assume that all the members in the networks have a common point Q in G_1. Let A and B be two entities in the networks who are going to agree to some session keys. They can perform the following protocol:

1. A chooses a random number a in Z_q^* and sends aP to B;
2. B chooses a random number b in Z_q^* and sends bP to A;
3. A computes $K_A=e(bP,Q)^a$ and sends $t_A=E_{K_A}(aP)$ to B;
4. B computes $K_B=e(aP,Q)^b$ and sends $t_B=E_{K_B}(bP)$ to A.

The last two steps are to authenticate that A and B are really the established the session key. If $bP=D_{K_A}(t_B)$, then A saves K_A as the session key between A and B. Similarly, if $aP=D_{K_B}(t_A)$, then B saves K_B as the session key between A and B. It is easy to see that $K_A=e(bP,Q)^a=e(P,Q)^{ab}=e(aP,Q)^b=K_B$. Thus, after A and B perform the protocol, they get the session key $K=e(P,Q)^{ab}$. When all the pairwise session keys have been constructed, the common secret Q can be erased.

After a period of time, if A and B want to update their session key, they may perform the following protocol:

1. A chooses a random number a' in Z_q^* and sends $a'P$ to B;
2. B chooses a random number b' in Z_q^* and sends $b'P$ to A;
3. A computes $K'_A = e(b'P, H(K)P)^{a'}$ and sends $t'_A = E_{K'_A}(a'P)$ to B;
4. B computes $K'_B = e(a'P, H(K)P)^{b'}$ and sends $t'_B = E_{K'_B}(b'P)$ to A.

For user A, if $b'P= D_{K'_A}(t'_B)$, he saves K'_A as the new session key between A and B. Similar verification can been done by user B. After A and B perform the update protocol, they can get the session key $K'= e(P,P)^{a'b'H(K)}$ and the old session key K can be erased.

3.2 Protocol 2

We assume that all the members in the networks have a common point Q in G_1. Let A and B be two entities in the networks who are going to agree to some session keys. They can perform the following protocol:

1. A chooses a random number a in Z_q^* and computes $R_A=H(aP)aQ$. Then he sends $\{aP,R_A\}$ to B;
2. B chooses a random number b in Z_q^* and computes $R_B=H(bP)bQ$. Then he sends $\{bP,R_B\}$ to A.

After user A receives the message $\{bP,R_B\}$, he verifies whether $e(bP,H(bP)Q)=e(P,R_B)$ is hold or not. If the equation is hold, he saves $K_A=e(a(bP),R_B)^{H(aP)a}$ as the session key between A and B. Otherwise, he discards the message $\{bP,R_B\}$. Similarly, after user B receives the message $\{aP,R_A\}$, he verifies whether $e(aP,H(aP)Q)=e(P,R_A)$ is hold or not. If the equation is hold, he saves $K_B=e(b(aP), R_A)^{H(bP)b}$ as the session key between A and B. Otherwise, he discards the message $\{aP,R_A\}$. It is easy to see that $K_A=e(a(bP),R_B)^{H(aP)a}= e(P,Q)^{H(aP)H(bP)a^2b^2} =e(b(aP),R_A)^{H(bP)b}$ $=K_B$. Thus, after A and B perform the protocol, they can get the session key $K= e(P,Q)^{H(aP)H(bP)a^2b^2}$. When all the pairwise session keys have been constructed, the common secret Q can be erased.

After a period of time, if A and B want to update their session key, they may perform the following protocol:

1. A chooses a random number a' in Z_q^* and computes $R'_A =H(a'P)a'H(K)P$. Then he sends $\{a'P, R'_A\}$ to B;
2. B chooses a random number b' in Z_q^* and computes $R'_B =H(b'P)b'H(K)P$. Then he sends $\{b'P, R'_B\}$ to A.

