MAKALAH IF5054 - KRIPTOGRAFI
STUDI MENGENAI MQV (Menezes-Qu-Vanstone)

Oleh:
Yosep Kurniawan – 13503059

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2006

STUDI MENGENAI MQV (Menezes-Qu-Vanstone)

Yosep Kurniawan – NIM : 13503059
Program Studi Teknik Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
E-mail : if13059@students.if.itb.ac.id

Abstrak

Makalah ini membahas tentang studi mengenai MQV (Menezes-Qu-Vanstone). MQV adalah sebuah protokol kriptografi untuk konfirmasi pertukaran kunci berdasarkan skema Diffie-Hellman. Tidak seperti Diffie-Hellman, MQV menyediakan perlindungan terhadap serangan yang aktif. Secara umum, protokol ini dapat diubah untuk dapat bekerja pada sebuah kelompok terbatas yang berubah-ubah dan secara khusus untuk kelompok *elliptic curve* atau lebih dikenal dengan ECMQV. MQV pertama kali diusulkan oleh Menezes, Qu dan Vanstone pada tahun 1995, lalu diperbaiki oleh Law dan Solinas pada tahun 1998. MQV ini dimasukkan ke dalam standar kunci-publik IEEE P1363. MQV dipatenkan oleh Certicom.

Kata kunci: MQV, Menezes, Diffie-Hellman, Certicom, *elliptic curve*.

1. Pendahuluan

Banyak sistem saat ini membutuhkan algoritma kriptografi yang handal untuk melindungi integritas dan kerahasiaan dari data. Algoritma simertri seperti Advanced Encryption Standard (AES) 128-bit dapat memberi perlindungan yang kuat untuk lebih dari tahun 2036. Namun demikian, untuk menggunakan algoritma ini dibutuhkan pengadaan pertukaran kunci sebelumnya.

Walaupun kunci dapat ditukarkan secara manual dengan menggunakan jasa kurir, namun hal ini tidak lagi dapat dipraktekkan apabila jumlah pengguna sistem berkembang dan sistem juga berkembang. Oleh karena itu, algoritma kriptografi juga perlu didukung dengan skema pertukaran kunci.

Skema pertukaran kunci adalah suatu teknik yang digunakan oleh sekelompok orang yang ingin menggunakan komunikasi yang aman di mana kedua belah pihak yang berkomunikasis juga berkontribusi dalam pengadaan kunci rahasia bersama tersebut. Kunci rahasia bersama ini digunakan untuk menurunkan kunci simetri, yang kemudian digunakan untuk membuat saluran komunikasi yang aman. Metode dengan membangkitkan kunci rahasia bersama ini dikenal dengan skema pertukaran kunci dan skema ini sangat berguna untuk aplikasi di mana kelompok-kelompoknya mempertukarkan data dalam waktu yang *real-time*. Skema pertukaran kunci harus menjamin bahwa tidak ada orang ketiga (yang tidak diinginkan oleh sistem) yang akan mendapatkan kunci rahasia bersama tersebut.

Para kriptografer selalu berusaha untuk menemukan suatu algoritma yang handal dan sulit untuk dipecahkan atau bahkan tidak dapat dipecahkan oleh kriptanalis. Selain dalam mengenkripsi isi dari pesan, perlu pula diperhatikan protokol pertukaran kunci yang digunakan oleh pengirim dan penerima. Protokol pertukaran kunci ini banyak digunakan antara lain untuk membangkitkan rangkaian bilangan acak, otentikasi pengirim atau penerima, dll. Sebagai contoh pada aplikasi *e-mail* saat ini, telah banyak vendor e-mail yang memiliki fitur untuk mengotentikasi pengirim. Hal ini penting mengingat peredaran *virus* maupun *spam* melalui *e-mail* kian hari kian meningkat.

2. Protokol Kriptografi

Protokol adalah aturan yang berisi rangkaian langkah-langkah, yang melibatkan dua atau lebih orang, yang dibuat untuk menyelesaikan suatu kegiatan. Protokol kriptografi adalah protokol yang menggunakan kriptografi. Orang yang berpartisipasi dalam protokol kriptografi memerlukan protokol tersebut misalnya untuk berbagi komponen rahasia untuk menghitung sebuah nilai, membangkitkan rangkaian bilangan acak, meyakinkan identitas orang lainnya (otentikasi), dan sebagainya.

DLC (*Discrete Logarithmic Cryptography*) terdiri dari FCC (*Finite Field Cryptography*) dan ECC (*Elliptic Curve Cryptography*). DLC membutuhkan pasangan kunci publik dan kunci privat sebagai bagian dari parameter nilai awal. Pengguna dari parameter domain

sebelum menggunakannya harus menjamin bahwa parameter tersebut adalah valid. Meskipun parameter domain adalah informasi yang bersifat publik, mereka sebaiknya mengatur hal ini agar pasangan kunci dan kumpulan domain parameter ini dapat dipertahankan oleh semua pengguna pasangan kunci tersebut. Parameter domain mungkin tetap sama untuk periode waktu tertentu, dan satu kumpulan domain parameter dapat digunakan untuk beberapa skema pertukaran kunci.

2.1 Pembangkitan Parameter Domain FCC

Parameter domain untuk skema FCC memiliki bentuk:

(p, q, g, [*SEED*, *pgenCounter*])

dengan

p = bilangan prima (besar)

q = bilangan prima, q < p

g = fungsi pembangkit q

SEED dan *pgenCounter* = nilai yang digunakan untuk membangkitkan dan memvalidasi p dan q

Berikut adalah tabel yang menunjukkan bahwa kunci FFC yang lebih panjang akan memberikan jaminan keamanan yang lebih namun membutuhkan waktu dan "ruang" untuk menggunakannya.

Bits of Security	*Bit length of subgroup order q*	*Bit length of field order p*
80	160	1024
112	224	2048
128	256	3072
192	384	8192
256	512	15360

2.2 Pembangkitan Parameter Domain ECC

Parameter domain untuk skema ECC memiliki bentuk:

(q, FR, a, b, [*SEED*], G, n, h)

dengan

q = ukuran *field*, dapat berupa bilangan prima ganjil atau 2^m, di mana m adalah bilangan prima ganjil

FR = basis awal

a dan b = *field* yang mendefinisikan persamaan kurva

SEED = bit string (opsional)

G = titik pembangkitan bilangan prima pada kurva, x_G dan y_G.

n = orde titik G

h = kofaktor, nilainya sama dengan orde kurva dibagi n.

Berikut adalah tabel yang menunjukkan tingkat keamanan penggunaan kunci ECC

Bits of Security	*Bit length of subgroup order q*	*Bit length of field order p*
80	160	65536
112	224	65536
128	256	65536
192	384	65536
256	512	65536

Sama seperti pembangkitan parameter domain FCC, pada pembangkitan parameter domain ECC juga memberikan jaminan keamanan yang lebih untuk kunci ECC yang lebih panjang, tetapi membutuhkan waktu dan "ruang" untuk menggunakannya. Untuk mendapatkan perkiraan yang masuk akal dalam memecahkan kunci dibutuhkan secara berurutan 2^{80}, 2^{112}, 2^{128}, 2^{192}, 2^{256} operasi.

