# Internet Key Exchange (IKE) with MQV

Yongge Wang
Certicom Research (ywang@certicom.com)

July 19, 2004

**Abstract**

MQV [3] is an efficient protocol for authenticated key agreement in the asymmetric (public-key) setting. The protocol is based on Diffie-Hellman key agreement protocol and can be modified to work in an arbitrary finite group. The MQV protocol has been standardized in ANSI X9.42 [1], ANSI X9.63 [2], and IEEE 1363 [5]. In this note, we will present an MQV-based IKE variant which improves the efficiency of IKE key negotiation.

## 1 Introduction

Internet Key Exchange (IKE) protocol (see RFC 2409 [4]) is a hybrid protocol. It is based on a framework defined by the Internet Security Association and Key Management Protocol (ISAKMP) (see RFC 2408 [7]) and implements parts of two key management protocols—Oakley [8] and SKEME [6]. In addition IKE defines two exchanges of its own. IKE defines how security parameters are negotiated and shared keys are established for other protocols. For an application in IPSec, the IPSec DOI [10] specifies those attributes that need to be negotiated through IKE for IPSec security associations (SAs).

MQV [3] is an efficient protocol for authenticated key agreement in the asymmetric (public-key) setting. The protocol is based on Diffie-Hellman key agreement protocol and can be modified to work in an arbitrary finite group. The MQV protocol has been standardized in ANSI X9.42 [1], ANSI X9.63 [2], and IEEE 1363 [5]. In this note, we will present an MQV-based IKE variant which improves the efficiency of IKE key negotiation.

## 2 IKE protocol

Internet Key Exchange (IKE) protocol is based on a framework defined by the Internet Security Association and Key Management Protocol (ISAKMP) and implements parts of two key management protocols—Oakley [8] and SKEME [6]. ISAKMP defines how two peers communicate, how the message they use to communicate are constructed, and the state transitions they go through to secure their communication. It provides the means to authenticate a peer, to exchange information for a key exchange, and to negotiate security services.

Messages exchanged in an ISAKMP-based key management protocol are constructed by chaining together ISAKMP payloads to an ISAKMP header. There are 13 distinct payloads defined in ISAKMP. The type of ISAKMP payload that follows the current payload is denoted by the next payload field in the ISAKMP generic payload header. The distinct payloads are: security association payload (SA 1), proposal payload (P 2), transform payload (T 3), key exchange payload (KE 4), identification payload (ID 5), certificate payload (CERT 6), certificate request payload (CR 7), hash payload (HASH 8), signature payload (SIG 9), nonce payload (NONCE 10), notification payload (N 11), delete payload (D 12), and vender ID payload (VID 13).

ISAKMP describes two separate phases of negotiation. In the first phase, peers establish an authenticated and secure channel between themselves. In the second phase that authenticated and secure channel is used to negotiate security services for a different protocol like IPSec.

In the IKE specification, phase I can either be in the Main Mode (corresponding to the Identity Protection exchange type of ISAKMP, the value is 2) or the Aggressive Mode (corresponding to the Aggressive exchange type of ISAKMP, the value is 4). The phase II must be in the Quick Mode. There is no corresponding Quick Mode in ISAKMP specification. The value for Quick Mode is 32.

When using public key signatures, the aggressive mode is shown in Figure 1.

| Initiator | | Responder |
|---|---|---|
| Header, SA, KE, Ni, IDii | → | |
| | ← | Header, SA, KE, Nr, IDir, [CERT, ] SIG_R |
| Header, [CERT, ] SIG_I | → | |

Figure 1: Aggressive Mode

The main mode is shown in Figure 2.

| Initiator | | Responder |
|---|---|---|
| Header, SA | → | |
| | ← | Header, SA |
| Header, KE, Ni | → | |
| | ← | Header, KE, Nr |
| Header*, IDii, [CERT, ] SIG_I | → | |
| | ← | Header*, IDir, [CERT, ] SIG_R |

Figure 2: Main Mode

In the above messages, the notation Header* indicates payload encryption.

