Attacking the IPsec Standards When Applied to IPv6 in Confidentiality-only ESP Tunnel Mode

Dongxiang Fang, Peifeng Zeng, Weiqin Yang

College of Computer Science and Technology, Donghua University, Songjiang, Shanghai 201620, China

fdx321@126.com, zengpf@yahoo.com, july0810@163.com

Abstract— Attacks which can break RFC-compliant IPsec implementation built on IPv6 in confidentiality-only ESP tunnel mode are proposed. The attacks combine the thought of IV attack, oracle attack and spoof attack to decrypt a encrypted IPv6 datagram. The attacks here are more efficient than the attacks presented by Paterson and Degabriele because no checksum issue has to be handled. The paper shows that using IPsec with confidentiality-only ESP configuration is insecure to convince users to select it carefully.

***Keywords*— IPsec, IPv6, ESP, confidentiality-only, Security**

I. Introduction

IPsec is a security protocol suite to provide various security services for traffic at the IP layer, in both IPv4 and IPv6 environments. Two major security protocols of IPsec are Authentication Header (AH) and Encapsulating Security Payload (ESP) [9], [12]. AH provides connectionless integrity check and data origin authentication for IP datagrams [10], [13]. ESP provides confidentiality as well as integrity and authentication services [11], [14].

ESP can be applied alone, in combination with AH, or in a nested fashion. The IPsec standard [14] allows ESP to be configured in three modes: confidentiality-only, integrity-only, confidentiality and integrity. Encryption without integrity is vulnerable. There are plenty of high-profile attacks [1]–[5] that can attack the encryption without high integrity checks successfully. Yet the latest IPsec standard still allows confidentiality-only configuration. The RFCs [11], [14] just indicate that using confidentiality without integrity/authentication may subject the traffic to some attacks which could undermine the encryption. However, the warnings and the well-known attacks fail to prevent the developers from offering the confidentiality-only mode to users, nor stop the users from choosing this configuration.

Bellovin [1] was the first to point out the confidentiality failures and spoofing failures of IPsec. He proposed some attacks that can read encrypted data and transmit phony data in the IPsec channel. But the attacks' prerequisite that the attacker has a legitimate login on one or both of the machines performing IPsec is rather unrealistic. And the attacks can't be applied to the fully implemented systems which are faithful to the IPsec standards. In particularly， encryption padding check [11], [14] can make it fail. But Bellovin's attacks are well-known and have been cited in the subsequent versions of ESP standards to avoid these failures.

McCubbin [2] analysed several IV attacks which are based on modifying the unauthenticated initialization vector (IV) of a CBC-encrypted packet during transmission. The attacker can change the first block of the decrypted plaintext by modifying the IV then use it to spoof the packet receiver. Vaudenay [3] sketched the steps to operate a padding oracle attack against IPsec, but he didn't point out how to build a suitable padding oracle.

All these attacks introduced above are rather theoretical. They haven't been implemented in practice, and it must have many obstacles when applying them in practice. Paterson and Yau [4] aimed to convince users not to select confidentiality-only mode. They presented attacks against the CBC (Cipher-block chaining) encryption-only, tunnel mode configuration of IPsec, and showed how to build the attacks against Linux IPsec in practice. However, their attacks can't be applied to RFC-compliant implementations of IPsec which carries out the policy checks [9], [12]. And they require the attacker to be able to monitor all the traffic emanating from a host performing IPsec. So, Paterson and Degabriele [5] proposed some new attacks based on the ideas of IV attacks and padding oracle attacks. The attacks can break any RFC-compliant implementations of IPsec using confidentiality-only ESP and relaxed the prerequisite to just monitor the traffic in two directions in an IPsec tunnel instead of all the traffic emanating from a host.

But their attacks can be applied only in IPv4 network. In this paper, we proposed two kinds of attack that can break the IPsec when it is applied to IPv6 network with confidentiality-only ESP tunnel mode configuration. We borrow the main idea from [5]. The attacker can modify some bits of the IV to change one field of the inner IP header, so that the packet receiver produces an ICMP error message. And the detection of this message can be used as an oracle to perform the padding oracle attack. The attacks we proposed is more efficient than the attacks in [5], because there is no checksum in IPv6 header so that the attacker doesn't need to do extra job to fix the checksum as described in [5]. A further theme of this paper is to show that using IPsec with confidentiality-only ESP configuration is very risky, and try to convince users select it carefully.

The rest of the paper is organized as follows: the background on IPsec, IPv6 and ICMPv6 is described in section 2. In section 3, the new attacks against IPsec applied to IPv6 with confidentiality-only ESP tunnel configuration are analysed in detail. Then we conclude in section 4.

