ID: A Mandatory Field in IKE?

ZHAOHUI CHENG[1], MANOS NISTAZAKIS[2] and RICHARD COMLEY[3]
School of Computing Science
Middlesex University
White Hart Lane, London N17 8HR
United Kingdom

Abstract: - The Internet Key Exchange (IKE) protocol is widely used as the kernel part of IPsec to build the virtual private networks. In the standardized IKE (RFC 2049), an identity (ID) field is included in the protocol exchanges in order to identify an involved party. However both in literature and practice, misuses of the ID field have occurred. In this paper, we address and clarify this problem, i.e. how to use an ID to identify a party in IKE in practice. We then go on to conclude that in most cases this mandatory field is not necessary in the main mode exchange and having the field as an option, IKE can achieve extra security properties.

Key-Words: - IPsec, IKE, Identification

1 Introduction

IPSec is a layer-three security protocol suit to provide network security for the IP-based communication. In the protocol suit, Authentication Header (AH) [9] and Encapsulation Security Payload (ESP) [10] are used along with the exchanged secrets between the communicating parties to provide security services. IKE [7] is used to establish the agreed secret dynamically.

An IKE session consists of two phases. The phase-one exchange assumes that each of the two parties engaged in the session has an identity by which the other side recognizes it, and associated with that identity is some type of secret that can be verified by the other side. The secret might be a pre-shared secret key or the private portion of a public/private key pair. During phase one, a mutual authentication based on that secret is performed and a Diffie-Hellman (D-H) key agreement is used to establish a session key that will be used to protect the remainder of the session. Phase one can be completed in two modes, i.e. the main mode and the aggressive mode. The aggressive mode uses fewer messages than the main mode but with less security. With the variants of the secrets, the protocol in phase one can use pre-shared-secret-keyed pseudorandom function, public key encryption or signature to complete the mutual authentication. After phase one, an Internet Security Association and Key Management Protocol (ISAKMP) [15] security association (SA) will be established.

Phase two (or the so called "quick mode") is used to establish one or more IPSec SAs. This phase includes the distribution of secrets and the negotiation of transforms used by AH or ESP to protect the subsequent communications between the two involved parties. The messages exchanged in phase two are protected by the ISAKMP SA that was established during phase one. Note that multiple phase-two exchanges can be launched under the protection of the same ISAKMP SA.

In the IKE protocol, four identities have been used either explicitly or implicitly (in fact, one more identity is used in the quick mode), i.e.

1. *ID1 is the IP address that is used to transmit the messages.*
2. *ID2 is ID_X that is declared by a party X in the messages to the intended party.*
3. *ID3 is the identification of a party's key (or one of the party's keys) that is bound with a public key of the party (possibly through a certificate) or a pre-shared secret key.*
4. *ID4 is the generic ID of a party.*

In a normal protocol, ID2, ID3 and ID4 are the same. For example, in a client and server system using a password-based protocol, each user has a unique user ID (ID4), and each user ID is bound with a secret password. Hence ID3 is equal to ID4. When a client tells the server its user ID via a message to log in, the ID field (ID2) in the message denotes a unique user. Hence ID2 acts as ID4. In the protocol exchanges, the client shows the server the witness of its possession of the password owned by user ID2. Because ID3 is equal to ID4, the above process can be interpreted in another way, i.e. ID2 in the message works as ID3. When the server is convinced that the client has possession of the

[1] Associated with OLYM-TECH Inc. as an external researcher.

password identified by ID2, it is assured that the client is just the user who owns the password.

