Zero-configuration Identity-based IP Network Encryptor

Sammy H.M. Kwok, *Member*, IEEE, Hayden K.H. So, *Member,* IEEE,
Edmund Y. Lam, *Senior Member*, IEEE, K.S. Lui, *Senior Member*, IEEE

Abstract — *For corporations or individuals who wish to protect the confidentiality of their data across computer networks, network-layer encryption offers an efficient and proven method for preserving data privacy. Network layer encryption such as IPSec is more flexible than higher layer solutions since it is not application-dependent and can protect all end-to-end traffics that go between two hosts. Using IPSec, two hosts must first establish a session key through message exchanges before they can communicate. In this paper, we present an Identity Based Encryption (IBE) scheme that allows a host to calculate the per-packet encryption key based on the IP address of the destination host, without going through the expensive key exchange process as in IPSec. Our mechanism is compatible with the current IP protocol and we tested our scheme with live HTTP and ICMP traffic. Our results show that our protocol provides a zero-configuration network layer encryption solution for end-to-end secure communications that is ideal for consumer electronics applications.*

Index Terms — Network encryptor, Identity-based encryption (IBE), Tate pairing, Supersingular curve.

I. INTRODUCTION

Network encryptor can be used to provide end-to-end encryption of data sent from the source node to the destination node. Compared to application layer encryption, such as S/MIME and PGP, which only works for a specific application and requires installation of programs in the client workstations and server hosts, network encryptor is independent of applications and is completely transparent to the end users.

Internet Protocol Security (IPSec), which operates at the network layer, is a well-established protocol that provides end-to-end encryption for securing IP communications. In IPSec, both the sender and receiver must use the Internet key exchange protocol (IKE or IKEv2) to set up a security association (SA) for negotiating encryption protocols and algorithms, and for generating the encryption and authentication keys before an IPSec-protected communication can be set up. These generated encryption and authentication keys are then reused throughout the lifetime of the association.

Unfortunately, because of its complexity, IPSec is commonly implemented in software requiring operating

Sammy H.M.Kwok (e-mail: samkwo@graduate.hku.hk), K.H. So (e-mail: hso@eee.hku.hk), Edmund Y. Lam (e-mail: elam@eee.hku.hk) and K.S. Lui (e-mail: kslui@eee.hku.hk) are with the Department of Electrical and Electronic Engineering, the University of Hong Kong, Pokfulam Road, Hong Kong

system support. Not only will such complex software setup prevent such security system to scale to future high-speed networks, the large per-host setup task also presents a significant challenge to system administrators of large corporations with thousands of clients.

In this paper, we present a layer-3 end-to-end network encryptor using identity based encryption (IBE). In our scheme, the IP address of a target host is used as its identity for encryption. The proposed network encryptor has the following advantages:

- Because of the simplicity of our IBE scheme, the encryptor can be implemented entirely in hardware as a black box, eliminating any need of per-host software setup. Furthermore, such hardware solution has the potential to scale with future high-speed networks.

- Because IBE is used, complicated key management and exchange processes as in IPSec are unnecessary. As a result, the overhead of our encryption scheme is small when compared to IPSec.

- Since complex key negotiation is not needed in our scheme, each IP packet can be encrypted using a different key, eliminating the risk of key-hijacking.

- Compared to traditional public-key technologies such as RSA public key system (PKCS#1) and Elliptic Curve Diffie-Hellman public key system (ECDH), our scheme eliminates the burden of complex key management from the key servers. In fact, the presence of the key generating server (KGS) in our scheme is not required for data communications.

As the proposed encryption scheme requires no per-host software setup, we term this scheme a *zero-configuration network encryption scheme*. Because of its very simple key management mechanism, our scheme is particularly suitable for use in large corporations in which secure IP communication must be provided for use among a large number of devices in the corporation's intranet and for roaming Internet access.

The uses of identity-based cryptosystems have been proposed under various different contexts. In [1], an identity-based encryption scheme was proposed to provide secure and anonymous roaming wireless access, which improves on schemes such as [2] that utilizes symmetric key for encryption. Similarly, an identity-based key exchange scheme was used in [3] to address man-in-the-middle attack of their previously proposed FEA-M scheme [4]. However, we are not aware of any prior work utilizing such identity-based scheme for IP network traffic encryption.