II. Background on IPsec, IPv6 and ICMPv6

IPsec is described in this section, we just focus on the padding and CBC mode which are most related to our attacks. The fields of IPv6 header used in the attacks are described in detail. What will cause ICMPv6 error messages and the format of these messages are also presented here.

A. IPsec

IPsec is a protocol suite for securing IP communications by authenticating and/or encrypting each IP packet of a communication session, as defined in RFCs 2401-2412 and 4301-4309. It uses the following protocols to provide various services:

- AH provides connectionless integrity check and data origin authentication for IP datagrams, and it can protect host against replay attacks.
- ESP can provide data origin authentication, connectionless integrity check, and anti-replay services as well as AH. The primary difference between the integrity provided by ESP and AH is the extent of coverage. ESP also provides a confidentiality service.

IPsec can be deployed in a transport mode or tunnel mode. In Transport mode, only the payload of the IP packet is encrypted and/or authenticated. In tunnel mode, it is operated on the entire IP packet.

In this paper, we focus on the tunnel mode ESP operating block cipher algorithms in CBC mode. In tunnel mode, the entire IP packet to be protected is called inner IP datagram as shown in Figure 1. The inner IP datagram is treated as the payload data as shown in Figure 2. It's padded with a particular pattern of bytes and then the PL (Pad Length) and NH (Next Header). The NH is presented for the packet receiver to decide which protocol the bytes that precede the Padding should be passed to. In tunnel mode, this value should be 4 indicating IPv4 or 41 indicating IPv6. The Padding, PL, and NH is called ESP trailer.

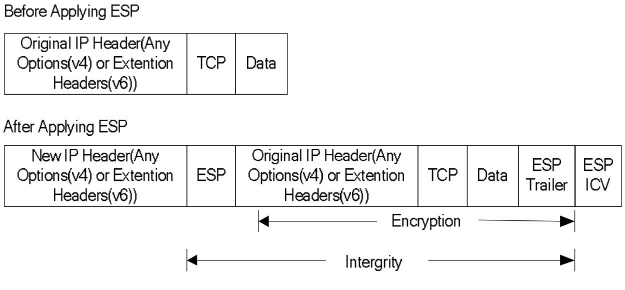

Before Applying ESP
Original IP Header(Any Options(v4) or Extention Headers(v6))
TCP
Data
After Applying ESP
New IP Header(Any Options(v4) or Extention Headers(v6))
ESP
Original IP Header(Any Options(v4) or Extention Headers(v6))
TCP
Data
ESP Trailer
ESP ICV
Encryption
Intergrity

Figure 1. Tunnel Mode ESP packet format according to RFC 4303

Our attacks delicately depend on how padding is operated, so we described it here in detail. Padding is used to extend the payload data to a size that fits the encryption's cipher block size and to align the next field. If padding bytes are needed but the encryption algorithm does not specify the padding contents. The bytes must be initialized with a null string or series of integer values (unsigned, 1 byte). The first byte appended to the plaintext is numbered 1, with subsequent padding bytes making up a monotonically increasing sequence: 1, 2, 3, … , n (0 < n < 256). For example, "1234554" is a valid ESP trailer, the second 5 indicates PL, and second 4 is the value of NH filed.