Dalam pengaturan parameter domain, setiap bagian dari kumpulan parameter domain harus dilindungi dari modifikasi dan pengubahan sampai dengan kumpulan parameter tersebut rusak atau sampai dengan kumpulan tersebut tidak digunakan lagi. Setiap pasangan kunci privat – kunci publik seharusnya memiliki kumpulan parameter domain yang spesifik seperti contohnya pada sertifikat kunci publik.

2.3 Proses Validasi Kunci Publik

2.3.1 Proses Validasi FFC *Full Public Key*

Validasi FFC *full public key* tidak memerlukan pengetahuan mengenai kunci privat yang berasosiasi sehingga hal ini dapat dilakukan kapan pun oleh semua orang. Metode ini sebaiknya digunakan dengan kunci publik FFC yang bersifat statis dan berlangsung sementara. [28]

Masukan:

1. (p, q, g [, *seed, pgenCounter*]: kumpulan parameter domain FCC yang valid.
2. y: salah satu calon FCC *public key*.

Proses:

1. Lakukan pengujian apakah

   $$2 \leq y \leq p-2$$

   Hal ini dilakukan untuk memastikan bahwa kunci memiliki representasi yang unik, benar dan berada dalam jangkauan dari *field*.

2. Lakukan pengujian apakah

   $$y^q = 1 \pmod p$$

   Hal ini dilakukan untuk memastikan bahwa kunci memiliki urutan yang benar dalam *subgroup*.

Keluaran:

Jika ada salah satu dari pengecekan di atas yang gagal, maka keluaran akan tidak valid. Sebaliknya, jika berhasil, maka keluarannya adalah validasi penuh berhasil.

2.3.2 Proses Validasi ECC *Full Public Key*

Proses ini mengecek semua properti aritmetika dari calon ECC *public key*. Validasi ECC *full public key* tidak memerlukan pengetahuan mengenai kunci privat yang berasosiasi sehingga hal ini dapat dilakukan kapan pun oleh semua orang. Metode ini sebaiknya digunakan dengan kunci publik ECC yang bersifat statis dan berlangsung sementara. [28]

Masukan:

1. (q, FR, a, b, [SEED,] G, n h): kumpulan parameter domain ECC yang valid.
2. $Q' = (x_{Q'}, y_{Q'})$: salah satu calon ECC *public key*.

Proses:

1. Lakukan pengujian apakah Q' bukan titik tak terhingga O. Hal ini untuk menguji sebagian dari kunci publik apakah berada dalam jangkauan yang tidak valid dalam grup EC.
2. Lakukan pengujian apakah $x_{Q'}$ dan $y_{Q'}$ adalah bilangan bulat dalam selang [0, p-1] dalam kasus q = p adalah sebuah bilangan prima ganjil, atau $x_{Q'}$ dan $y_{Q'}$ adalah bit string dengan panjang m bits dalam kasus $q = 2^m$.
3. Jika q=p adalah bilangan prima ganjil, lakukan pengujian apakah

   $$(y_{Q'})^2 \equiv (x_{Q'})^3 + ax_{Q'} + b \pmod p$$

   Jika $q=2^m$, lakukan pengujian apakah

   $$(y_{Q'})^2 + x_{Q'}y_{Q'} = (x_{Q'})^3 + a(x_{Q'})^2 + b$$

   berada dalam GF(2^m).

   Hal ini dilakukan untuk memastikan apakah kunci publik berada dalam urutan yang benar dalam grup EC.

4. Lakukan pengujian apakah

   $$nQ' = O$$

   Hal ini dilakukan untuk memastikan bahwa kunci publik memiliki urutan yang benar. Selama pengecekan 1, pastikan bahwa kunci publik berada dalam jangkauan yang benar dari EC *subgroup*.

Keluaran:

Jika ada salah satu dari pengecekan di atas yang gagal, maka keluaran akan tidak valid. Sebaliknya, jika berhasil, maka keluarannya adalah validasi penuh berhasil.

2.3.3 Proses Validasi ECC *Partial Public Key*

Proses ini mengecek beberapa tapi tidak seluruh properti aritmetika dari calon ECC *public key*. ECC *partial public key* mengabaikan validasi keanggotaan subgroup, dan biasanya proses yang dibutuhkan lebih cepat daripada validasi ECC *full public key*. Validasi ECC *full public key* tidak memerlukan pengetahuan mengenai kunci privat yang berasosiasi sehingga hal ini dapat dilakukan kapan pun oleh semua orang. Metode ini sebaiknya digunakan dengan kunci publik ECC yang bersifat statis dan berlangsung sementara. [28]

Masukan:

1. (q, FR, a, b, [SEED,] G, n h): kumpulan parameter domain ECC yang valid.
2. Q' = ($x_{Q'}$, $y_{Q'}$): salah satu calon ECC *public key*.

Proses:

1. Lakukan pengujian apakah Q' bukan titik tak terhingga O. Hal ini untuk menguji sebagian dari kunci publik apakah berada dalam jangkauan yang tidak valid dalam grup EC.
2. Lakukan pengujian apakah $x_{Q'}$ dan $y_{Q'}$ adalah bilangan bulat dalam selang [0, p-1] dalam kasus q = p adalah sebuah bilangan prima ganjil, atau $x_{Q'}$ dan $y_{Q'}$ adalah bit string dengan panjang m bits dalam kasus $q = 2^m$.
3. Jika q=p adalah bilangan prima ganjil, lakukan pengujian apakah

   $$(y_{Q'})^2 \equiv (x_{Q'})^3 + ax_{Q'} + b \pmod p$$

   Jika $q=2^m$, lakukan pengujian apakah

   $$(y_{Q'})^2 + x_{Q'}y_{Q'} = (x_{Q'})^3 + a(x_{Q'})^2 + b$$

   berada dalam GF(2^m).

   Hal ini dilakukan untuk memastikan apakah kunci publik berada dalam urutan yang benar dalam grup EC.

   Catatan: Karena urutan kunci publik tidak diverifikasi, maka tidak ada pengecekan apakah kunci publik berada dalam EC *subgroup*.

Keluaran:

Jika ada salah satu dari pengecekan di atas yang gagal, maka keluaran akan tidak valid. Sebaliknya, jika berhasil, maka keluarannya adalah validasi sebagian berhasil.