The rest of the paper is organized as follows. Background information about IBE using Tate pairing is first presented in

Section 2, followed by the description of the proposed encryption scheme in Section 3. Section 4 describes our initial implementation results and a method to enhance throughput. We discuss security considerations and the scheme's ability to coexist with common network protocol in Section 5 and Section 6, respectively. Concluding remarks are then given in Section 7.

II. BACKGROUND

The concept of IBE was introduced by Shamir in 1984 [5]. The idea is to let a user's identity to be used as his public key. The corresponding private key of a user can be generated by a publicly trusted Key Generating Server (KGS) using the user's public key and the master secret key of the KGS. In 2001, Boneh and Franklin [6] invented the first feasible solutions for IBE using the Weil pairing on elliptic curves. Since then, many ID-based key agreement protocols and signature schemes using bilinear pairing have been suggested [7].

We propose to use the Boneh-Franklin IBE scheme [6] to encrypt IP packets. In our scheme, IP address is used as identity of network hosts and so any message sender can calculate the public key of the message receiver using his IP address. Tate pairing [8] on an elliptic curve, E is used to generate the shared secret between the message sender and receiver. Such shared secret is used as the per-packet key for encrypting the IP packet. Tate pairing was chosen in our scheme because of its relatively low computational cost.

The ease of implementation and degree of security depends on the choice of this curve, E. We will defer the discussion of our choice of E in our current implementation to Section 3.A.

We now briefly describe the mechanism of Tate pairing. Let E over F_q be an elliptic curve where q is a power of a prime and let $P \in E(F_q)[l]$ and $Q \in E(F_{q^k})[l]$ be a pair of points on E where k is the embedding degree of E. The Tate pairing operates on P and Q, and produces a result in $F_{q^k}^*$:

$$\tau : E(F_q)[l] \times E(F_{q^k})[l] \rightarrow F_{q^k}^* / (F_{q^k}^*)^l \qquad (1)$$

We write $\tau(P,Q)$ for the Tate pairing of P and Q. For P of order n, to get $\tau(P,Q)$, we first find a rational function f_P so that $div(f_P)$ is equivalent to $n(P) - n(0)$ and then evaluate f_P at a divisor, D_Q equivalent to $(Q) - (0)$. That is,

$$\tau(P,Q) = f_P(D_Q)^{(q^k-1)/l} \qquad (2)$$

An effective algorithm for finding f_P was proposed by Miller and was further improved by the works of [9] and [10]. In [11] and [12], a fast formula for the Tate paring computation of supersingular elliptic curve, E_b over binary field, with $\gcd(m,2) = 1$ was proposed. That is,

$$E_b : y^2 + y = x^3 + x + b \text{ where } b = 0,1 \qquad (3)$$

The corresponding elliptic curves in the form of Equation 3 have the embedding degree, $k = 4$ and have orders dividing $22m + 1$ [13]. The most useful property of Tate pairing for use in IBE is its bilinearity [11]. That is,

$$\tau(aP,bQ) = \tau(P,Q)^{ab} \qquad (4)$$

III. PROPOSED ENCRYPTION SYSTEM

There are two distinct phases in our encryption scheme. First, a user must register with a central key generating server (KGS) to obtain its private key. The KGS holds the master key of the system which is required for generating the private key of a user. Once equipped with his private key, a user may then engage in encrypted communication with another user without contacting the KGS again. In this sense, the workload of our KGS is much lower than a certificate authority (CA) of a conventional public key infrastructure.

Then, when a user A wants to transmit a stream of packets to user B, A would encrypt each individual packet with a unique key generated based on the IP address of B. Forty (40) bytes of information that is needed to decrypt a packet is embedded in the packet. To ensure that the encrypted packet may travel through standard routers, only packet payload and its corresponding header fields are modified in-place. Upon receiving an encrypted packet, B decrypts the packet using the information carried in the packet and his own private key.

A. System Setup

As mentioned in Section 2, the choice of E affects not only the efficiency of all subsequent computations, but also the security level of our scheme. To facilitate decryption of a packet on the receiving end, each IP packet needs to be embedded with extra information. The size of this information is directly proportional to the degree of E. At the same time, the security level of our encryption scheme increases with the degree of E. Therefore, a trade-off must be made to balance the effects.

In our current implementation, we have chosen this underlying elliptic curve as a supersingular curve E over $F(2^m)$ where $m = 283$. E is in the form of Equation 3 with $b = 0$. That is,

$$E : y^2 + y = x^3 + x \qquad (5)$$

and the generating polynomial for the finite field is

$$f(x) = x^{283} + x^{12} + x^7 + x^5 + 1 \qquad (6)$$

As the embedding degree of E is 4, the result of Tate pairing is in $F(2^{4m})$.