In the phase I of IKE, the peers will generate four secrets: SKEYID, which is the secret on which all subsequent secrets are based; SKEYID$_d$, which is used to derive keying material for IPSec and other SAs; SKEYID$_a$, which is used to provide data integrity and data source authentication to IKE messages; and, SKEYID$_e$, which is used to encrypt IKE messages. The generation of SKEYID is dependent on the authentication method negotiated. All other SKEYID-based secrets are generated identically, regardless of authentication method.

## 3 MQV key agreement

For a general Diffie-Hellman key agreement protocol, two exponents $g^x$ and $g^y$ are exchanged first, then the shared secret $g^{xy}$ is computed separately. The two peers authenticate themselves to each other by signing the exponents. A simple analysis shows that each party needs to compute five exponentiations at least.

The MQV key agreement protocol eliminates the authentication step by using a concept of implicit authentication, thus reduce the number of exponentiation computations. Specifically, let $g$ be the generator of the group, $q$ be the order of $g$, $f$ be any predetermined function, and $h$ be a constant. For two peers *initiator* and *responder* whose private/public key pairs are $(s_i, g^{s_i})$ and $(s_r, g^{s_r})$ respectively, the MQV protocol proceeds as follows:

1. The *initiator* generates a random integer $x_i$, $1 \leq x_i \leq n-1$, computes $R_i = g^{x_i}$, and sends this to *responder*.
2. The *responder* generates a random integer $x_r$, $1 \leq x_r \leq n-1$, computes $R_r = g^{x_r}$, and sends this to the *initiator*.
3. The *initiator* computes
$$k_i = (x_i + f(R_i)s_i) \bmod n$$
and
$$K = (R_r \cdot g^{s_r \cdot f(R_r)})^{hk_i}.$$
4. The *responder* computes
$$k_r = (x_r + f(R_r)s_r) \bmod n$$
and
$$K = (R_i \cdot g^{s_i \cdot f(R_i)})^{hk_r}.$$

For elliptic curves, the function $f(X)$ can be defined as the the second half part of the first coordinate of $X$, and the $h$ is the cofactor. For other groups, the function $f$ can be defined similarly and $h = 1$.

## 4 Embedding MQV into IKE

In this section, we show how to set up an SA for IKE with the MQV implicit authentication method. In phase I of ISAKMP, the following attributes are negotiated as part of the ISAKMP security association. (These attributes pertain only to the ISAKMP security and not to any security associations that ISAKMP may be negotiating on behalf of other services).

- encryption algorithm
- hash algorithm
- authentication method
- information about a group over which to do Diffie-Hellman

All of these attributes are mandatory and MUST be negotiated. In addition it is possible to optionally negotiate a pseudo-random function ("PRF").

The currently supported authentication methods which may be negotiated are:

- pre-shared key (value 1)
- DSS signatures (value 2)
- RSA signatures (value 3)
- encryption with RSA (value 4)
- revised encryption with RSA (value 5)
- encryption with El-Gamal (value 6)
- revised encryption with El-Gamal (value 7)
- ECDSA signatures (value 8)

Values 9–65000 are reserved to IANA. Values 65001–65535 are for private use among mutually consenting parties.

Phase I can be either Main Mode or Aggressive Mode. For MQV implicit authentication in IKE, we have two choices: We can either reserve the value 9 (or others) from IANA or use the private value 65001. In this note, we will use the second choice.

Note that, like any other digital signature authentication methods, MQV implicit authentication requires the parties to know and trust each other's public key. This can be done by exchanging certificates, possibly within the Phase 1 negotiation, if the public keys of the parties are not already known to each other.