For user A, if $e(b'P, H(b'P)H(K)P)=e(P, R'_B)$, he saves the $K'_A = e(a'(b'P), R'_B)^{H(a'P)a'}$ as the new session key between A and B. Similar verification can been done by user B. After user A and user B perform the update protocol,

they get the new session key $K'=e(P,P)^{(a')^2(b')^2H(a'P)H(b'P)H(K)}$ and the old session key K can be erased.

3.3 Protocol 3

We assume that all the members in the networks have two common point Q, S in G_1. Furthermore, we assume that Q and S satisfy the condition: there exists an s in Z_q^* such that $Q=sP$ and $S=s^2P$. Let A and B be two entities in the networks who are going to agree to some session keys. They can perform the following protocol:

1. A chooses a random number a in Z_q^* and sends $\{aP,aQ\}$ to B;
2. B chooses a random number b in Z_q^* and sends $\{bP,bQ\}$ to A.

After user A receives the message $\{bP,bQ\}$, he verifies whether $e(bP,S)=e(Q,bQ)$ is hold or not. (If the message $\{bP,bQ\}$ is generated by B, $e(bP,S)=e(bP,s^2P)=e(sP,bsP)=(Q,bQ)$.) If the equation is hold, he saves $K_A=e(bQ,S)^a$ as the session key between A and B. Otherwise, he discards the message $\{bP,bQ\}$. Similarly, after user B receives the message $\{aP,aQ\}$, he verifies whether $e(aP,S)=e(Q,aQ)$ is hold or not. If the equation is hold, he saves $K_B=e(aQ,S)^b$ as the session key between A and B. Otherwise, he discards the message $\{aP,aQ\}$. It is easy to see that $K_A=e(bQ,S)^a=e(P,P)^{abs^3}=e(aQ,S)^b=K_B$. Thus, after A and B perform the protocol, they can get the session key $K=e(P,P)^{abs^3}$. When all the pairwise session keys have been constructed, the common secret Q, S can be erased.

After a period of time, if A and B want to update their session key, they may perform the following protocol:

1. A chooses a random number a' in Z_q^* and sends $\{a'P,a'H(K)P\}$ to B;
2. B chooses a random number b' in Z_q^* and sends $\{b'P,b'H(K)P\}$ to A.

For user A, if the equation $e(b'P,H(K)^2P)=e(H(K)P,b'H(K)P)$ is hold, he saves $K'_A=e(b'H(K)P,H(K)^2P)^{a'}$ as the new session key between A and B. Similar verification can been done by user B. After A and B perform the update protocol, they get the session key $K'=e(P,P)^{a'b'H(K)^3}$ and the old session key K can be erased.

4 Performance and Security Analysis

We evaluate the proposed key agreement protocols in terms of communication, computation and security.

4.1 Performance

Assume that a group of users want to transfer massages in the network. To achieve security in the networks, it is important to be able to encrypt an authenticate massages sent among the users. In order to authenticate a user is the member of the group, a solution is to let all the members share a secret. Any pair of members can use this global secret to achieve key agreement and obtain a new pairwise key.

Table 1 Performance parameters of the proposed protocols

	Secret (bits)	Communication (bits)		Computation	
		Agreement phase	Update phase	Agreement phase	Update phase
Protocol 1	$2l_1$	$2l_1+l_2$	$2l_1+l_2$	$M+W+exp+2Enc$	$M+H+W+exp+2Enc$
Protocol 2	$2l_1$	$4l_1$	$4l_1$	$3M+H+3W+exp$	$3M+3H+3W+exp$
Protocol 3	$4l_1$	$4l_1$	$4l_1$	$2M+3W+exp$	$2M+H+3W+exp$

Here, we suppose that the point in the group can be presented in $2l_1$ bits, and the element in the group G_2 can be presented in l_2 bits. Let M be the operation of computing multiple points in the group G_1, H the operation of computing hash function, W the operation of computing Weil pairing value of two points, exp the operation of

computing an exponentiation in the group G_2, and *Enc* the operation of computing encryption.