This choice of E is made because its corresponding Menezes, Okamoto, and Vanstone (MOV) security level [15] is 1132 which is comparable to a 1024-bit RSA encryption scheme.

B. Master Key and User Registration

For each instance of our IBE system, the KGS must first generate a set of system-wide parameters according to the following steps:

Step i   : Select a point P on E.

Step ii   : Generate a field $x \in F(2^m)$ randomly.

Step iii : Calculate xP.

While x is kept by the KGS as the master key, P and xP are announced to all users as the public parameters for the system.

When a user registers with the KGS, the latter generates the user's private key based on his IP address using the following procedures:

Step i   : Calculate the public key, A for the user, e.g. Alice, with IP address, ip_A

$$A \leftarrow GenPubKey(ip_A)$$

where the function GenPubKey() is defined in Algorithm 1.

Step ii   : Calculate the private key of Alice, xA.

Step iii : Send xA to Alice through a secure channel.

Algorithm 1: Generate Public Key from IP

  function GenPubKey(ip)

    Let $P(P_x, P_y)$ be a point on E

    $P_x \rightarrow MAP(ip)$

    repeat

      Use Equation 3 to find P_y

      if P_y does not exist **then**

        $P_x = P_x + 1$

      end if

    until P_y exists

    return P

  end function

  function $MAP(ip)$

    for i = 0 to m -1 **do**

      $P_x[i] = ip[i \bmod 32]$

    end for

    return P_x

  end function

C. Encrypting IP Packets

When Alice sends a stream of IP packet to Bob, each packet is encrypted using the following steps:

Step i   : Calculate the public key of Bob, B using his IP address, ip_B :

$$B \leftarrow GenPubKey(ip_B)$$

Step ii   : Pick r randomly from $F(2^m)$

Step iii : Calculate rP.

Step iv   : Calculate the shared secret, $s \in F(2^{4m})$:

$$s = \tau(B, xP)^r \tag{7}$$

Step v   : Convert s to a key of length of m bits and use it as the per-packet key, $K_s \in F(2^m)$. That is:

$$K_s = s_1 + s_2 + s_3 + s_4 , \tag{8}$$

where $s = \{s_1, s_2, s_3, s_4\}$ and

$$s_1, s_2, s_3, s_4 \in F(2^m)$$

Step vi   : Encrypt the IP packet by a RC4 stream cipher using K_s as the encryption key.

Step vii : Send the x-coordinate of rP, $X(rP)$, and the encrypted IP packet to Bob.

D. Decrypting IP Packets

When Bob receives an encrypted IP packet, he takes the following steps to decrypt the packet:

Step i   : Extract $X(rP)$ from the received packet.

Step ii   : Calculate the y-coordinate of rP based on $X(rP)$,

Step iii : Calculate s using his private key, xB :

$$s \leftarrow \tau(xB, rP) \tag{9}$$

Step v   : Convert s to a key of length of m bits and use it as the per-packet key, K_s.

Step vi   : Decrypt the received IP packet by a RC4 stream cipher using K_s as the decryption key.

Note that according to Equation 4,

$$\tau(xB, rP) = \tau(B, P)^{xr} = \tau(B, xP)^r \tag{10}$$

As a result, s calculated in Equation 9 is identical to the s calculated in Equation 7 that was used to encrypt the packet, making decryption possible.

E. Encrypted IP Datagram Format

The network encryptor works at the network layer to encrypt payload data in IP datagram. Because a RC4 stream cipher is used, the payload of the original packet is modified in-place with encrypted data. However, according to the protocol in Section 3.C step vii, $X(rP)$ must be appended into the payload of the encrypted IP packet. The size of $X(rP)$ is $\lceil m/8 \rceil$ = 36 bytes.

Since extra information must be inserted into the original IP packets, the resulting IP packets may exceed the maximum transmission unit (MTU) of 1500 bytes [16]. For such cases, the original packet is fragmented into two encrypted packets, with the first one containing the encrypted payload and the second one containing $X(rP)$. We call the first fragment F_1 and the second fragment F_2. To determine if a packet is fragmented, an encrypted packet type identifier (EPT) is appended in the encrypted packets.