2.4 Primitif *Discrete Logarithmic Cryptography*

Primitif untuk melakukan perhitungan pada penyebaran kunci didefinisikan pada ANSI X9.42 dan X9.63. Primitif adalah sebuah proses sederhana yang didefinisikan untuk memfasilitasi implementasi pada subrutin perangkat keras atau pada subrutin perangkat lunak. Setiap skema pertukaran kunci membutuhkan satu primitif. Berikut adalah beberapa skema pertukaran kunci yang selanjutnya akan dibahas:

1. Primitif FFC Diffie-Hellman: Primitif ini sebaiknya digunakan untuk skema *dhHybrid1, dhEphem, dhHybridOneFlow, dhOneFlow* dan *dhStatic* yang berdasarkan *finite field cryptography* dan algoritma Diffie-Hellman.
2. Primitif ECC Cofactor Diffie-Hellman: merupakan modifikasi dari primitif Diffie-Hellman. Primitif ini sebaiknya digunakan untuk skema *Full Unified Model, Ephemeral Unified Model, One-Pass Unified Model, One-Pass Diffie-Hellman*, dan *Static Unified Model* yang berdasarkan *elliptic curve cryptography* dan algoritma Diffie-Hellman.
3. Primitif FFC MQV: Primitif ini sebaiknya digunakan untuk skema MQV2 dan MQV1 yang berdasarkan *finite field cryptography* dan algoritma MQV.
4. Primitif ECC MQV: Primitif ini sebaiknya digunakan untuk skema *Full MQV* dan *One-Pass MQV* yang berdasarkan *elliptic curve cryptography* dan algoritma MQV.

2.4.1 Primitif Diffie-Hellman

2.4.1.1 Primitif *Finite Field Cryptography* (FFC) Diffie-Hellman

Rahasia Z dihitung dengan menggunakan parameter domain (p, q, g [, SEED, *pgenCounter*]), kunci publik kelompok dan kunci privat. Primitif ini digunakan pada skema *dhHybrid1, dhEphem, dhHybridOneFlow, dhOneFlow* dan *dhStatic*. Asumsikan bahwa kelompok A adalah kelompok yang melakukan perhitungan, dan kelompok B adalah kelompok lainnya yang terkait pada pertukaran kunci. Catatan: kelompok A dapat berperan sebagai *initiator* U atau *responder* V.

Masukan:

1. (p, q, g [, SEED, *pgenCounter*]): parameter domain.
2. x_A: kunci privat.
3. y_B: kunci publik kelompok lain (kelompok B).

Proses:

1. $Z = y_B^{x_A} \bmod p$
2. Jika Z = 1, keluaran = "Gagal".
3. Jika tidak, keluaran = Z.

Keluaran:

Z atau "Gagal".

2.4.1.2 Primitif *Elliptic Curve Cryptography* (ECC) Cofactor Diffie-Hellman (CDH)

Rahasia Z dihitung dengan menggunakan parameter domain (q, FR, a, b, [SEED,] G, n, h), kunci publik kelompok, dan kunci privat. Primitif ini digunakan pada skema *Full Unified Model, Ephemeral Unified Model, One-Pass Unified Model, One-Pass Diffie-Hellman*, dan *Static Unified Model*. Asumsikan bahwa kelompok A adalah kelompok yang melakukan perhitungan, dan kelompok B adalah kelompok lainnya yang terkait pada pertukaran kunci. Catatan: kelompok A dapat berperan sebagai *initiator* U atau *responder* V.

Masukan:

1. (q, FR, a, b, [SEED,] G, n, h): parameter domain.
2. d_A: kunci privat.
3. Q_B: kunci publik kelompok lain (kelompok B).

Proses:

1. Lakukan perhitungan titik $P = hd_AQ_B$.
2. Jika P = O, keluaran = "Gagal".
3. $Z = x_P$ di mana x_P adalah koordinat x dari titik P.

Keluaran:

Z atau "Gagal".

2.4.2 Primitif Menezes-Qu-Vanstone (MQV)

2.4.2.1 Primitif *Finite Field Cryptography* (FFC) MQV

Rahasia Z dihitung dengan menggunakan parameter domain (p, q, g [, SEED, *pgenCounter*]), kunci publik kelompok dan kunci privat dan kunci publik kelompok yang memilikinya. Asumsikan bahwa kelompok A adalah kelompok yang melakukan perhitungan, dan kelompok B adalah kelompok lainnya yang terkait pada pertukaran kunci. Catatan: kelompok A dapat berperan sebagai *initiator* U atau *responder* V.

Masukan:

1. (p, q, g [, SEED, *pgenCounter*]): parameter domain.
2. x_A: kunci privat statis.
3. y_B: kunci publik statis kelompok lainnya (kelompok B).
4. r_A: kunci privat kedua.
5. t_A: kunci publik kedua.
6. t_B: kunci publik kedua milik kelompok lain.

Proses:

1. $w = \|q\| / 2$.
2. $T_A = (t_A \bmod 2^w) + 2^w$.
3. $S_A = (r_A + T_A x_A) \bmod q$.
4. $T_B = (t_B \bmod 2^w) + 2^w$.
5. $Z = \left(\left(t_B \left(y_B^{T_B}\right)\right)^{S_A}\right) \bmod p.$
6. Jika Z = 1, keluaran = "Gagal".
7. Jika tidak, keluaran = Z.

Keluaran:

Z atau "Gagal".

2.4.2.1.1 *Finite Field Cryptography* MQV2

Skema MQV2 adalah salah satu skema yang menggunakan primitif FFC MQV. Pada skema ini, setiap kelompok memiliki pasangan kunci statis dan pasangan kunci sementara. Asumsikan bahwa kelompok A adalah kelompok yang melakukan perhitungan, dan kelompok B adalah kelompok lainnya yang terkait pada pertukaran kunci. Catatan: kelompok A dapat berperan sebagai *initiator* U atau *responder* V.

Pada skema ini, masukan keempat dan kelima (kunci privat dan kunci publik kedua) pada masukan FFC MQV menjadi pasangan kunci sementara, kunci privat dan kunci publik

sementara (r_A dan t_A), sedangkan masukan keenam (kunci publik kedua milik kelompok lain) menjadi kunci publik sementara kelompok lain (t_B).

2.4.2.1.2 *Finite Field Cryptography* MQV1

Skema MQV1 adalah salah satu skema yang menggunakan primitif FFC MQV. Pada skema ini, *initiator* memiliki pasangan kunci statis dan sebuah pasangan kunci sementara, sedangkan *responder* hanya memiliki pasangan kunci statis. *One-Pass* MQV (bentuk *store and forward*) menggunakan primitif MQV dengan menggunakan pasangan kunci statis *responder* sebagai pasangan kunci kedua *responder* (*responder* tidak memiliki pasangan kunci sementara).

Initiator menggunakan kunci publik statis *responder* sebagai kunci publik kedua. Sebagai contoh, masukan keenam pada algoritma FFC MQV menjadi kunci publik statis kelompok lain (y_A).

Responder menggunakan kunci privat statis miliknya sebagai kunci privat kedua. Sebagai contoh, masukan keempat menjadi kunci privat statis *responder* x_A dan masukan kelima menjadi kunci publik statis *responder* (y_A).

2.4.2.2 Primitif *Elliptic Curve Cryptography* (ECC) MQV

Rahasia Z dihitung dengan menggunakan parameter domain (q, FR, a, b, [SEED,] G, n, h), kunci publik kelompok, dan kunci privat dan kunci publik kelompok yang memilikinya. Versi ECC MQV menggunakan kofaktor h dalam perhitungannya. Asumsikan bahwa kelompok A adalah kelompok yang melakukan perhitungan, dan kelompok B adalah kelompok lainnya yang terkait pada pertukaran kunci. Catatan: kelompok A dapat berperan sebagai *initiator* U atau *responder* V.