TFC (Traffic Flow Confidentiality) is permissible to precede Padding as shown in Figure 2, it is used to disguise the true length of the packet. For simplicity，we assume no TFC is used here. And the sender may add 0-255 bytes of padding, in the rest of the paper, we assume minimum bytes conforming to the rules above is used for padding (It is trivial to apply our attacks to the situations that using TFC or variable bytes of padding).

```
  0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |               Security Parameters Index (SPI)                 |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                      Sequence Number                          |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+---
 |                    IV (optional)                              | ^ p
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | a
 |                    Rest of Payload Data  (variable)           | | y
 ~                                                               ~ | l
 |                                                               | | o
 +               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ | a
 |               |         TFC Padding * (optional, variable)    | v d
 +-+-+-+-+-+-+-+-+         +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+---
 |                         |        Padding (0-255 bytes)        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                               |  Pad Length   | Next Header   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |        Integrity Check Value-ICV   (variable)                 |
 ~                                                               ~
 |                                                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2. ESP packet format according to RFC 4303

B. Padding and CBC mode

After padding, the bytes are encrypted in CBC mode. Let's assume that the bytes being divided into q blocks, each block has n bits. P denotes plaintext block, C denotes ciphertext block, and K means the key used for encryption/decryption. E denotes the encryption processing, and D denotes the corresponding decryption processing. ⊕ denotes the bit-wise exclusive-or. The first ciphertext block C_0 is a n-bits initialization vector selected at random. The plaintext is encrypted as below:

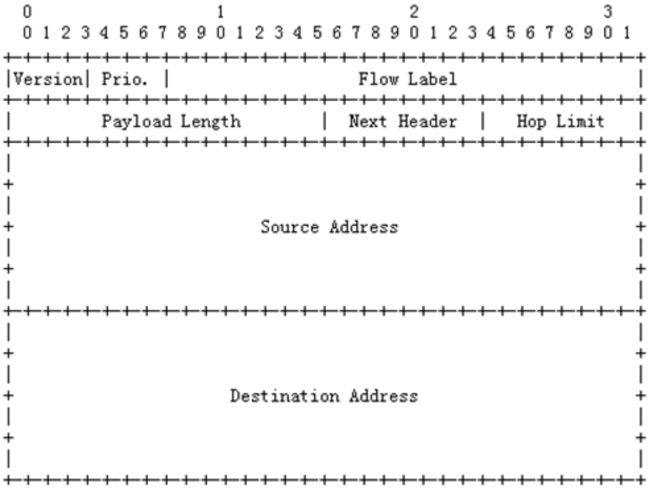
$$C_0 = IV, C_i = E_k(P_i \oplus C_{i-1}), (1 \le i \le q)$$

And here is decryption processing:

$$P_i = D_k(C_i) \oplus C_{i-1}, (1 \le i \le q)$$

C. IPv6 and ICMPv6

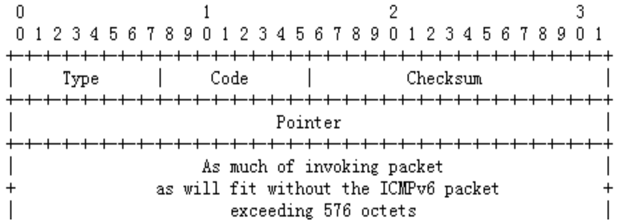
An IPv6 datagram consists of a fixed header and payload, and the payload consists of extension headers and data. The extension headers are optional. Here we focus on the fields of the fixed IPv6 header which will be used for the attacks. The header format is shown in Figure 3. For detail, see RFC 1883.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Prio. |                   Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                         Source Address                        +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      Destination Address                      +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3. IPv6 header format according to RFC 1883

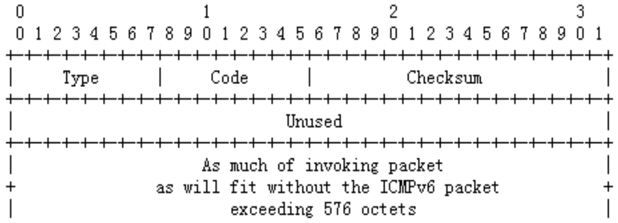
ICMPv6 is an integral part of IPv6 and must be fully implemented by every IPv6 node. It is used by IPv6 nodes to report errors, or perform other functions, like path MTU discovery, multicast group membership maintenance, neighbour discovery. Here we focus on the Time Exceeded and Parameter Problem ICMPv6 error messages.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            Pointer                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    As much of invoking packet                 |
+               as will fit without the ICMPv6 packet           +
|                       exceeding 576 octets                    |
```

Figure 4. ICMPv6 parameter problem message according to RFC 1885

The Next Header field (Figure 3) is 8 bits long, identifies the type of header immediately following the IPv6 header. If there is no extension header, it acts like the Protocol field in IPv4 header to indicate the upper-layer protocol (TCP, UDP, etc.). Or, it indicates the type of the next extension header (51 means AH, 52 means ESP, etc.). If the value doesn't mean any type, and the packet receiver can't recognize it to complete processing the packet, the packet must be discarded and ICMPv6 Parameter Problem message (Figure 4) should be sent to the source.