However, in IKE the situation is complicated. A party in IKE is no longer a simple user or a security gateway. In fact a party is a legitimate delegation of a traffic flow identified by traffic flow selectors [8] (Section 3 discusses the implication of a party in IKE). A party can have more than one key, so it has more than one ID3. Hence ID3 is no longer equal to ID4. Furthermore normally there is no direct relation between a party's ID and the IDs of keys owned by the party in many cases. Although in IKE, ID2 is assumed to act as ID4 to identify a party [16] (in fact, in some cases ID2 is equivalent to ID4, while in other cases, ID2 is just ID3), the only assurance that the peer party has possession of a secret identified by ID2 (ID2 acts as ID3) can no longer guarantee the security, because an internal attacker can launch the man-in-the-middle attack to impersonate other parties. For example, in a system, there are three security gateways SG_V, SG_{AC} and SG_{BD}. SG_{AC} manages a traffic flow from A to C which is associated with a certificate whose identity is ID_{AC}, while SG_{BD} manages a traffic flow from B to D which is associated with a certificate whose identity is ID_{BD}. If SG_V regards ID2 as ID3 and only checks if SG_{BD} has possession of the private key corresponding to ID_{BD}, SG_{BD} can impersonate SG_{AC} to build a security channel for the traffic flow from A to C. Hence, it is crucial to check the authenticity of the declared ID and at same time differentiate the roles of IDs. However, this problem is more complicated in IKE because of the use of two phases and two IDs. In the literature, the problem seems to be ignored when people are designing or analyzing the protocol and in practice mistakes are made in some implementations. In this paper, we analyze how to use the ID field in the main mode and the quick mode and also discuss the implications of this problem to the protocol design.

The paper is organized as follows. For the ease of understanding the problem, some security flaws in using ID2 in practice are presented in the next section. We discuss the proper use of the ID field in practice in section 3 and the implications of the ID problem in section 4. A conclusion is drawn in the final part.

2 Mistakes In Practice

FreeSwan is a major open source project, which implements IPsec. A recognized feature of FreeSwan is the Opportunistic Encryption (OE) [25][20] whose aim is to allow encryption without any specific pre-arrangement on the pair of systems involved. The basic idea of the OE is to use the DNSSEC [4][5][6] systems to distribute public keys securely and use the public key encryption or signature to authenticate the parties involved. Some new types of DNS records are introduced, including a TXT format record (delegation record) to store the possible gateways for the source of a traffic flow, a KEY record to store the public key for a party identified by a fully qualified domain name (FQDN) or an IP address, etc. A new specification to store the information used by IPsec in DNS is under development [19]. The exact procedure to establish a session key using the OE is specified in [25][20]. The procedure can be demonstrated by the following example.

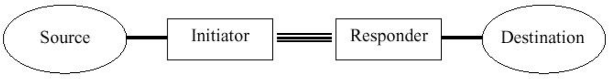

Figure 1. Network Topology

As the topology in Fig. 1, the Source sends out a packet that is intercepted at the Initiator. Given an intercepted packet, the Initiator must according to the Destination IP address of the packet quickly determine the Responder (the Destination's Security Gateway (SG)) and fetch everything (a public key record and a delegation record) needed to authenticate the Responder from a DNS server with security extensions. The Responder must do likewise for the Initiator. To prevent an illegitimate party from initiating an IKE session, phase one and two are connected. In phase two, the Responder should check if the Initiator has the authority to delegate the Source to establish the security association. Due to lack of space, the details of the OE are omitted (the attack presented in the following part follows the OE procedure specified in [25] exactly).

In the OE specification, ID2 acts as ID4 and the OE uses ID2 to retrieve the public key of the party through DNSSEC servers. Hence ID2 is also used as ID3. Therefore the OE faces the same question, i.e. how to guarantee that ID2 is a legitimate representation of ID4. The OE uses the DNSSEC systems to guarantee the authenticity of the public key corresponding to ID2 and uses the TXT record(s) to verify that ID2 is a legitimate delegation for the intended traffic flow. In principle, if both parties eventually confirm that the peer party is authorized to act on behalf of the client host behind it (if any), the man-in-the-middle attack is not feasible. But in the specification, there is still a flaw under certain circumstances, specifically if the OE follows the procedure using step 1, 2B, 3-8, 9B, 10-13 in [25], the man-in-the-middle attack is still feasible based on the following prerequisites (any internal attacker satisfies the requirements):

1. The adversary ID_E has a legitimate KEY record in the DNS system.
2. The adversary ID_E has a legitimate TXT record in the DNS system, which indicates that ID_E is authorized to act as the SG for another source S (which could be a subset or a single host IP address).