Masukan:

1. (q, FR, a, b, [SEED,] G, n, h): parameter domain.
2. $d_{S, A}$: kunci privat statis.
3. $Q_{S, B}$: kunci publik statis kelompok lain (kelompok B).
4. $d_{e, A}$: kunci privat kedua.
5. $Q_{e, A}$: kunci publik kedua.
6. $Q_{e, B}$: kunci publik kedua kelompok lain (kelompok B).

Proses:

1. $\text{implicitsig}_A = (d_{e, A} + \text{avf}(Q_{e, A})\ d_{S, A}) \bmod n$.
2. $P = h(\text{implicitsig}_A)(Q_{e, B} + \text{avf}(Q_{e, B})\ Q_{S, B})$.
3. Jika P = O, keluaran = “Gagal”.
4. $Z = x_P$, di mana x_P adalah koordinat x dari P.

Keluaran:

Z atau “Gagal”.

2.4.2.2.1 *Elliptic Curve Cryptography Full* MQV

Skema ECC *Full* MQV adalah salah satu skema yang menggunakan primitif ECC MQV. Pada skema ini, setiap kelompok memiliki pasangan kunci statis dan pasangan kunci sementara. Asumsikan bahwa kelompok A adalah kelompok yang melakukan perhitungan, dan kelompok B adalah kelompok lainnya yang terkait pada pertukaran kunci. Catatan: kelompok A dapat berperan sebagai *initiator* U atau *responder* V.

Pada skema ini, pasangan kunci kedua milik kelompok yang melakukan perhitungan menjadi pasangan kunci sementara kelompok yang melakukan perhitungan dan pasangan kunci kedua milik kelompok lain menjadi pasangan kunci sementara kelompok lain.

2.4.2.2.2 *Elliptic Curve Cryptography One-Pass* MQV

Skema ECC *One-Pass* MQV adalah salah satu skema yang menggunakan primitif ECC MQV. Pada skema ini, *initiator* memiliki pasangan kunci statis dan pasangan kunci sementara, sedangkan responder hanya memiliki pasangan kunci statis. setiap kelompok memiliki pasangan kunci statis dan pasangan kunci sementara. *One-Pass* MQV (bentuk *store and forward*) menggunakan primitif MQV dengan menggunakan pasangan kunci statis *responder* sebagai pasangan kunci kedua *responder* (*responder* tidak memiliki kunci sementara)

Initiator menggunakan kunci publik statis *responder* sebagai kunci publik kedua *responder*. Sebagai contoh, masukan keenam pada algoritma ECC MQV menjadi kunci publik statis kelompok lain ($Q_{S, B}$).

Responder menggunakan kunci privat statis miliknya sebagai kunci privat keduanya. Sebagai contoh, masukan keempat menjadi kunci privat statis *responder* $d_{S, A}$ dan masukan kelima menjadi kunci publik statis *responder* ($Q_{S, A}$).

3. MQV

MQV (Menezes-Qu-Vanstone) adalah sebuah algoritma untuk konfirmasi pertukaran kunci berdasarkan skema Diffie-Hellman. Tidak seperti Diffie-Hellman, MQV menyediakan perlindungan terhadap serangan yang aktif. Secara umum, protokol ini dapat diubah untuk dapat bekerja pada sebuah kelompok terbatas yang berubah-ubah dan secara khusus untuk kelompok *elliptic curve* atau lebih dikenal dengan ECMQV.

Terdapat dua jenis protokol pertukaran kunci secara umum, yaitu *authenticated key agreement (AK) protocol* dan *authenticated key agreement with key confirmation (AKC) protocol*. Protokol pertukaran kunci yang menyediakan fitur autentikasi kunci secara implisit kepada kedua belah entitas yang terkait disebut sebagai *authenticated key agreement (AK) protocol*, sedangkan protokol pertukaran kunci yang menyediakan fitur autentikasi kunci secara eksplisit kepada kedua belah entitas yang terkait disebut sebagai *authenticated key agreement with key confirmation (AKC) protocol.*

Terdapat beberapa atribut keamanan yang sebaiknya terdapat pada dua jenis algoritma pertukaran kunci di atas, yaitu

1. *known-key security*.
   Setiap protokol pertukaran kunci yang digunakan antara entitas A dan B harus menghasilkan kunci yang unik dan rahasia, seperti kunci sesi. Protokol juga harus dapat tetap memiliki tujuannya dalam menghadapi lawan yang telah mempelajari beberapa kunci sesi.

2. *(perfect) forward secrecy*.
   Jika kunci privat jangka panjang dari satu atau lebih entitas membahayakan, maka kerahasiaan dari kunci sesi sebelumnya yang dibangkitkan oleh entitas yang baik tidak berpengaruh.

3. *key-compromise impersonation.*
   Andaikan kunci privat jangka panjang A telah diketahui. Dalam keadaan ini, lawan dapat menirukan A, karena nilai kunci privat A digunakan untuk mengidentifikasi A. Sebuah protokol diharapkan dapat menangani kelemahan ini dan entitas selain A tidak dapat menirukan A.

4. *unknown key-share.*
   Entitas A tidak dapat dipaksa untuk melakukan pertukaran kunci dengan entitas B tanpa sepengetahuan A. Sebagai contoh, A mempercayai bahwa kunci diberikan kepada sebuah entitas C bukan B dan B yang sebenarnya percaya bahwa ia sedang bertukar kunci dengan A.

5. *key-control.*
   Tidak ada entitas baik A maupun B yang mampu memaksa kunci sesi untuk memiliki nilai yang telah dipilih terlebih dahulu.

6. *identity assurance*
   Setiap kelompok memiliki jaminan terhadap identitasnya.

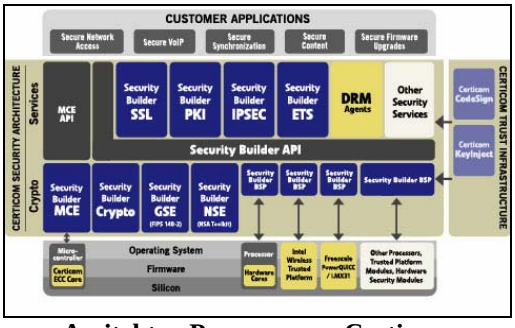
MQV pertama kali diusulkan oleh Menezes, Qu dan Vanstone pada tahun 1995, lalu diperbaiki oleh Law dan Solinas pada tahun 1998. MQV ini dimasukkan ke dalam standar kunci-publik IEEE P1363. MQV dipatenkan oleh Certicom.