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |     Code      |          Checksum             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Unused                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    As much of invoking packet                 |
+               as will fit without the ICMPv6 packet           +
|                       exceeding 576 octets                    |
```

Figure 5. ICMPv6 time exceeded message according to RFC 1885

The Hop Limit field (Figure 3) is 8 bit long, avoids data packet circling in the network. The Hop Limit field is set value at the source host, decremented by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero. If a router receives a packet with a Hop Limit of zero, or a router decrements it to zero, it must discard the packet and send an ICMPv6 Time Exceeded message (Figure 5) to the source.

III. The attacks

In this section, the attacks are described in detail. For simplicity, the assumptions below are made, though it is easy to modify the attacks to handle other situations.

- Traffic Flow Confidentiality padding is not used.
- Minimum amount of padding is used.
- The implementation performs a strict padding check. RFCs [11], [14] suggest that the packet receiver should inspect the Padding field to against certain forms of "cut and paste" attacks. And the detection of ICMPv6 error message can be used as an oracle is based on the prerequisite that the message is generated when the padding, PL and NH of ESP trailer are correctly formatted.
- Confidentiality-only ESP is used in tunnel mode between a pair of gateways as shown in Figure 6. This is a common configuration of VPN based on IPsec.
- The attacker can monitor and capture ESP-protected packets between two gateways.
- The attacker can inject modified packets into the network between the two gateways.
- CBC mode is used for encryption, block size is 64bits, and the key used for encryption is fixed.

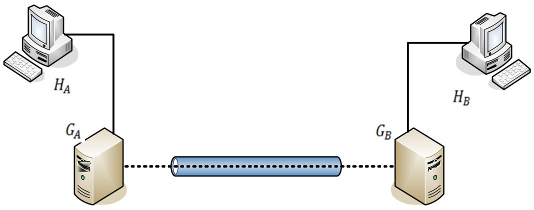


Figure 6. ESP Tunnel Network

A. *An Attack Based on Next Header Field*

The main idea is to take an existing encrypted packet captured from the network, transform it to a new ESP protected packet by modifying some bits of the IV block. The modification will cause the Next Header value of the inner IP datagram to be wrong, so that the new packet will always causes an ICMPv6 error message. The modified encrypted inner IP datagram can be used in the subsequent padding oracle attack.

1) ***First Step*:** We assume the attacker has captured an ESP protected packet as short as possible directed towards G_B from G_A. The encrypted inner IP datagram of the packet can be expressed as $C_0^{'}, C_1^{'}, \cdots C_r^{'} (r \geq 3)$ (r is bigger than 3 because in tunnel mode, the inner IP header should be transmitted). Modify $C_0^{'}$ by flipping bit 48 (begin with 0) to get a new block

C_0'', Bit 48 corresponds to the first bit of the Next Header field. Flipping it will cause the Next Header value to be unaccepted by the receiver, this will cause a Parameter Problem ICMPv6 error message to be sent from G_B to G_A. At this point, we have got $r+1$ chipertext blocks $C_0'', C_1', \cdots C_r'$ which can cause an ICMPv6 error message.

2) ***Second Step:*** Assume C_i is the target block we are going to attack, intercepted from the traffic flow from G_A to G_B. Use the blocks got in the first step to build a new ciphertext that has r+3 blocks, and inject it to the network.

$$C_0'', C_1', \cdots C_r', R^6, C_i$$

R^6 is a 64 bits random block. The last two blocks R^6 and C_i have no effect on the inner IP header, so the new ciphertext can still cause ICMPv6 error message unless it can't pass the IPsec padding check. G_B will discard the packet if the padding is invalid and NH is not 41 (IPv6). As described in [5], the most likely valid pattern is the one of length 0, which means PL equals 0 and NH equals 4. So we can varies R^6 at byte 6 and 7 from 0x00 to 0xFF, and monitor the network to look for an ICMPv6 error message from G_B to G_A. If ICMPv6 is not under IPsec policy, we can just see the payload of the message to get most plaintext of the original encrypted packet. Here we assume ICMPv6 is under the IPsec protection. When the message is detected, it is known to the attacker that the last two bytes of $R^6 \oplus D_k(C_i)$ are 0 and 4. C_{i-1} is already known, so it's trivial to extract byte 6 and 7 (begin with 0) of P_i by $C_{i-1} \oplus D_k(C_i)$. The maximum time of varying R^6 is 2^{16}, average time is 2^{15}.

The challenge here is the detection of ICMPv6 error message. Figure 4 and 5 show the structure of the messages. As much of invoking packet as will fit without the ICMPv6 packet exceeding 576 octets. This means the message includes the header and payload of the original packet without the ICMPv6 packet exceeding 576 octets. So the total length of the message can be used for the detection. For example, if the length of the packet which causes the Parameter Problem message is 80 (include IPv6 header and the payload), the message's length will be 88. If the length of the original packet is bigger than 568, the message's length will always be 576.