The attack can be launched in the following way.

- In steps 5 and 6, the adversary replaces the D-H exchanges of the Responder and Initiator with its own selections respectively.
- In step 7, the adversary replaces the Initiator's ID in the message to the Responder with its own ID ID_E.
- In step 8, the adversary replaces the Responder's ID in the message to the Initiator with its own ID ID_E.

Phase one is broken by the man-in-the-middle attack successfully. The attack is feasible because, for the Responder, the Initiator's ID in step 7 will overrule the KEY result obtained in step 4 and, for the Initiator, in step 9B it uses the Responder's ID in the message sent in step 8 to retrieve the KEY. This attack is not feasible if the OE is in the case of using 2A and 9A because the Initiator will not use the ID of the Responder in the message sent in step 8 to retrieve the KEY record, instead it uses the Responder's IP address (or FQND from the DNSSEC) to retrieve the public key, as in step 2A.

The OE uses delegation authority check (checks whether the ID field is legitimate) in phase two to preclude some types of man-in-the-middle attacks. So the completion of phase one does not guarantee that phase two can be completed. And if the phase-two procedure is not completed successfully, the adversary can obtain some useful information, but cannot read the content of the subsequent communications. However, in the scenario where 2B and 9B is used, the adversary can complete phase two indeed. When the adversary intercepts the message sent by the Initiator in step 10, it can then launch two types of attacks:

- Attack 1. The adversary acts as the Responder to respond to the message sent in step 10 and it can indeed construct the response with the information collected in the previous steps. Once phase two is completed, the adversary can decrypt any message sent from the Source.
- Attack 2. After the operation in Attack 1, the adversary creates a new phase-two message to the Responder with its own authorized source S. Because the adversary uses its own source S, this message passes the Responder's check in step 11. Hence, the adversary creates a security association with the Initiator and the Responder separately. In this attack, the adversary should change the source IP address of all the packets sent from the Source to one IP address selected from its own authorized source S and change the destination IP address of all the packets sent from Responder to the original Source's IP address. At the same time, the adversary must decapsulate and encapsulate the packets between two security associations. In attack 2, the adversary can now read the messages from both sides.

Recently, Thor presented two types of attacks on vendor implementations of IKE [24]. Both cases are related to the ID problem. For example, in the first attack presented by Thor, "validation of certificate authority rather than identity signed for by authority in certificate allows session stealing or 'Server' impersonation". The reason that the attack is feasible is quite simple. If two parties are configured to accept certificates signed by a specific CA instead of specifying the peer ID in the policies, a legitimate party with a valid certificate can impersonate any party in the system. This is a type of attack that should always be kept in mind when implementing IPsec. However, according to the report, it seems that this vulnerability exits in some major vendor implementations.

3 How to Identify a Party

From the arguments in the previous sections, we know that if a party wants to use the ID field in the exchanged messages as an identity of a peer party, the party needs to be able to check that the ID is a valid identification of the intended party. There are three questions that need to be answered. (1) What is an intended party in the protocol? (2) How to identify a party? (3) How to check the validity of an intended party's ID?

To answer question (1), first we need to analyze why IKE should execute phase one to establish ISAKMP SAs. As addressed in Section 1, an ISAKMP SA (in IKEv2, it is called an IKE SA) is used to protect the subsequent IPsec SAs procedures (in IKEv2, they are called CHILD SAs). An IPsec SA is used to protect a packet traffic flow properly defined by traffic selectors. An execution of the IKE protocol is driven by applying a security policy. A normal security policy should specify the following information: Source Selector, SG1, SG2, Destination Selector, Security Process (authentication, encryption, etc.), this endpoint's ID and peer endpoint's ID [13]. If IPsec is used in the transport mode, Source is SG1 and Destination is SG2. The policy specifies that the IP traffic flow from Source to Destination satisfying the two traffic selectors respectively should pass through SG1 and SG2 with the security process. And the identity of SG1 and SG2 for this traffic flow should be the two IDs in the policy. In the OE, this policy is not pre-configured, but SG1 and SG2 should be found through the DNSSEC system and the security process is set as a default configuration and the IDs are the FQDN or IP address of the SGs that are the delegation of the traffic flow. In principle legitimate delegations of endpoints of the traffic flow are the intended parties. Normally these delegations are the security gateways. In IKEv2, the designers consider the "colocated services" scenario [17] where two or more services are on the same SG (Alice or Bob) and state that:

> *"In some cases Bob might host many different services (e.g., distinct web sites with different identities). All these identities would have the same IP address, but would have different keys and certificates. Having Alice initiate a connection to Bob's IP address does not inform Bob whom she wants to communicate with. Therefore, IKEv2 allows Alice to specify an identity for Bob. This feature was given the affectionate name "You Tarzan. Me Jane." by Hugh Daniel. The name is quite appropriate because in the same message in which Alice reveals her identity she requests a specific identity for Bob."*

In this case, the intended party is a service defined by a traffic flow on a security gateway.

For question (2), no matter whether a party is a SG or a service defined by a traffic flow on a SG, we firstly need to identify a SG. In the network, the basic way is to use an IP address or a FQDN. Many other ways can also be used, e.g. a fully-qualified RFC822 email address string, a binary DER encoding of an ASN.1 X.500 general name, etc. A service defined by a traffic flow can be naturally identified by the traffic selectors. Moreover, to authenticate a party, one or more secrets are associated with a party. In the scenario where public keys are used, a party is associated with one or more public/private key pairs. Each key pair should have a unique identification (ID3). One consideration that more than one public/private key pair is associated with a party is to provide "key roll-over", i.e. when there is a requirement for high security, the used key pair should be replaced by a new one after a period. The "colocated services" scenario is a similar case (where) in which, on a single SG different services use different key pairs. The difference between these two cases is that in the "key roll-over" scenario, each valid key pair of a party can be used for all the policies related to the party, but in the "colocated services" scenario, each key pair should be applied in a policy to protect one service traffic flow. And of course, each service can also have two or more keys for "key roll-over". On the other hand, it is common that a security gateway maintains the secrets for all the services on it. In this case, the security gateway works as a legitimate delegation of the services that it hosts. However now it is possible that a party (SG)'s ID is no longer associated with the party's secret and different keys are used for different traffic flows.

Note that a party must know the peer's identification or at least a valid identification set in advance, which means peer's ID or an ID set must be specified in the security policy of a party. Otherwise, the man-in-the-middle attack is obviously feasible, e.g. the examples in Section 2. In principle, if a valid ID set is given, all the private keys bound with the valid IDs in the set must be managed by the same entity (for example the security gateway which hosts the multiple services), otherwise, the man-in-the-middle attack is feasible because a party with an ID in the valid set can impersonate any party whose ID is in the same set. However, we will find that in the dynamic IP scenario, in some cases, we have to accept the usage that the private key of each valid ID is maintained by a separate party. Moreover we can use some amendments to remove the vulnerability introduced by this usage. The newly proposed identity-based cryptosystems [1] provide some valuable flexibility, e.g. using a fixed FQDN appended with an expiry date as the ID of a party, which is at the same time a public key of the party. Hence, without presetting a valid ID in a policy, a party is still able to check the validity of a declared ID in a message sent by a peer party. However, to enjoy this flexibility, we need to sacrifice the generality of ID formats in IKE. Other methods, such as using ID format rules to define a valid ID set, suffer from the same drawback.