IEEE P1363 adalah sebuah standar dari *Institute of Electrical and Electronic Engineers* (IEEE) untuk kriptografi kunci-publik. Standar ini mencakup spesifikasi sebagai berikut:

a. Kriptografi kunci publik tradisional (P1363-2000 dan P1363A-2004) termasuk tanda tangan digital dan pengadaan kunci menggunakan beberapa pendekatan matematis, seperti:
   1. Pemfaktoran bilangan (RSA).
   2. Logaritma Diskrit (Diffie-Hellman, DSA).
   3. *Elliptic Curve Discrete Logarithm* (MQV).
b. Pola geometris berdasarkan enkripsi dan tanda tangan digital (NTRUEncrypt dan NTRUSign) kriptografi kunci-publik (P1363.1).

c. Sandi lewat berdasarkan kriptografi kunci-publik (P1363.2).
   1. Pertukaran kunci dengan otentikasi sandi lewat (contoh: EKE, SPEKE, SRP).
   2. Skema untuk mendapatkan kunci dengan otentikasi sandi lewat (contoh: Ford dan Kaliski).
d. Kriptografi kunci-publik berdasarkan identitas dan pasangan-pasangan (P1363.3) (proyek disetujui bulan September 2005).

Certicom *Corporation* adalah sebuah sebuah perusahaan di bidang kriptografi yang ditemukan oleh Dr. Scott Vanstone pada tahun 1985. Arsitektur pengamanan Certicom terdiri dari kumpulan penyedia perangkat lunak kriptografi yang modular dan *portable*. Penyedia perangkat lunak yang terdapat pada arsitektur ini menyediakan validasi FIPS 140-2 dan memenuhi pedoman *National Security Agency* (NSA) untuk ECC dan layanan pengamanan seperti SSL, IPSec, PKI, DRM, dan *Embedded Trust Services* (ETS). *Board Support Packages* (BSP) memungkinkan fungsional kriptografi diterapkan pada perangkat keras. Solusi yang komprehensif ini didesain secara spesifik untuk memenuhi kebutuhan pasar seperti komunikasi pemerintahan, mobilitas, DRM dan akses kondisional, dan perangkat lunak *enterprise*. Gambar berikut menunjukkan arsitektur pengamanan Certicom.



Arsitektur Pengamanan Certicom

NSA mengadopsi 26 paten ECC yang dikeluarkan Certicom untuk komunikasi pemerintah yang rahasia dan komunikasi yang tidak rahasia tapi sensitif. Pelanggan Certicom meliputi *General Dynamics*, *Motorola*, *Oracle*, *Research In Motion* dan *Unisys*.

Seperti telah dijelaskan sebelumnya, bahwa algoritma MQV merupakan variasi dari algoritma Diffie-Hellman dan terdapat 2 bentuk MQV, yaitu:

1. Interaktif (MQV2) – simetris
2. *Store and Forward* (MQV1) – asimetris

MQV2

Berikut adalah skema pertukaran kunci MQV2:

(Asumsi: Kelompok U adalah kelompok *initiator* dan kelompok V adalah kelompok *responder*)

	Kelompok U	Kelompok V
Data statis	1. Parameter domain (p,q, g) 2. Kunci privat statis, x_U 3. Kunci publik statis, y_U	1. Parameter domain (p,q, g) 2. Kunci privat statis, x_V 3. Kunci publik statis, y_V
Data sementara	1. Parameter domain (p,q, g) 2. Kunci privat sementara, r_U 3. Kunci sementara statis, t_U	1. Parameter domain (p,q, g) 2. Kunci privat sementara, r_V 3. Kunci publik sementara, t_V
Masukan	(p, q, g), x_U, y_V, r_U, t_U, t_V	(p, q, g), x_V, y_U, r_V, t_V, t_U
	$w = \|\|q\|\| / 2$	$w = \|\|q\|\| / 2$
	$t_U' = (t_U \bmod 2^w) + 2^w$	$t_V' = (t_V \bmod 2^w) + 2^w$
	$S_U = (r_U + t_U' x_U) \bmod q$	$S_V = (r_V + t_V' x_V) \bmod q$
	$t_V' = (t_V \bmod 2^w) + 2^w$	$t_U' = (t_U \bmod 2^w) + 2^w$
	$Z_{MQV} = ((t_V (y_V \wedge t_V')) \wedge S_U) \bmod p$	$Z_{MQV} = ((t_U (y_U \wedge t_U')) \wedge S_V) \bmod p$
Keluaran	$ZZ = oct(Z_{MQV})$	$ZZ = oct(Z_{MQV})$
Penurunan Kunci	Menggunakan fungsi untuk menurunkan kunci menggunakan ZZ, panjang kunci dan informasi lainnya	Menggunakan fungsi untuk menurunkan kunci menggunakan ZZ, panjang kunci dan informasi lainnya

MQV1

Kedua kelompok memberikan jumlah informasi yang berbeda:

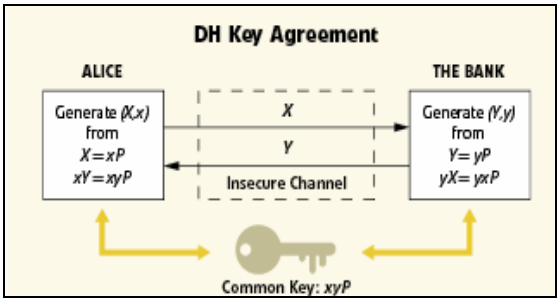
1. Initiator (U) memberikan dua pasang kunci – statis dan sementara.
2. Responder (V) memberikan sebuah pasang kunci – statis.

Berikut adalah skema pertukaran kunci MQV1:

	Kelompok U	Kelompok V
Data statis	1. Parameter domain (p,q, g) 2. Kunci privat statis, x_U 3. Kunci publik statis, y_U	1. Parameter domain (p,q, g) 2. Kunci privat statis, x_V 3. Kunci publik statis, y_V
Data sementara	1. Parameter domain (p,q, g) 2. Kunci privat sementara, r_U 3. Kunci sementara statis, t_U	N/A
Masukan	(p, q, g), x_U, y_V, r_U, t_U, w	(p, q, g), x_V, y_U, t_U, w
	$t_U' = (t_U \bmod 2^w) + 2^w$	$y_V' = (y_V \bmod 2^w) + 2^w$
	$S_U = (r_U + t_U' x_U) \bmod q$	$S_V = (r_V + y_V' x_V) \bmod q$
	$y_V' = (y_V \bmod 2^w) + 2^w$	$t_U' = (t_U \bmod 2^w) + 2^w$
	$Z_{MQV} = (y_V (y_V \wedge y_V')) \wedge S_U \bmod p$	$Z_{MQV} = ((t_U (y_U \wedge t_U')) \wedge S_V) \bmod p$
Keluaran	$ZZ = oct(Z_{MQV})$	$ZZ = oct(Z_{MQV})$
Penurunan Kunci	Menggunakan fungsi untuk menurunkan kunci menggunakan ZZ, panjang kunci dan informasi lainnya	Menggunakan fungsi untuk menurunkan kunci menggunakan ZZ, panjang kunci dan informasi lainnya

Perbandingan Diffie-Hellman dan MQV

Diffie-Hellman



Skema Pertukaran Kunci Diffie-Hellman

Keterangan:
Alice dan Bank menyepakati untuk menggunakan Diffie-Hellman.