4.2 Security analysis

In the following, we briefly outline the attributes of our protocol:

Passive attacks:

In protocol 1, if the adversary knows the massage transformed between two users A and B, i.e., he knows aP, bP, he cannot compute the secret session key $K=e(P,Q)^{ab}$ because he does not know the shared secret Q. Even if the adversary knows Q, he cannot get the K for he cannot obtain the secret number a chosen by A or b chosen by B. In the update phase, if the adversary knows the massage transformed between two users A and B, i.e., he knows $a'P$, $b'P$, he cannot compute the secret session key $K'=e(P,P)^{a'b'H(K)}$, for he does not know a', b', and the shared session key K.

In protocol 2, if the adversary knows the massage transformed between two users A and B, i.e., he knows aP, $H(aP)aQ$, bP, $H(bP)bQ$, he cannot compute the secret session key $K=e(P,Q)^{H(aP)H(bP)a^2b^2}$, for he does not know the shared secret Q. Even if the adversary knows the common secret Q, he still cannot compute the secret session key shared between A and B, for he cannot obtain the secret number a chosen by A or b chosen by B. In the update phase, if the adversary knows the massage transformed between two users A and B, i.e., he knows $a'P$, $b'P$, $H(a'P)a'H(K)P$, $H(b'P)b'H(K)P$, he cannot compute the secret session key $K'=e(P,P)^{(a')^2(b')^2H(a'P)H(b'P)H(K)}$, for he does not know a', b', and the shared session key K.

In protocol 3, if the adversary knows the massage transformed between two users A and B, i.e., he knows aP, aQ, bP, bQ, he cannot compute the secret session key $K=e(P,P)^{abs^3}$, for he do not know the shared secret Q. Even if the adversary knows the common secret Q and S, he still cannot compute the secret session key shared between A and B, for he cannot obtain the secret number a, or b. In the update phase, if the adversary knows the massage transformed between two users A and B, i.e., he knows $a'P$, $a'H(K)P$, $b'P$, $b'H(K)P$, he cannot compute the secret session key $K'=e(P,P)^{a'b'H(K)^3}$, for he does not know a', b', and the shared session key K.

Known-Key security:

If an adversary has obtained some session keys in the networks, it is still difficult for him to get the unknown session keys. For the users in the networks uniformly choose the points in the group, the session keys shared between different pairs are independent.

For example, in protocol 2, if the adversary knows the massage transformed between two users A and B, i.e., he knows aP, $H(aP)aQ$, bP, $H(bP)bQ$, he cannot compute the secret session key $K=e(P,Q)^{H(aP)H(bP)a^2b^2}$, for he cannot obtain the secret number a, or b. Even if the adversary knows the secret Q, he still cannot compute the secret session key shared between A and B.

Forward secrecy:

If an adversary has obtained the current session key used between two users A and B in the networks, it is still difficult for him to get the previous session keys between A and B. After A and B perform the key update protocol, they get the new session key and erase the old session key. So, the adversary cannot get the previous session in a direct way. Although he knows the relationship between the new session key and the old session key, he cannot get

the old session key through this relationship.

For example, in protocol 3, the new session key between A and B is $K'=e(P,P)^{a'b'H(K)^3}$, where K is the old session key. If the adversary wants to compute K, he needs to solve the discrete logarithm and find an inverse image of the hash function. This is difficult for the adversary by our assumption.

Key-Compromise impersonation:

The compromise of one user's private session key does not imply that the private session keys of other users will also be compromised in our protocols. The adversary may impersonate the compromised user in the subsequent protocols since he knows the private session key of the compromised user, but he cannot impersonate other users.

For example, in protocol 2, if the adversary knows all the private session key belong to a user C, he cannot obtain the private session key shared between A and B. The adversary knows aP, $H(aP)aQ$, bP, $H(bP)bQ$, he cannot compute the secret session key $K=e(P,Q)^{H(aP)H(bP)a^2b^2}$, for he cannot obtain the secret number a chosen by A or b chosen by B. Even if the adversary knows the common secret Q, he still cannot compute the secret session key shared between A and B. This attribute is still hold for key update phase.