3) ***Third Step:*** Use the similar way to extract byte 5. Build a new ciphertext.

$$C_0'', C_1', \cdots C_r', R^5, C_i$$

R^5 is the same with R^6 except the 6th byte of it, $R_6^5 = R_6^6 \oplus 1$. This is to ensure that the last 2 bytes of the plaintext corresponding to the new ciphertext is 1, 4. Vary byte 5 of R^5 for 0x00 to 0xFF, inject it to the network. When detecting the ICMPv6 error message, it's known that the byte 5 of $R^5 \oplus D_k(C_i)$ is 1. Because only when the plaintext corresponding to the new ciphertext ending with 1, 1, 4, ICMPv6 message appear. Then extract the byte 5 of P_i.

In this way, all the bytes of P_i can be extracted, with maximum 2^8 trials, average 2^7 trials per byte.

B. *An Attack Based on Hop Limit Field*

This attack is similar to the attack based on Next Header Field, just different in the first step. Here, vary a byte (from bit 56 to 63) of C_0' from 0x00 to 0xFF to get a new block C_0'', the byte corresponds to the Hop Limit Field. Then inject $C_0'', C_1', \cdots C_r'$ to the network. When the Hop Limit value is 0, G_B will discard the packet and send an ICMPv6 Time Exceeded message. At this point, $C_0'', C_1', \cdots C_r'$ are the blocks we need. The next steps are the same as the attack based on Next Header attack.

IV. Conclusion

Two kinds of attacks based on the idea of Paterson and Degabriele are proposed in this paper. We extend the idea of them to IPv6 environment. The attacks can break RFC-compliant IPsec implementations which carry out strict padding check when the IPsec is applied to IPv6 with confidentiality-only ESP tunnel mode. The attacks are more efficient than the attacks based on the fields of IPv4 header, because the attacker doesn't need to deal with the checksum issue as there is no checksum in IPv6 header. The attacks highlight the point that using IPsec in confidentialiy-only configuration is vulnerable, and other factors should also be considered for the security, such as the ICMP blocking. Finally, it is more secure to be careful about the factors referred in this paper when using IPsec.

References

[1] S. Bellovin, "Problem Areas for the IP Security Protocols", in Proceedings of the sixth Usenix Unix Security Symposium, pp. 1-16, San Jose, CA, Jul. 1996.

[2] C. B. McCubbin, A. A. Selcuk and D. Sidhu, "Initialization Vector Attacks on the IPsec Protocol Suite." in 9th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE 2000), IEEE Computer Society, pp. 171-175, 2000.

[3] S. Vaudenay, "Security Faws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS...", in L. R. Knudsen (ed.), Advances in Cryptology - EUROCRYPT 2002, LNCS Vol. 2332, Springer-Verlag2002, pp. 534-545.

[4] K. G. Paterson and A. K. L. Ysau, "Cryptography in Theory and Practice: The Case of Encryption in IPsec." in S. Vaudenay (ed.), Advances in Cryptology - EUROCRYPT2006, LNCS Vol. 4004, SpringerVerlag, 2006, pp. 12-29. Full version available at http://eprint.iacr.org/2005/416.

[5] J. P. Degabriele and K. G. Paterson, "Attacking theIPsec Standards in Encryption-only Configurations." in IEEE Symposium on Privacy and Security, IEEEComputer Society, pp. 335-349, 2007.

[6] J. Postel, "Internet Control Message Protocol", RFC 792, Sept. 1981.

[7] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, Dec. 1995.

[8] A. Conta and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)", RFC 1885, Dec. 1995.

[9] S. Kent and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, Nov. 1998.

[10] S. Kent and R. Atkinson. "IP Authentication Header", RFC 2402, Nov. 1998.

[11] S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, Nov. 1998.

[12] S. Kent and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301 (obsoletes RFC2401), Dec. 2005.

[13] S. Kent, "IP Authentication Header", RFC 4302(obsoletes RFC 2402), Dec. 2005.

[14] S. Kent, "IP Encapsulating Security Payload (ESP)", RFC 4303 (obsoletes RFC 2406), Dec. 2005.

Dongxiang Fang received a B.E. degree in software engineering from Donghua University in 2012. He is currently studying in Donghua University for M.S. degree in computer science and technology. His research interests are in areas of network protocols, image processing and pattern recognition.

Peifeng Zeng received the B.E. and M.E. degrees from Southeast University in 1984 and 1990, respectively. He received the PhD degree from Nagoya University in 2002. He is currently a professor in College of Computer Science and Technology, Donghua University. Hi research areas are embedded systems, image processing and pattern recognition. Mr. Zeng is a member of the IEEE and the ACM.

Weiqin Yang received her B.E degree in Network Engineering from DongHua University, China, in 2013. She has been studied in DongHua University for her M.E degree in Computer Science and Technology since 2013. Her research interests are in areas of image recognition and tracking.