Now let us consider question (3), i.e. how to check the validity of the ID field. The ultimate aim of checking the ID field is to guarantee that the peer party is a legitimate delegation of a traffic flow. However an IKE session is divided into two phases and in phase one, a party does not know for which traffic flow peer party is going to build a security channel. Hence what a party can only do in phase one is to bind a policy (or policies) with the current IKE session identified by two cookies (SPIs) and the sender's IP address. To complete the binding, a party can only use the information available including two SGs and two IDs in a policy. And the simplest way to complete the binding securely is to use the ID field to identify a policy (sometimes a bunch of policies related with the same peer party whose identification is the same ID. We refer to these policies as a policy group), through verifying that the peer party has possession of the secret key associated with the ID. We think that it is more reasonable to regard the ID in phase one as the ID of a secret of a potential party instead of an ID of a party. However only completing the binding is not enough to provide security. Two extra steps (at least one step) are needed to guarantee that the ID field is associated with a legitimate SG.

First a SG needs to perform *check 1* to make sure that the source IP address of a received message is the same as the address of peer SG in the policy. Without the address check, SG A can use a victim SG C's IP address to establish an ISAKMP SA with SG B who has a policy using A's ID. It is possible that after establishing the SA, B is configured a new policy with C. Because phase two will reuse the ISAKMP SA established in phase one, if A knows the policy configuration which is not highly sensitive information and not well protected, A can launch a phase-two procedure using the established ISAKMP SA. If B as a responder does not perform *check 2* (B uses the two traffic selectors sent in the messages to determine the peer ID in the corresponding policy and compares the found ID from the policy with the ID used by the ISAKMP SA), A will succeed in impersonating C.

The above attack can be prevented simply by executing the *check 2* procedure in phase two (in transparent mode, the selectors are the IP addresses of two SGs). In fact, the *check 2* procedure connects the two phases and guarantees that a party, as the legitimate delegation of a traffic flow, has possession of the required secret. However we strongly suggest the completion of both checks because it is not reasonable to allow the establishment of an ISAKMP SA for a nonexistent policy in phase one, while on the other hand sometimes *check 1* cannot guarantee the security. For example, in a policy that allows road warrior (RW) access by specifying the peer SG as any IP address (wildcard), a RW sets its IP address as a potential SG C's address, but currently there is no policy related to C in SG B. The RW initiates the phase one procedure with B using an ID pre-configured for all RWs (or its own ID which is in the valid ID set of the policy). The RW can complete phase one because it can pass the ID check and the IP address check (*check 1*). And if the check procedure (*check 2*) in phase two is missing, then obviously the impersonation attack is feasible. FreeSwan applies a method similar to *check 1* in phase one but checks IP addresses first, and then matches the ID in the found policies.

For the OE scenario, there is no pre-configured policy. The security gateway is found by using the destination IP address of a traffic flow to retrieve the delegation record (TXT) from a DNSSEC server. The intended party is identified by the IP address or FQDN of the gateway. And a public key is associated with the gateway's ID. To prevent an illegitimate party from initiating an IKE session, phase one and two are connected. In phase two, the Responder should check if the Initiator has the authority to delegate the Source to establish the security association. The attack in Section 3 cannot be precluded because the Initiator does not have enough information to check the validity of the Responder's ID. Fortunately, this scenario corresponds to the case in which a Source tries to establish a security association with a road warrior, and this case hardly ever happens in a real network. The common scenario is that a road warrior is an initiator of the protocol. Some people suggest using the OE to establish SAs between two road warriors. We do not think this is a prudent suggestion.

We can interpret the above procedure in another way as follows. In IKE, every party should transmit two identities to the peer party, one in phase one and the other in phase two. ID2 in phase one is used as ID3 that is the identity of a secret, while ID2 in phase two acts as ID4 that is the traffic flow selector. Two identities normally have no direct relation and are bound together by a preset policy or a delegation TXT record in a DNSSEC system. If we use the identity-based cryptosystems, e.g. [1][21][22], it is convenient to use a traffic selector as a public key. Hence, ID3 equals to ID4. This will reduce the system complexity and we also can use this method to implement the OE without the complicated DNSSEC system.