Since MQV requires the use of SHA-1 hash function, implementers may find it convenient to specify SHA-1 as the value of the hash algorithm attribute when using MQV as the authentication method. Implementers may also find it convenient to use MQV implicit authentication in conjunction with an elliptic curve group for the IKE key agreement.

$$\begin{array}{rcl} \underline{\text{Initiator}} & & \underline{\text{Responder}} \\ \text{Header, SA, KE, Ni, IDii} & \longrightarrow & \\ & \longleftarrow & \text{Header, SA, KE, Nr, IDir, [CERT, ] HASH-R} \\ \text{Header, [CERT, ] HASH-I} & \longrightarrow & \end{array}$$

Figure 3: Aggressive Mode with MQV

### 4.1 Aggressive mode

An aggressive mode phase I communication with MQV implicit authentication is shown in Figure 3.

where HASH-R is the authenticating hash of responder and HASH-I is the authenticating hash of initiator, which together authenticate the exchange.

Assume that SHA-1 is chosen as the hash function and HMAC-SHA-1 is the chosen PRF. Let $R_i = g^{x_i}$ and $R_r = g^{x_r}$ be the exponents included in the key exchange payloads and let

$$\text{K} = g^{h(x_i+f(R_i)s_i)(x_r+f(R_r)s_r)}$$

where $s_i$ and $s_r$ are the secret keys of initiator and responder respectively. Then

$$\begin{array}{rcl} \text{SKEYID} & = & \text{PRH}(nonce_i|nonce_r, \text{K}), \\ \text{HASH-I} & = & \text{PRF}(\text{SKEYID}, R_i|R_r|\text{CKY-I}|\text{CKY-R}|\text{SA-offer}|\text{ID}_i), \\ \text{HASH-R} & = & \text{PRF}(\text{SKEYID}, R_r|R_i|\text{CKY-R}|\text{CKY-I}|\text{SA-offer}|\text{ID}_r). \end{array}$$

In IKE, the derivations of SKEYID$_d$, SKEYID$_a$, and SKEYID$_e$ are independent of authentication methods. However, if MQV implicit authentication method is used, then the derivations of these values should have corresponding changes. Specifically, they will be derived as follows.

$$\begin{array}{rcl} \text{SKEYID}_d & = & \text{PRF(SKEYID, K}|\text{CKY-I}|\text{CKY-R}|0), \\ \text{SKEYID}_a & = & \text{PRF(SKEYID, SKEYID}_d|\text{K}|\text{CKY-I}|\text{CKY-R}|1), \\ \text{SKEYID}_e & = & \text{PRF(SKEYID, SKEYID}_a|\text{K}|\text{CKY-I}|\text{CKY-R}|2). \end{array}$$

Note the the only difference from the IKE specification is to replace the value $g^{x_i x_r}$ with the new value K.

### 4.2 Main mode

The function of the aggressive mode is limited. Due to message construction requirements the group, on which the MQV key exchange is based, cannot be negotiated. In addition, the IDs of the peers are exchanged in clear texts. For situations where the rich attribute negotiation capabilities of IKE are required or the ID protection is required, the main mode may be required. A main mode phase I communication with MQV implicit authentication is shown in Figure 4.

Assume that SHA-1 is chosen as the hash function and HMAC-SHA-1 is the chosen PRF. Let $R_i = g^{x_i}$ and $R_r = g^{x_r}$ be the exponents included in the key exchange payloads. Then the values of K, SKEYID,

| Initiator | | Responder |
|---|---|---|
| Header, SA | → | |
| | ← | Header, SA |
| Header, KE, Ni | → | |
| | ← | Header, KE, Nr |
| Header*, IDii, [CERT, ] HASH-I | → | |
| | ← | Header*, IDir, [CERT, ] HASH-R |