Alice membangkitkan kunci sesi (X, x) dari hasil pembangkitan bilangan acak dengan x sebagai umpannya, lalu menghitung X = xP di mana x adalah sebuah bilangan dan X adalah titik pada kurva *elliptic*.

Bank membangkitkan kunci sesi (Y, y) dari hasil pembangkitan bilangan acak dengan y sebagai umpannya, lalu menghitung Y = yP.

P adalah titik pembangkitan pada kurva *elliptic*.

Kemudian Alice mengirimkan X kepada Bank dan Bank mengirimkan Y kepada Alice. Alice menerima Y dari Bank, lalu menghitung xY = xyP. Bank menerima X dari Alice, lalu menghitung yX = yxP.

Sekarang mereka telah memiliki kunci rahasia xyP.

Pada algoritma pertukaran kunci Diffie-Hellman ini, masih terdapat kelemahan, yaitu Alice dan Bank tidak memiliki sesi yang benar aman, karena kunci publik dan kunci privat dibangkitkan *on the fly*, tidak ada autentikasi terhadap pengguna yang menjamin tidak adanya penyamaran entitas. MQV menjawab permasalahan ini.

MQV

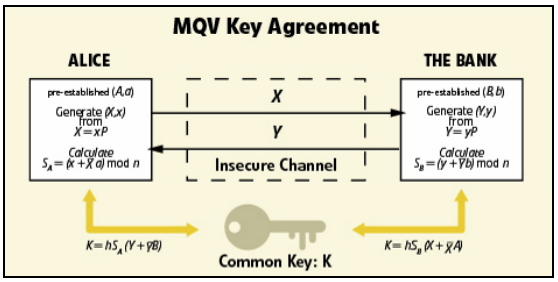
Dibandingkan dengan algoritma pertukaran kunci Diffie-Hellman, MQV mampu mengatasi *unknown key-share attack* yang merupakan salah satu atribut yang diharapkan dimiliki pada algoritma pertukaran kunci yang telah dijelaskan sebelumnya. Selain itu, MQV juga memiliki atribut *key-compromise impersonation*. Kedua atribut ini tidak ditemukan pada algoritma pertukaran kunci Diffie-Hellman. Dengan demikian, protokol kriptografi yang menggunakan algoritma pertukaran kunci MQV dapat memiliki proses autentikasi yang lebih kuat dan memastikan entitas lawan tidak mampu untuk menyamar sebagai kelompok ketiga dari entitas-entitas yang melakukan pertukaran kunci.

MQV juga memiliki atribut lainnya yang memberikan nilai tambah pada protokol ini, antara lain yaitu biaya komunikasi yang rendah, *role-symmetric*, tidak interaktif dan tidak menggunakan enkripsi atau *time-stamping*.

Sebagai hasilnya, MQV memiliki semua atribut keamanan yang diinginkan dalam pertukaran kunci. Selain itu, jika perhitungan dilakukan dengan menggunakan *elliptic curve*

cryptography, MQV menawarkan performansi yang menguntungkan yang signifikan dibandingkan skema pertukaran kunci yang lain. Hal ini menyebabkan pengembangan protokol keamanan dan sistem yang membutuhkan skema pertukaran kunci yang efisien dan terautentikasi menjadi ideal apabila menggunakan skema ini.

Berikut adalah gambar yang menunjukkan skema pertukaran kunci MQV secara umum.



Skema Pertukaran Kunci MQV

Keterangan:

Alice menghitung S (disebut sebagai *implicit signature*):

$$S_A = (x + Xa) \bmod n$$

, di mana n adalah orde dari pembangkitan titik P.

Bank menghitung S_B dengan cara yang sama:

$$S_B = (y + \overline{Y}\, b) \bmod n$$

Alice dan Bank menghitung K:

$$K = hS_A(Y + \overline{Y}\, B)$$

$$= hS_B(X + \overline{X}\, A)$$

, di mana h adalah kofaktor yang didefinisikan pada P1363.

$\overline{X}$ (atau $\overline{Y}$) merepresentasikan L bit pertama dari komponen pertama dari titik X (atau Y), di mana $L = \left\lceil \frac{([\log_2 n] + 1)}{2} \right\rceil$

Alice memiliki pasangan kunci (A, a), di mana A adalah kunci publiknya dan a adalah kunci privatnya.

Bank juga memiliki pasangan kunci (B, b), di mana B adalah kunci publik bank dan b adalah kunci privat bank.

Alice membangkitkan kunci sesi (X, x) dari hasil pembangkitan bilangan acak dengan x sebagai umpannya, lalu menghitung X = xP di mana x adalah sebuah bilangan dan X adalah titik pada kurva *elliptic*.

Bank membangkitkan kunci sesi (Y, y) dari hasil pembangkitan bilangan acak dengan y sebagai umpannya, lalu menghitung Y = yP.

P adalah titik pembangkitan pada kurva *elliptic*.

Kemudian Alice mengirimkan X kepada Bank dan Bank mengirimkan Y kepada Alice. Asumsi: Alice telah memiliki kunci publik Bank dan Bank telah memiliki kunci publik Alice. Kunci publik telah diterima dengan cara yang terjamin.

4. Kesimpulan

MQV memiliki semua atribut keamanan untuk algoritma pertukaran kunci. Protokol kriptografi yang menggunakan algoritma MQV sebagai algoritma pertukaran kuncinya akan memberikan autentikasi yang kuat dan MQV dapat menjamin bahwa tidak ada entitas lawan yang dapat menyamar sebagai kelompok ketiga dari entitas-entitas yang melakukan pertukaran kunci.

MQV terbagi menjadi dua bentuk, interaktif (MQV2) dan *store and forward* (MQV1). MQV2 melibatkan dua kelompok (*initiator* [U] dan *responder* [V]), di mana kedua kelompok tersebut memiliki pasangan kunci yang sama, yaitu pasangan kunci statis dan pasangan kunci sementara.

MQV1 melibatkan satu kelompok (*initiator* [U]), di mana kelompok ini memiliki kedua pasangan kunci, yaitu pasangan kunci statis dan pasangan kunci sementara. Sedangkan kelompok *responder* [V] hanya memiliki pasangan kunci pasangan kunci statis saja.

Pada primitif MQV, *initiator* [U] dan *responder* [V] hanya memiliki pasangan kunci statis.

Skema Diffie-Hellman yang sederhana, memiliki beberapa kekurangan yang utama dalam hal kemananan. MQV yang merupakan pengembangan dari skema Diffie-Hellman mampu mengatasi cacat yang terdapat pada skema Diffie-Hellman, tapi MQV dapat membuat protokol yang menggunakan skema ini menjadi tidak efisien. Jika protokol keamanan yang diinginkan adalah sebuah protokol yang memiliki performansi yang signifikan, maka MQV masih dapat digunakan.

Berikut adalah beberapa kelemahan lain yang dimiliki primitif MQV.

1. *Group representation attack*

   Keamanan MQV sangat rentan dalam hal representasi elemen-elemen grup dalam protokol. MQV membutuhkan representasi grup yang memiliki batasan tertentu untuk membuktikan bahwa MQV aman.