4 The Implications of Binding IDs

If a SG has pre-configured policies with IDs specified, we argue that in most cases there is no need to transmit an ID field in the main mode exchange. A SG can use the IP addresses (source and destination: ID1) in the packets to bind a security policy (or a policy group) with the current phase-one session. The basic step is to use two IP addresses of a message to identify a policy or a policy group of which the two security gateways use the same IP addresses as the one of the message.

The analysis of the following cases supports our point.

1. A security gateway has only one public/private key pair associated with any traffic flow through it. Then the IP addresses of two SGs uniquely identify a policy of a policy group that uses the same ID. Hence the IP addresses can be used to bind a policy (more likely a bunch of policies using the same two SGs) with the current IKE session and to find the uniquely specified ID.
2. There are two or more public/private key pairs associated with a party and the key pairs are used simultaneously. Because a party must know the valid public key set or public key ID set (maybe a possible valid set but checkable through interacting with the public key infrastructure (PKI)) associated with the peer party, it can try each public key in the valid set to check the signature. And one of the keys should be able to verify the signature (this method is adopted in the main mode with public key encryption in IKEv1, in which the Initiator's ID field is encrypted by one of the Initiator's public keys). In practice, this situation rarely happens. Even in the "key roll-over" scenario, in a specific period normally only one valid key pair is used which can be identified uniquely as in case 1.
3. In the "colocated services" scenario introduced by IKEv2, because normally only one key pair is associated with a service and the service traffic selectors are transmitted in message 3 and 4 in IKEv2, it is easy to use the SGs' IP addresses and the traffic selectors to bind a policy and determine the ID uniquely. (The procedure is no longer called "main mode" in IKEv2.)
4. This approach is not applicable directly if a security gateway is behind a network address translator (NAT), because the source IP address will be changed when packets pass through a NAT. [12] introduces a NAT-OA (NAT Original Address) payload which can be used to send the original address in a phase-two procedure. We can simply use the NAT-OA in message 3 or 4 in phase one and use the original IP address in the NAT-OA instead of the IP address in the IP packet to bind policies. The NAT-OA should be authenticated.
5. Normally this approach is not applicable to the road warrior situation, because in the policies for road warriors one party of the protocol is not specified explicitly, instead it is configured as a wildcard. Hence, the policies cannot even be used to uniquely determine a party pair. In this type of policies, if all the RWs use the same ID, the method is feasible. However, a valid ID set specified by an ID format rule is commonly used. Normally it is impossible to enumerate all the valid IDs in the set. Hence transmitting the ID field in the messages is necessary.

As far as the non-configured OE is concerned, in most cases, there is no need to transmit the ID field in the main mode, because the ID is the IP address or FQDN of the security gateway and the resolution of a FQDN to an IP address is protected by the DNSSEC system. However if the Initiator is a road warrior, the Initiator's ID field in the message is necessary, because the Initiator's IP address is dynamically assigned which cannot be used to identify the Initiator and there is no pre-configured policy to determine the ID.

Overall, to securely use the ID field in IKE, a party should use the ID to bind a policy (or policies) with the current IKE session and apply the two check procedures in phase one and two respectively. However, using the IP addresses in an IP packet instead of the ID field, in most cases, is enough for the secure completion of the binding. We argue that without the ID field in the main mode, the protocol's security is not affected, although there are plenty of attacks due to name omission, because the ID check still can implicitly be completed successfully. Moreover, a SG can use the two IP addresses of an IP packet in place of the ID field in all the related computations (if the NAT is used, the original address in the NAT-OA is applied). Hence, in practice the ID field in the main mode messages is not necessary in most cases. There are some advantages in having the ID field as an option. (1) Reducing the ID process is a minor improvement on the huge complexity of IKE. (2) One design rationale of IKE is "hiding the ID". However as stated in [18], it is impossible to successfully hide the identities under active attacks. But using IP addresses to bind a policy (or policies with the same ID) with an IKE session is indeed able to hide the identities of two endpoints because the IDs are determined implicitly. (3) Applying the IP addresses in the related computation in place of IDs can help prevent the authentication failure attack presented in [14].