Figure 4: Main Mode with MQV

HASH-I, HASH-R, SKEYID$_d$, and SKEYID$_a$ are defined exactly the same way as in the aggressive mode (see section 4.1). The only difference is the definition of SKEYID$_e$ which is used to derivate the excryption keys. Note that before the *initiator* or the *responder* sends her third message, they may not know the ID of the other peer. Hence they have to use a different encryption key for their third messages. In this case, we define SKEYID$_e$ as follows:

$$\begin{array}{lcl} \text{SKEYIDTEMP} & = & \text{PRH}(nonce_i|nonce_r, g^{x_i x_r}), \\ \text{SKEYID}_e & = & \text{PRF(SKEYIDTEMP, } g^{x_i x_r}|\text{CKY-I}|\text{CKY-R}|2). \end{array}$$

### 4.3 Advantages and discussions

For an IKE phase I key exchange with a DSS or ECDSS signature scheme, in addition to the two exponentiations to compute $g^{x_i}$ (or $g^{x_r}$) and $g^{x_i x_r}$), one exponentiation is needed for signing the message and two exponentiations are needed for verifying the signature. Hence an IKE phase I exchange (either aggressive mode or main mode) authenticated with DSS or ECDSS signature scheme requires five exponentiations for each side. For an aggressive mode phase I exchange authenticated with MQV implicit authentication, only three exponentiations (2.5 indeed) are needed for each side (one for the computation of $g^{x_i}$ or $g^{x_r}$, and two for K). For a main mode phase I exchange authenticated with MQV implicit authentication, four exponentiations (3.5 indeed) are needed for each side (one for the computation of $g^{x_i}$ or $g^{x_r}$, one for $g^{x_i x_r}$, and two for K). It should also be noted that messages exchanged have the same size in the both cases.

As a summary, we list the properties of different modes (with or withour MQV) in Table 1. In the table, AM stands for Aggressive Mode, MM stands for Main Mode, and PFS stands for Perfect Forward Secrecy. We do not count the steps for verifying the certificates.

Table 1: A comparison

| | AM (ECDSA) | MM (ECDSA) | AM (MQV) | MM (MQV) |
|---|---|---|---|---|
| ID protection | no | yes | no | yes |
| PFS | yes | yes | yes | yes |
| SA negotiation | limited | flexible | limited | flexible |
| Number of exp. | 5 | 5 | 2.5 | 3.5 |

## 5 An example IKE with MQV in aggressive mode

In this section, we will give an example of setting up Security Associations for IPSec with the aggressive mode in IKE phase I. The main mode will be similar. For this example, the IKE phase I is implicitly authenticated with MQV.

The protocol begins when the *initiator* send the message in Table 2 to the *responder*.

Table 2: Initiator's first message

| 0–3 | 4–7 | 8–11 | 12–15 | 16–19 | 20–23 | 24–27 | 28–31 |
|---|---|---|---|---|---|---|---|
| Initiator cookie | | | | | | | |
| Responder cookie $= 0$ | | | | | | | |
| NPL (SA 1) | | MAV | MIV | XCHG (4) | | Flags (C) | |
| Message ID $= 0$ | | | | | | | |
| Length | | | | | | | |
| NPL (KE 4) | | RESERVED | | SA Payload length | | | |
| Domain of Interpretation (1) | | | | | | | |
| Situation (SIT_IDENTITY_ONLY) | | | | | | | |
| NPL (0) | | PROTO_ISAKMP | | SPI size $= 0$ | | # of T = 1 | |
| NPL (0) | | RESERVED | | T Payload length | | | |
| T # = 1 | | KEY_IKE | | RESERVED | | | |
| AF=1, Encryption Algorithm = 1 | | | | DES-CBC = 1 | | | |
| AF=1, Hash Algorithm = 2 | | | | SHA = 2 | | | |
| AF=1, Authentication method = 3 | | | | MQV = 65001 | | | |
| AF=1, Group Description = 4 | | | | EC2N163 = 6 | | | |
| NPL (NC 10) | | RESERVED | | KE Payload length | | | |
| Key exchange data (one EC point $R_i$) | | | | | | | |
| NPL (ID 5) | | RESERVED | | Nonce Payload length | | | |
| Nonce data Ni | | | | | | | |
| NPL (0) | | RESERVED | | ID Payload length | | | |
| ID_FQDN | | Protocol ID = 0 | | Port = 0 | | | |
| Identification Data = palm.security.com | | | | | | | |