2. *Unknown Key-Share* (UKS) *attack*

   Meskipun MQV diakui dapat mengatasi hal ini, namun Kaliski menunjukkan bahwa MQV dengan bukti yang dimilikinya masih gagal dalam menghadapi serangan ini. Dan meskipun dengan menambahkan *key confirmation* yang berarti menambahkan pesan ketiga dalam protokol ini, MQV masih tetap dikatakan gagal dalam menghadapi serangan ini. Permasalahannya adalah MQV gagal apabila penyerang dapat mempelajari status sesi sementara.

3. *Lack of Perfect Forward Secrecy* (PFS)

   MQV tidak menyediakan atribut *Perfect Forward Secrecy*. Hal ini adalah turunan dari keterbatasan autentikasi secara implisit yang dimiliki protokol 2-pesan yang berdasarkan autentikasi kunci publik. Sebenarnya tidak ada protokol yang dapat menyediakan *Perfect Forward Secrecy*.

4. *Key-compromise impersonation* (KCI) *attack*

   Karena MQV rentan terhadap serangan autentikasi yang utama menyebabkan ketahanan MQV tidak dapat dipenuhi.

5. *Prime-order checks*

   MQV membutuhkan pengecekan orde prima oleh setiap *peer*-nya. Pengecekan ini menambah biaya secara eksponensial untuk setiap peer dan mempengaruhi performansi MQV secara signifikan.

Skema ECMQV dengan perhitungan *elliptic curve cryptography* akan memberikan performansi yang menguntungkan dan dapat menjadikan ECMQV sebagai algoritma pertukaran kunci yang lebih baik dibandingkan algoritma pertukaran kunci yang lain yang ada saat ini. Alasan utama yang menyebabkan ECMQV menjadi algoritma pertukaran kunci yang baik adalah karena ukuran kunci yang digunakan lebih kecil dan perhitungan yang dilakukan lebih cepat.

Hal-hal di atas membuat MQV sangat cocok untuk lingkungan yang penuh dengan batasan seperti kartu cerdas, mobile devices dan RFIDs.

Namun demikian, secara umum, dalam menentukan skema mana yang akan dipilih, maka perlu dipertimbangkan pula atribut mana yang penting dalam tingkat keamanan pada sistem yang akan menggunakan protokol atau skema pertukaran kunci tersebut.

Dalam memilih skema mana yang akan diimplementasikan, pertimbangan mengenai skema pertukaran kunci yang akan dipakai menjadi salah satu langkah yang harus dilalui dalam mengimplementasikan protokol keamanan dalam sebuah sistem. Selain itu, perlu juga diperhatikan pembobotan atribut keamanan yang dibutuhkan oleh sistem.

Setelah itu, langkah selanjutnya adalah pemilihan primitif. Saat ini, dua primitif untuk pertukaran kunci yang dapat menjadi bahan pertimbangan adalah algoritma Diffie-Hellman atau MQV. Kemudian, metode perhitungannya juga menjadi salah satu faktor yang mempengaruhi performansi dari sebuah skema. Metode perhitungan yang dapat digunakan adalah *Discrete Logarithmic Cryptography*

(DLC) atau *Elliptic Curve Cryptography* (ECC).

Setiap langkah yang dipilih harus dipertimbangkan berdasarkan atribut keamanan yang berbeda-beda di setiap sistem keamanan dan bagaimana performansi yang diinginkan dari atribut tersebut. Secara umum, menggunakan ECC dan MQV adalah pilihan yang terbaik untuk mendapatkan hasil yang terbaik dalam melakukan *trade-off* performansi keamanan.

Tabel pada halaman selanjutnya (Tabel Acuan Pemilihan Primitif) dapat dijadikan sebagai acuan dalam memilih primitif yang akan digunakan.

Untuk mengatasi kelemahan yang dimiliki MQV, terdapat sebuah pengembangan dari MQV yang disebut HMQV (Hash MQV).

Perhitungan kunci sesi pada HMQV berbeda dengan MQV, di mana nilai kunci statis sementara dan kunci publik sementara melibatkan *hashing* terhadap nilai Diffie-Hellman (X dan Y) dan identitas dari setiap *peer*.

Dari deskripsi ini dapat dilihat bahwa HMQV mempertahankan performansi MQV yang menonjol dan dalam waktu yang sama, HMQV mengatasi semua atribut keamanan yang menjadi kelemahan MQV.

HMQV dibuktikan dalam model acak oracle, dan dalam asumsi *Computational Diffie-Hellman* bahwa protokol ini aman dalam model keamanan Canetti-Krawczyk. Protokol HMQV ini tahan terhadap serangan penyamaran, *known-key attack*, dan *unknown key-share attack*.

Untuk aplikasi yang membutuhkan *full Perfect Forward Secrecy* (PFS), HMQV menyediakan varian dari HMQV dengan tiga pesan yang menambahkan pesan ketiga dan sebuah perhitungan MAC. Varian ini menjamin aplikasi memiliki *full PFS*. Varian ini disebut HMQV-C. Protokol ini menyediakan *key confirmation* untuk kedua kelompok yang terkait.

Setelah melihat protokol HMQV, ternyata beberapa kelemahan yang disebutkan pada artikel *HMQV: A High-Performance Secure Diffie-Hellman Protocol* [9], dibantah pada artikel *Another Look At HMQV* [10]. Pada artikel [10] dikatakan bahwa tidak seluruh kelemahan tersebut adalah benar. Berikut adalah tiga serangan yang tidak dapat dinyatakan sebagai serangan pada MQV.

1. Ketahanan terhadap *basic impersonation attack*.

   Beberapa implementasi dari MQV mungkin lemah karena kurangnya pilihan representasi. Sebagai contoh, apabila representasi yang dipilih adalah representasi di mana *l least-significant* bit dari semua koordinat x dari *elliptic curve* adalah konstan, maka MQV akan gagal menghadapi *impersonation attack*. Penelitian ini mengilustrasikan MQV tidak dapat dinyatakan aman terhadap kelompok *generic prime-order*, tapi hal ini sulit untuk dikatakan sebagai sebuah serangan.

2. *Prime-order testing* dan performansi eksponensial.

   Hal ini bukanlah serangan, melainkan sebuah peninjauan bahwa validasi dari kunci publik sementara membutuhkan biaya mahal yang eksponensial.

3. Validasi dari kunci publik jangka panjang

   Hal ini juga bukanlah sebuah serangan, melainkan sebuah peninjauan bahwa validasi kunci publik statis membutuhkan tambahan pada *certification authorities*.

Dari tujuh kelemahan dan serangan terhadap protokol MQV yang dinyatakan dalam [9], dapat disimpulkan bahwa tiga serangan pertama adalah bukan serangan dan apa yang diimplementasikan pada protokol HMQV adalah tidak benar dan kesalahan ini menuju kepada cacat yang fatal dari protokol HMQV.

Tiga serangan berikutnya adalah sebagai berikut.