In the case of unfragmented packet, together with the EPT, a total of 40 bytes is appended to the original IP packet. As a result, both the checksum and the length fields of the header must be modified accordingly. The resulting datagram is depicted in Figure 1.

Since F_1 needs to carry the 4-byte EPT, 4-byte of encrypted payload is now carried by F_2. Therefore, F_2 contains the remaining 4 bytes of encrypted payload together with $X(rP)$ and EPT.

Ver	header len	type of service	length' = length + 40	
16-bit identifier			flags	fragment offset
time to live		upper layer	internet checksum'	
Source IP address				
Destination IP address				
Options				
Encrypted payload data (in place)				
X(rP)				
EPT = NORMAL				

Fig. 1. Encrypted IP datagram with payload replaced.

IV. IMPLEMENTATION RESULTS

To demonstrate the feasibility of the above methodology, we have implemented the proposed network encryptor in software.

A. Implementation Details

The network encryption scheme was implemented under Microsoft Windows environment using Microsoft Visual C++ 6.0 development tool. An open source library, WinPCap V4.0.2 was used for receiving and sending IP packets. Another open source library called MIRACL was used for implementing all the cryptographic algorithms.

Figure 2 shows the high-level block-diagram of our test setup. Two computers, Host A and Host B, which act as source and destination for generating traffic, are set up within a local area network (LAN) that is connected to the Internet through a gateway router. All traffic between Host A and Host B are encrypted and decrypted by the two encryptors currently implemented by two 1.6GHz Pentium IV computers. The remaining traffic going on to the Internet are not encrypted.
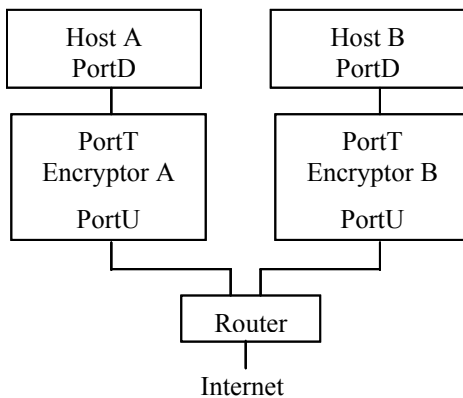

Fig. 2. Hardware test platform.

Between the two hosts, ICMP packets were first used for testing the basic functionality of our encryption scheme.

Subsequently, standard web browsing traffic for downloading a file of 9M bytes was generated between the two hosts. Together with the encrypted traffic, background traffic going to the Internet, including FTP, POP, SMTP and HTTP traffic, are generated. The encrypted and the unencrypted traffic have been demonstrated to coexist successfully using our initial implementation.

Using our initial software implementation, the processing time for encrypting an IP datagram is 70ms while the processing time for decrypting an IP datagram is 64ms.

B. Network Bandwidth Overhead

Referring to Section 3.E, the overhead for the encryption is 40 bytes if the size of the original IP datagram is 1460 bytes or less. If the packet is fragmented because of the size is larger than 1460, the overhead will be equal to the size of F_2. The size of F_2 is the sum of the size of IP header, 4 bytes of payload from F_1, $X(rP)$ and EPT, i.e., 20+4+36+4 = 64 bytes.

The overhead of our scheme can be calculated using the following formula:

$$overhead = \sum_{i=28}^{1500} o(i)P(i) \qquad (11)$$

where $o(i)$ is the overhead of our scheme when the packet size is i, and $P(i)$ is the probability that a packet has a size of i. Assuming the size of IP packets is uniformly distributed between the minimum size of 28 bytes and the maximum size of 1500 bytes, then the network bandwidth overhead is:

$$\sum_{i=28}^{1460} \frac{40}{i} P(i) + \sum_{i=1461}^{1500} \frac{64}{i} P(i)$$
$$= \frac{40}{1500-28+1} \sum_{i=28}^{1460} \frac{1}{i} + \frac{64}{1500-28+1} \sum_{i=1461}^{1500} \frac{1}{i}$$
$$= 10.87\%$$

Such overhead does not impose significant traffic loading to the network.

C. Speed Improvement

The most time-consuming step in the proposed IBE scheme is the calculation of Tate pairing. To improve the throughput of the encryptor, we propose a key caching scheme to eliminate the need of Tate pairing calculation for every IP packet.