The message in Table 2 consists of the generic ISAKMP header, the Security Association (SA) Payload followed by one Proposal Payload, one Transform Payload, the Key Exchange (KE) Payload, the Nonce Payload, and the Identification (ID) Payload. The following is an explanation of some of the values:

- XCHG value is 4 which means that aggressive mode is used.
- The Domain of Interpretation value in the SA Payload is 1 which means that this IKE is used for IPSec.
- The Situation value in the SA Payload is SIT_IDENTITY_ONLY which means that the security association will be identified by source identity information present in the associated Identification Payload.

- Note that the ID Payload uses the fully-qualified domain name (ID_FQDN) string palm.security.com as its identity.

After receiving the above message from *initiator*, if the *responder* accepts the offer in the SA and agrees to go on with the protocol, it may send back a message as in Table 3.

Table 3: Responder's first message

| 0–3 | 4–7 | 8–11 | 12–15 | 16–19 | 20–23 | 24–27 | 28–31 |
|---|---|---|---|---|---|---|---|
| Initiator cookie | | | | | | | |
| Responder cookie | | | | | | | |
| NPL (SA 1) | | MAV | MIV | XCHG (4) | | Flags (C) | |
| Message ID $= 0$ | | | | | | | |
| Length | | | | | | | |
| NPL (KE 4) | | RESERVED | | SA Payload length | | | |
| Domain of Interpretation (1) | | | | | | | |
| Situation (SIT_IDENTITY_ONLY) | | | | | | | |
| NPL (0) | | PROTO_ISAKMP | | SPI size $= 0$ | | # of T = 1 | |
| NPL (0) | | RESERVED | | T Payload length | | | |
| T # = 1 | | KEY_IKE | | RESERVED | | | |
| AF=1, Encryption Algorithm = 1 | | | | DES-CBC = 1 | | | |
| AF=1, Hash Algorithm = 2 | | | | SHA = 2 | | | |
| AF=1, Authentication method = 3 | | | | MQV = 65001 | | | |
| AF=1, Group Description = 4 | | | | EC2N163 = 6 | | | |
| NPL (NC 10) | | RESERVED | | KE Payload length | | | |
| Key exchange data (one EC point $R_r$) | | | | | | | |
| NPL (ID 5) | | RESERVED | | Nonce Payload length | | | |
| Nonce data Nr | | | | | | | |
| NPL (HASH 8) | | RESERVED | | ID Payload length | | | |
| ID_FQDN | | Protocol ID = 0 | | Port = 0 | | | |
| Identification Data = server.security.com | | | | | | | |
| NPL (0) | | RESERVED | | HASH Payload length | | | |
| Authenticating hash data | | | | | | | |

The message in Table 3 consists of the generic ISAKMP header, the Security Association (SA) Payload followed by one Proposal Payload, one Transform Payload, the Key Exchange (KE) Payload, the Nonce Payload, the Identification (ID) Payload, and the Hash (HASH) Payload. The explanation of the values are similar to those in Table 2. The HASH Payload is used for the responder to authenticate herself.

After the initiator gets the message from responder, he will carry out some computation and verification. If he agrees to go on, he will authenticate himself and send the message as in Table 4 to the responder.

After Phase I ends successfully, the two peers have set up a secure channel (an SA on each side) for further communication, and Phase II can begin. In Phase II, MQV implicit authentication method is not needed since it offers no advantage compared with traditional Diffie-Hellman key exchange method.