5 Conclusion

As a mandatory field in the IKE protocol, ID is used to identify an involved party in a protocol session. But how to use the ID field to identify a party is far from straightforward because a party's identity is no longer directly related to the party's key identity. In practice, some mistakes are made when using the ID field. In this paper, we clarify the problem of using the ID field in practice and present the two check procedures to prevent the ID related attacks. Moreover we find that in most cases, the ID field is not necessary in the main mode exchange and therefore it might have the ID field as an optional field in the IKE protocol instead of a mandatory one. By using IP addresses to bind policies, IKE can achieve extra security properties.

References:

[1] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, *Advances in Cryptology - Crypto'2001*, LNCS 2139.

[2] R. Canetti and H. Krawczyk, Analysis of Key Exchange Protocols and Their Use for Building Secure Channels, *Advances in Cryptology-EUROTCRYPT 2001.*

[3] R. Canetti and H. Krawczyk, Security Analysis of IKE's Signature-based Key-Exchange Protocol, *The Proceedings of Crypto' 2002.*

[4] D. Eastlake and C. Kaufman, Domain Name System Security Extensions, IETF RFC 2065, Jan. 1997.

[5] D. Eastlake, Domain Name System Security Extensions, IETF RFC 2535, Mar. 1999.

[6] D. Eastlake, DNS Request and Transaction Signatures (SIG(0)s), IETF RFC 2931, Sep. 2000.

[7] D. Harkins and D. Carrel, The Internet Key Exchange Protocol (IKE), IETF RFC 2409, Nov. 1998

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

[9] S. Kent and R. Atkinson, IP Authentication Header (AH), IETF RFC 2402 Nov. 1998

[10] S. Kent and R. Atkinson, IP Encapsulating Security Pay-load (ESP), IETF RFC 2406, Nov. 1998

[11] C. Kaufman, Internet Key Exchange (IKEv2) Protocol, version 11, Oct. 2003

[12] T. Kivinen, B. Swander, A. Huttunen and V. Volpe, Negotiation of NAT-Traversal in the IKE, IETF draft version 7, 29 Sep 2003.

[13] J. Jason, L. Rafalow and E. Vyncke, IPsec Configuration Policy Information Model, IETF RFC 3585, August 2003

[14] W. Mao, Moden Cryptography Theory and Pratice, pp.398-340, Pearson Education, 2003.

[15] D. Maughan et al., Internet Security Association and Key Management Protocol (ISAKMP), IETF RFC 2408, Nov. 1998

[16] D. Piper, The Internet IP Security Domain of Interpretation for ISAKMP, IETF RFC 2407, Nov. 1998

[17] R. Perlman, Understanding IKEv2: Tutorial, and rationale for decisions, version 1, Feb. 2003.

[18] R. Perlman and C. Kaufman, Key Exchange in IPsec: Analysis of IKE, *IEEE Internet Computing*, Nov. 2000.

[19] M. Richardson, A method for storing IPsec keying material in DNS, version 7, Sep. 2003.

[20] M. Richardson and D. Redelmeier, Opportunistic Encryption using The Internet Key Exchange (IKE), version 12, Jun. 2003.

[21] M. Scott, Authenticated ID-based Key Exchange and remote log-in with insecure token and PIN number, Cryptology ePrint Archive, Report 2002/164

[22] N. P. Smart, An Identity Based Authenticated Key Agreement Protocol Based on the Weil Pairing, Electronics Letters 38 (2002), pp. 630-632.

[23] D. K. Smetters and G. Durfee, Domain-Based Administration of Identity-Based Cryptosystems for Secure Email and IPSEC, 12th USENIX Security Symposium, Washington, DC, 2003

[24] Thor Lancelot Simon, Multiple vulnerabilites in vendor IKE implementations including Cisco, available on http://seclists.org/lists/bugtraq/2003/Dec/0199.html, December 11, 2003

[25] H. Spencer and D. H. Redelmeier, Opportunistic Encryption, May 2001, available on http://www.freeswan.org