1. Ketahanan terhadap *Key-Compromise Impersonation Attacks*

   Baik MQV maupun HMQV gagal untuk menghadapi serangan ini. Penelitian ini dapat digunakan untuk membenarkan bahwa kunci sesi sebaiknya diturunkan dengan melakukan perhitungan hash dari kunci rahasia yang telah dibangkitkan.

2. Ketahanan terhadap penyingkapan Diffie-Hellman eksponen.

   Jika lawan dapat mengetahui salah satu kunci privat sementara (x atau y), maka baik MQV maupun HMQV-1 juga akan gagal terhadap serangan ini.

3. Ketahanan terhadap *unknown key-share attacks*

   Sebagai bahan acuan, telah dikatakan sebelumnya bahwa pada protokol MQV perlu dimasukkan identitas dari kelompok dalam fungsi penurunan kunci.

Tiga serangan ini bergantung pada kemampuan lawan dalam mempelajari informasi rahasia.

Serangan yang terakhir adalah *perfect forward secrecy*. Hal ini telah diteliti pada artikel [9] bahwa tidak ada protokol pertukaran kunci *two-pass* yang dapat mencapai "*full*" *forward secrecy*. Serangan ini merupakan sebuah penelitian yang umum bahwa hanya bentuk yang terbatas dari *perfect forward secrecy* yang dapat dicapai oleh *two-pass protocol*.

CATEGORY	CASES	SECURITY ATTRIBUTES
Two Party Participation	U and V generate an ephemeral key pair and have a static key pair	• Known-key security • Forward secrecy • Key-compromise impersonation resilience (MQV primitive only) • Unknown key-share resilience (MQV primitive only) • Key control • Identity assurance
	U and V generate an ephemeral key pair; no static keys are used	• Known-key security • Forward secrecy • Key control
One Party Participation	Initiator U has a static key pair and generates an ephemeral key pair; Responder V has only a static key pair	• Known-key security • Forward secrecy • Key-compromise impersonation resilience (MQV primitive only) • Unknown key-share resilience (MQV primitive only) • Key control • Identity assurance
	Initiator U generates an ephemeral key pair; the Responder V has only a static key pair	• Known-key security • Forward secrecy for initiator only • Key control • Identity assurance for responder only
Static Keys Only	U and V have a static key pair	• Key-compromise impersonation resilience (MQV primitive only) • Unknown key-share resilience (MQV primitive only) • Identity assurance • Known-key security and forward secrecy are possible if variability is introduced in the shared secret

Tabel Acuan Pemilihan Primitif

DAFTAR PUSTAKA

[1] http://en.wikipedia.org/wiki/MQV. Tanggal akses: 16 Oktober 2006 pukul 09:25.

[2] Kaliski, Burt. (2000). *Unknown Key Share Attacks and the MQV Protocol*. RSA Euro 2000. http://www.rsasecurity.com/rsalabs/staff/bios/bkaliski/publications/mqv-uks/kaliski-mqv-uks-rsa-2000e.ppt. Tanggal akses: 16 Oktober 2006 pukul 09:20

[3] http://www.rsasecurity.com/rsalabs/node.asp?id=2248 - What is Diffie-Hellman Tanggal akses: 16 Oktober 2006 pukul 09:20

[4] http://www.strongsec.com/zhw/KSy_Crypto.pdf Tanggal akses: 16 Oktober 2006 pukul 09:20

[5] http://www.randombit.net/pipermail/botan-devel/2005-June/000064.html Tanggal akses: 16 Oktober 2006 pukul 09:20

[6] http://www.cs.bu.edu/~reyzin/teaching/f06cs538/notes-7.ps Tanggal akses: 16 Oktober 2006 pukul 09:20

[7] http://csrc.nist.gov/encryption/kms/p1363.ppt Tanggal akses: 16 Oktober 2006 pukul 09:20

[8] http://www.secg.org/collateral/SECG_X509_Alg_990218/SECG-X509-Alg-990218.PPT - ECC X.509 Certificate Format Discussion Tanggal akses: 16 Oktober 2006 pukul 09:20

[9] Krawczyk, Hugo. 2005. *HMQV: A High-Performance Secure* Diffie-Hellman Protocol.* IBM Research. Dapat diakses di http://eprint.iacr.org/2005/176.

[10] Menezes, Alfred. 2005. *Another Look At Hmqv*. Waterloo: University of Waterloo. Dapat diakses di http://eprint.iacr.org/2005/205.

[11] Munir, Rinaldi. 2006. Bahan Kuliah IF5054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[12] *Guidelines on Algorithms Usage and Key Management*. http://www.ecbs.org/Download/TR406V40.pdf. Tanggal akses: 16 Oktober 2006 pukul 09:20

[13] http://www.world-knowledge-encyclopedia.com/?t=MQV. Tanggal akses: 7 November 2006 pukul 11:02

[14] http://www.answers.com/topic/cryptographic-protocol. Tanggal akses: 7 November 2006 pukul 11:02

[15] http://www.answers.com/topic/mqv-1. Tanggal akses: 7 November 2006 pukul 11:02

[16] http://www.answers.com/topic/certicom. Tanggal akses: 7 November 2006 pukul 11:02

[17] http://experts.about.com/e/m/mq/mqv.htm. Tanggal akses: 7 November 2006 pukul 11:02

[18] http://experts.about.com/e/c/ce/certicom.htm. Tanggal akses: 7 November 2006 pukul 11:02

[19] http://experts.about.com/e/i/ie/ieee_p1363.htm. Tanggal akses: 7 November 2006 pukul 11:02

[20] http://csrc.nist.gov/encryption/kms/x942.pdf - *ANSI X9.42 Agreement of Symmetric Keys Using Discrete Logarithm*. Tanggal akses: 13 November 2006 pukul 9:28

[21] http://csrc.nist.gov/encryption/kms/objectives.pdf - *Workshop for the Development of a Federal Key Management Standard*. Tanggal akses: 13 November 2006 pukul 9:28

[22] http://www.certicom.com/index.php?action=res,cc_index. Tanggal akses: 7 November 2006 pukul 11:02

[23] http://www.certicom.com/index.php?action=res,ecc_faq. Tanggal akses: 7 November 2006 pukul 11:02

[24] http://www.cacr.math.uwaterloo.ca/techreports/1998/corr98-05.pdf. Tanggal akses: 7 November 2006 pukul 11:02

[25] http://www.certicom.com/download/aid-89/C&C_vol1_iss1.pdf. Tanggal akses: 7 November 2006 pukul 11:02

[26] http://www.certicom.com/download/aid-90/C&C_vol1_iss2.pdf. Tanggal akses: 7 November 2006 pukul 11:02

[27] http://www.certicom.com/download/aid-325/c&c_vol1_iss4.pdf. Tanggal akses: 7 November 2006 pukul 11:02

[28] *Recommendation on Key Establishment Schemes Draft 2.0*. January 2003. NIST.

[29] http://www.cacr.math.uwaterloo.ca/techreports/1998/corr98-05.pdf. Tanggal akses: 13 November 2006 pukul 9:28