The proposed key caching scheme works as follows. After Alice sends the first encrypted IP datagram to Bob, she caches rP and s. When she sends the next IP datagram to Bob, she calculates rP and s using Equation 12 and Equation 13 respectively instead of following step ii to iv in Section 3.C.

$$rP \leftarrow rP + rP \qquad (12)$$
$$s \leftarrow s^2 \qquad (13)$$

Similarly, Bob also caches rP and s after decrypting the first IP datagram received from Alice. The cached values are subsequently used to calculate the new rP and s for

decrypting the next IP datagram from Alice. After sending N (a parameter predetermined by Alice) IP datagrams to Bob, Alice resets her key cache and follows step ii to iv in Section 3.C again to calculate rP and s.

Without any Tate pairing computation, the processing time for encrypting/decrypting an IP datagram is reduced to 520μs, which is about 130 times faster compared to the one with Tate pairing computation.

Subsequently, the frequency of resetting the cache will also affect the overall throughput. For example, if N = 1000, the average processing time for encrypting IP datagram is reduced to (70+0.52×999)/1000=0.59ms. Assuming the average size of IP datagram is 744 bytes and ignoring the processing time for receiving and sending IP packets, the throughput of the encryption for one-way traffic is 744×8/0.59ms ≈ 10Mbps.

Note that this proposed key caching scheme has intrinsic capability for solving the synchronization problem. That is, if Bob fails to receive any of the encrypted IP datagrams from Alice, the cached s cannot be used to calculate the new s for the decryption of the next encrypted IP datagram from Alice. However, Bob can still use the received $X(rP)$ and follow all steps in Section 3.D to decrypt the next IP datagram and resynchronize the key exchange between Alice and him.

V. SECURITY CONSIDERATIONS

The proposed network encryptor can provide end-to-end encryption of messages over IP network so they cannot be sniffed and uncovered by an unauthorized entity. However, as the encryptor works in transport mode, in which only the payload of the message but not the header information is protected, attackers may find some useful information from the unprotected header information to attack the network equipped with the network encryptors. On the other hand, at the present stage, the network encryptor only provides confidentiality but not source authentication and message integrity. Thus, it can only protect against passive attacks (eavesdropping and sniffing data as it passes over the network) but not active attacks (altering data and masquerading as another individual to send data over the network). We will explore how to enhance our mechanism to support authentication and message integrity in the future.

A. Chosen-Plaintext Attack

As any user in the system can use the encryptor to encrypt any chosen plaintext, the system is subject to chosen-plaintext attack in which an attacker can choose the plaintext that gets encrypted and obtain the corresponding ciphertext from the output of the encryptor. However, as a 283-bit RC4 cipher is used for the encryption, it is unlikely to discover the key for the encryption by simply analyzing the plaintext-ciphertext pairs. In addition, each IP datagram is encrypted by a unique per-packet key. Therefore, the security of the system will not be seriously affected even if one of the keys is discovered by the attacker.

B. Key Escrow

As the KGS is in possession of the master secret, x, it encompasses the full knowledge of private keys of all users, allowing it to decrypt any message sent to any user. There are two ways to reduce the risk of breaking the entire IBE system owing to the compromise of the KGS: 1. by using distributed key generating servers; and 2. by using short-lived master key.

In the first method, x is split into two or more parts, x_i, where $\sum_i x_i = x$. Each x_i is then kept independently by a different key generating server, KGS$_i$. When a user, such as Alice with public key A, registers with the system, she must approach each KGS independently. Each KGS$_i$ will then return a partial private key $x_i A$, as well as $x_i P$ to her after verifying her identity. Once equipped with such information, Alice may then calculate her true private key $xA = \sum_i x_i A$ as well as $xP = \sum_i x_i P$. Since each KGS$_i$ possesses only x_i, no individual KGS can calculate the private key of any user unless they conspire to do so, which also reduces the risk of compromising x if any one KGS is compromised.

The second method to lower the chance of compromising the master secret key, x is by employing a short-lived master key x. In this case, KGS changes the value of x at a regular interval. With each new master key, private keys for all users are also updated. The updated private keys may subsequently be pushed to each individual users registered in the system, or pulled by a user when the old private key is found incapable of decrypting IP datagram correctly.

VI. APPLICATION CONSIDERATIONS

This section discusses the implications of application protocols commonly deployed to our IBE encryption scheme and the precautions that must be made to ensure correct operations.