Table 4: Initiator’s second message

| 0–3 | 4–7 | 8–11 | 12–15 | 16–19 | 20–23 | 24–27 | 28–31 |
|---|---|---|---|---|---|---|---|
| Initiator cookie | | | | | | | |
| Responder cookie | | | | | | | |
| NPL (HASH 8) | | MAV | MIV | XCHG (4) | | Flags (C) | |
| Message ID $= 0$ | | | | | | | |
| Length | | | | | | | |
| NPL (0) | | RESERVED | | HASH Payload length | | | |
| Authenticating hash data | | | | | | | |

## 6 IKE Aggressive Mode variants

In previous sections, we proposed a fully IKE-compatible protocol with MQV authentication. The Aggressive Mode with MQV authentication requires that the *responder* knows the public key of the *initiator* in advance. This will limit the application of the protocol. In this section, we propose several IKE-variant protocols which overcome these problems. The cost we have to pay is that our new protocols do not conform to the IKE specifications.

### 6.1 IKE-variant I

An aggressive mode Phase I communication with MQV implicit authentication in an IKE-variant I is shown in Figure 5:

| Initiator | | Responder |
|---|---|---|
| Header, SA, KE, Ni, [CERT, ] IDii | → | |
| | ← | Header, SA, KE, Nr, IDir, [CERT, ] HASH-R |
| Header, HASH-I | → | |

Figure 5: IKE-variant I

The only difference from the IKE specification is that the *initiator* sends her certificate in the first message. Except for this difference, the protocol proceeds according to the standard IKE specification.

### 6.2 IKE-variant II

An aggressive mode Phase I communication with MQV implicit authentication in an IKE-variant II is shown in Figure 6.

In this protocol, the NONCE payload is replaced with the CERT payload. In all following computations, the peers takes $nonce_i = g^{x_i}$ and $nonce_r = g^{x_r}$. This will reduce the messages sent by each peer and keep all properties of IKE protocol: perfect forward secrecy, etc.

| Initiator | | Responder |
|---|---|---|
| Header, SA, KE, CERT, IDii | → | |
| | ← | Header, SA, KE, CERT, IDir, HASH-R |
| Header, HASH-I | → | |

Figure 6: IKE-variant II

### 6.3 IKE-variant III

In this variant, we suggest that the IDii and IDir be encrypted with a previous established session key. In order for the peers to decrypt the IDs, the "Message ID" field can be used to identify the key. Note that in IKE specification, the Message ID must be set 0 in all Phase I communications.

### 6.4 Comments

Since security association parameters are fixed for most applications, the main disadvantage of Aggressive Mode is the ID protection problem. But we should be aware that the Main Mode canot provide absolute ID protection either. A man-in-the-middle attack can easily get the ID of the *initiator* though he may fail to get any other useful information.

## References

[1] ANSI X9.42. Agreement of symmetric algorithm keys using Diffie-Hellman. Working draft, September 1997.

[2] ANSI X9.63. Elliptic curve key agreement and key transport protocol. Working draft, October 1997.

[3] Certicom Corp. SEC1: Elliptic Curve Cryptography. Standards for Efficient Cryptography.

[4] D. Harkins, D. Carrel. The internet key exchange (IKE). RFC 2409.

[5] IEEE 1363. Standard specifications for public-key cryptography. Working draft, November 1997.

[6] H. Krawczyk. SKEME: a versatile secure key exchange mechanism for Internet. In: *IEEE Proc. of Symp. on Network and Distributed Systems Security*, 1996.

[7] D. Maughan, M. Schertler, M. Schneider, and J. Turner. Internet security association and key management protocol (ISAKMP). RFC 2408.

[8] H. Orman. The Oakley key determination protocol. RFC 2412, November, 1998.

[9] P. Panjwani and Y. Poeluev. Additional ECC groups for IKE. Internet-Draft.

[10] D. Piper. The Internet IP security domain of interpretation for ISAKMP. RFC 2407.