Unknown key-share:

If the adversary wants to convince a user A in the networks that he shares a session key with the adversary, while in fact he shares the key with another user B, the adversary is required to know the common point shared by all the user at the beginning or the private session key of the user B. Otherwise, the attack hardly works.

For example, in protocol 3, if an adversary wants to share a session key with a user B, while B believes that he shares the session key with a user A, he needs to send B a message containing $\{aP,aQ\}$. Since the adversary does not know the point Q, he may guess a random point to construct the massage. When B receives the massage, he verifies whether $e(aP,S)=e(Q,aQ)$ holds or not. Since Q is a random chosen point, this equation will hold with negligible probability. This attribute is still hold for key update phase because the adversary has no information about the previous session key shared between the user A and user B.

No key control:

In our protocols, the secret session key between any user A and user B is determined by both the user A and user B, and none of the two users can influence the outcome of the secret session key, or enforce the session key to fall into a pre-determined interval.

For example, in protocol 1, the session key between A and B is $K=e(P,Q)^{ab}$, where a is chosen by user A and b is chosen by user B. Since user A does not know the value of secret number chosen by B, he cannot determine how to select the number a, such that the private session key is equal to the pre-determined value, or falls into a pre-determined interval.

5 Conclusion

The key agreement protocols result in shared secret session key between the users in the networks. In this paper, we propose three new key agreement protocols that emphasize their security and performance. The protocols use a common shared secret to authenticate that the user is the legal user in the networks, and the protocols have the security attributes such as known session key security, perfect forward secrecy, no key-compromise impersonation, no unknown key-share and no key control.

References:

[1] Menezes AJ, Oorschot PC, Vanstone SA. Handbook of Applied Cryptography. New York: CRC Press, 1997.

[2] Song B, Kim K. Two-Pass authenticated key agreement protocol with key confirmation. In: Roy BK, Okamoto E, eds. Proc. of the Indocrypt 2000. Berlin, Heidelberg: Springer-Verlag, 2000. 237−249.

[3] Boneh D, Franklin M. Identity-Based encryption from the Weil pairing. In: Kilian J, ed. Advances in Cryptology—Crypto 2001. Berlin, Heidelberg: Springer-Verlag, 2001. 213−229.

[4] Boneh D, Lynn B, Shacham H. Short signatures from the Weil pairing. In: Boyd C, ed. Advances in Cryptology—Asiacrypt 2001. Berlin, Heidelberg: Springer-Verlag, 2001. 514−532.

[5] Smart NP. An identity based authenticated key agreement protocol based on the Weil pairing. Electronics Letters, 2002,38(13): 630−632.

[6] Bellovin SM, Merritt M. Encrypted key exchange: Password-Based protocols secure against dictionary attacks. In: Cooper D, ed. Proc. of the 1992 IEEE Symp. on Security and Privacy. IEEE Computer Society Press, 1992. 72−84.

[7] Just M, Vaudenay S. Authenticated multi-party key agreement. In: Kim K, Matsumoto T, eds. Advances in Cryptology—Asiacrypt'96. Berlin, Heidelberg: Springer-Verlag, 1996. 36−49.

[8] Matsumoto T, Takashima Y, Imai H. On seeking smart public-key distribution systems. Trans. of the IECE of Japan, 1986,E69(2): 99−106.

[9] Joux A. A one round protocol for tripartite Diffie-Hellman. In: Bosma W, ed. Proc. of the 4th Algorithmic Number Theory Symp. Berlin, Heidelberg: Springer-Verlag, 2000. 385−394.

YAO Gang was born in 1975. He is an assistant staff of Institute of Software, Chinese Academy of Sciences. His current research areas are cryptography, computer security, automata theory, etc.

FENG Deng-Guo was born in 1965. He is a professor and doctoral supervisor at the Institute of Software, Chinese Academy of Sciences. His current research areas are telecommunication, computer security, network security, information system, etc.