A. Dynamic Host Configuration

Dynamic Host Configuration (DHCP) is a protocol used for assigning IP addresses to individual networked devices dynamically for operation in an IP network. As a fundamental assumption of our proposed IBE scheme is to utilize the IP address of the receiver as the unique identity for public key calculation, our scheme cannot work with an IP network using DHCP that randomly assigns IP addresses to users. However, as many DHCP servers do, it is possible to fix the mapping between a MAC address and the corresponding assigned IP address. With such a mapping, our scheme may then perform as expected.

B. Network Address Translation

When an IP packet is routed through a router, the source or destination IP address in the IP header may be translated by the router, which will affect the use of our proposed network encryptor. For our scheme to be compatible with NAT, extra encryptors and decryptors must be deployed together with the NAT server, as shown in Figure 3.
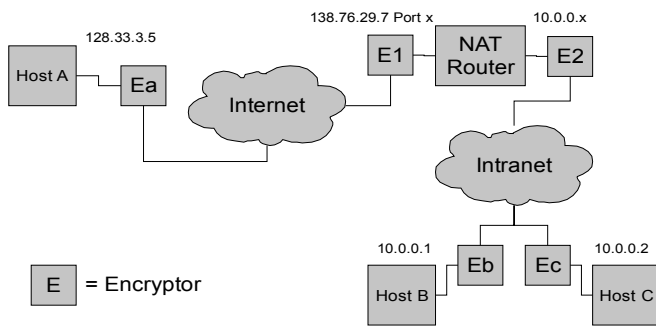
Fig. 3. Use of the network encryptors with a NAT router.

In the proposed network architecture, E1 keeps the private key for the IP address of 138.76.29.7 and it is only used for decrypting IP packets from the Internet, whereas E2 does not keep any private key and it is only used for encrypting IP packets from the intranet. For all outgoing IP packets from Host B or C to Host A, as there is no address translation, E1 and E2 will simply route the packets through them. For all incoming IP packets sent from Host A to Host B or C, the packets will be decrypted by E1 first and then encrypted again by E2 using the IP address of Host B or C as the public key before sending the packets to Host B or C.

C. Mobile IP

Mobile IP is an extension to the Internet Protocol which enables mobile computers to stay connected to the Internet regardless of their locations and without changing their IP addresses. Mobile IP uses two types of routing protocols, namely, indirect routing and direct routing.

In indirect routing protocol, IP packets sent to the mobile user use its home address. Since the destination address of an IP datagram remains pointing to the original home IP address, our proposed network encryption scheme, which requires only the destination address for encryption, works without any modification in this case.

In direct routing protocol, the following techniques can be used: During the link setup phase in which the correspondent user gets the care-of-address of the mobile user from the home agent, the encryptor connected to the correspondent user sniffs the packets sent between the correspondent user and the home agent, obtains the care-of-address of the mobile user, and associates it with the home address of the mobile user. When the corresponding user actually sends data to the mobile user using his care-of-address, the encryptor uses the care-of-address to look up the home address of the mobile user and uses the home address as the public key for encryption.

VII. CONCLUSION

In this paper, a zero-configuration Identity-based IP network encryptor is presented. The destination IP address of an IP datagram is used to generate unique keys to encrypt each individual IP datagram. Since all information required to generate encryption keys is available with the sender, no communication with centralized key servers is needed, eliminating any possible performance bottleneck. Furthermore, since a new per-packet key is generated for each individual IP datagram, the risk of key hijacking is virtually eliminated.

In addition, design considerations when our IP-centric encryption scheme must coexist with common network protocols, including DHCP and NAT, are presented.

The proposed encryptor has been implemented in software and the initial implementation results indicate that it can work effectively to provide strong encryption for IP packets sent among hosts in an IP network. Furthermore, using a key caching scheme, our proposed network encryptor can achieve a throughput of 10Mbps half-duplex traffic.

In the future, we plan to implement the IBE encryption scheme entirely in FPGA so as to provide the needed speed to cope with live traffic in a 10Gbps Ethernet network.

REFERENCES

[1] Z. Wan, K. Ren, and B. Preneel "A secure privacy-preserving roaming protocol based on hierarchical identity-based encryption for mobile networks," in *Proceedings of the First ACM Conference on Wireless Network Security (WiSec '08)*, Mar. 31 – Apr. 02, 2008, ACM, New York, NY, pp. 62-67.

[2] J. Zhu; J. Ma, "A new authentication scheme with anonymity for wireless environments," *Consumer Electronics, IEEE Transactions on* , vol.50, no.1, pp. 231-235, Feb 2004.

[3] X. Yi, C. H. Tan, C.K. Siew and M. R. Syed, "ID-based key agreement for multimedia encryption," *Consumer Electronics, IEEE Transactions on* , vol.48, no.2, pp.298-303, May 2002.

[4] X. Yi, C. H. Tan, C.K. Siew and M. R. Syed, "Fast encryption for multimedia," *Consumer Electronics, IEEE Transactions on*, vol.47, no.1, pp.101-107, Feb 2001.

[5] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of CRYPTO'84 on Advances in cryptology*. New York, NY, USA: Springer-Verlag New York, Inc., 1985, pp. 47-53.

[6] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *CRYPTO'01*. London, UK: Springer-Verlag, 2001, pp. 213-229.

[7] L. Martin, *Introduction to Identity-Based Encryption*, 1st ed. Artech House Publishers, Jan 2008.

[8] G. Frey *et al.*, "The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems," *Information Theory, IEEE Transactions on*, vol. 45, no. 5, pp. 1717-1719, Jul 1999.

[9] P. Barreto *et al.*, "Efficient algorithms for pairing based cryptosystems," in *CRYPTO 2002, LNCS*, vol. 2442. Berlin, Germany: Springer-Verlag, 2002, pp. 354-369.

[10] S. Galbraith *et al.*, "Implementing the tate pairing," in *ANTS 2002, LNCS*, vol. 2369. Berlin, Germany: Springer-Verlag, 2002, pp. 69-86.

[11] P. Barreto, S. Galbraith, C. hEigeartaigh and M. Scott., "Efficient pairing computation on supersingular abelian varieties," *Des. Codes Cryptography*, vol. 42, no. 3, pp. 239-271, 2007.

[12] S. Kwon, "Efficient tate pairing computation for elliptic curves over binary fields," in *ACISP'05, LNCS*, vol. 3574. Berlin, Germany: Springer-Verlag, 2005, pp. 134-145.

[13] A. Menezes, *Elliptic Curve Public Key Cryptosystems*. Kluwer Academic Publishers, 1993.

[14] I. Blake *et al.*, *Advances in Elliptic Curve Cryptography*, 1st ed. Cambridge University Press, 2005.

[15] A. Menezes *et al.*, "Reducing elliptic curve logarithms to logarithms in a finite field," *Information Theory, IEEE Transactions on*, vol. 39, no. 5, pp. 1639-1646, Sep 1993.

[16] J. Mogul, *RFC1191*, Nov 1990.

BIOGRAPHIES

Sammy H. M. Kwok (M'96) began his engineering study at the Electrical and Electronic Engineering Department of the University of Hong Kong in 1985. After receiving the B.S. degree in 1989, he joined the Government of Hong Kong to work as an engineer for the research and development of electronic and computer systems. In 1995, he received the M.S. degree in Communications and Signal Processing from the Imperial College of Science, Technology and Medicine, University of London. He then returned to the Government of Hong Kong to work as an engineering manager to lead a team of engineers for engineering design and implementation projects. He is currently a part-time PhD candidate at the University of Hong Kong, where he is focusing on the effective and secure FPGA implementations of cryptographic primitives and algorithms.

Hayden K. H. So (S'03–M'07) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer sciences from University of California, Berkeley, CA in 1998, 2000, and 2007 respectively. He was at Xilinx Research Lab, San Jose, CA as an Embedded & Reconfigurable Systems Intern. and at Googol Technology (HK) Ltd., Hong Kong as a hardware engineer. He is currently a Research Assistant Professor of electrical and electronic engineering at the University of Hong Kong.
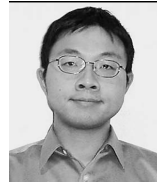
Edmund Y. Lam (S'97–M'00–SM'05) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA. He was at KLA-Tencor Corporation, San Jose, CA, as a Senior Engineer, and is now an Associate Professor of electrical and electronic engineering at the University of Hong Kong.
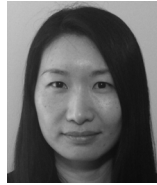
King-Shan Lui (S'00–M'03-SM'09) received the B.Eng. and M.Phil. degrees in Computer Science from the Hong Kong University of Science and Technology. After receiving her PhD degree from the University of Illinois at Urbana-Champaign, USA, she joined the Department of Electrical and Electronic Engineering of the University of Hong Kong.