On the Importance of Public-Key Validation in the MQV and HMQV Key Agreement Protocols

Alfred Menezes and Berkant Ustaoglu

Department of Combinatorics & Optimization, University of Waterloo
{ajmeneze, bustaoglu}@uwaterloo.ca

Abstract. HMQV is a hashed variant of the MQV key agreement protocol proposed by Krawczyk at CRYPTO 2005. In this paper, we present some attacks on HMQV and MQV that are successful if public keys are not properly validated. In particular, we present an attack on the two-pass HMQV protocol that does not require knowledge of the victim's ephemeral private keys. The attacks illustrate the importance of performing some form of public-key validation in Diffie-Hellman key agreement protocols, and furthermore highlight the dangers of relying on security proofs for discrete-logarithm protocols where a concrete representation for the underlying group is not specified.

1 Introduction

Public-key validation is a process whose purpose is to verify that a public key possesses certain arithmetic properties. Public-key validation is especially important in Diffie-Hellman protocols where a party $\hat{B}$ derives a secret session key K by combining his private key with a public key received from a second party $\hat{A}$ and subsequently uses K in some symmetric-key protocol (e.g., encryption or message authentication) with $\hat{A}$. A dishonest party $\hat{A}$ might select an invalid public key in such a way that the use of K reveals information about $\hat{B}$'s private key. Lim and Lee [18] demonstrated the importance of public-key validation by presenting *small-subgroup attacks* on some discrete logarithm key agreement protocols that are effective if the receiver of a group element does not verify that the element belongs to the desired group of high order (e.g., a prime-order DSA-type subgroup of $\mathbb{F}_p^*$). In [5,3], *invalid-curve attacks* were designed that are effective on elliptic curve protocols if the receiver of a point does not verify that the point indeed lies on the chosen elliptic curve. Kunz-Jacques et al. [15] showed that the zero-knowledge proof proposed in [4] for proving possession of discrete logarithms in groups of unknown order can be broken if a dishonest verifier selects invalid parameters during its interaction with the prover. More recently, Chen, Cheng and Smart [7] illustrated the importance of public-key validation in identity-based key agreement protocols that use bilinear pairings.

The MQV protocols [16] are a family of authenticated Diffie-Hellman protocols that have been widely standardized [1,2,9,27]. In the two-pass and three-pass versions of the protocol, the communicating parties $\hat{A}$ and $\hat{B}$ exchange static

R. Barua and T. Lange (Eds.): INDOCRYPT 2006, LNCS 4329, pp. 133–147, 2006.

(long-term) and ephemeral (short-term) public keys, and thereafter derive a secret key from these values. In the one-pass version, only one party contributes an ephemeral public key. In 2005, Krawczyk [12,13] presented the HMQV protocols, which are hashed variants of the MQV protocols. The primary advantages of HMQV over MQV are better performance and a rigorous security proof. The improved performance of HMQV is a direct consequence of not requiring the validation of ephemeral and static public keys — unlike with MQV where these operations are mandated. Despite the omission of public-key validation, Krawczyk was able to devise proofs that the HMQV protocols are secure in the random oracle model assuming the intractability of the computational Diffie-Hellman problem (and some variants thereof) in the underlying group.

Menezes [19] identified some flaws in the HMQV security proofs and presented small-subgroup attacks on the protocols. The attacks exploit the omission of validation for both ephemeral and static public keys, and allow an adversary to recover the victim's static private key. The attacks on the one-pass protocol are the most realistic, while the attacks on the two-pass and three-pass protocols are harder to mount in practice because the adversary needs to learn some of the victim's ephemeral private keys.

In this paper, we further investigate the effects of omitting public-key validation in HMQV and MQV. For the most part, we will only consider the two-pass HMQV protocol (which we call *the* HMQV protocol), which is the core member of the HMQV family. We identify a subtle flaw in the HMQV security proof which leads to an attack that does not require knowledge of ephemeral private keys, thereby contradicting the claim made in [13] that the HMQV protocol (without public-key validation) is provably secure if the adversary never learns any ephemeral private keys. We also consider the vulnerability of HMQV and MQV if only static public keys are validated, or if only ephemeral public keys are validated. These hypothetical scenarios are worth investigating because the reasons for omitting public-key validation can be different for ephemeral and static keys — validation of ephemeral public keys may be omitted for performance reasons, while validation of static public keys may be omitted because the certification authority may not be configured to perform such tests [13].

We emphasize that many of the attacks described in this paper cannot be mounted in realistic settings. For example, the aforementioned attack on HMQV that does not require knowledge of ephemeral private keys is described in certain underlying groups that have never been proposed for practical use. Moreover, this attack fails if the underlying group is a DSA-like group or a prime-order subgroup of an elliptic curve group as proposed for standardization in [14]. We also caution against inferring from our work that one must necessarily (fully) validate public keys in all Diffie-Hellman key agreement protocols. For example, the version of HMQV proposed in [14] only requires that a few simple and efficient checks be performed on static and ephemeral public keys. Moreover, even in the situation where one is concerned that ephemeral private keys might be leaked, [14] only requires that ephemeral and static public keys be *jointly* validated, thus saving a potentially expensive validation step (cf. §6).

The remainder of this paper is organized as follows. The MQV and HMQV protocols are reviewed in §2. The new attack on HMQV that does not require knowledge of ephemeral private keys is presented in §3, and the associated flaw in the HMQV security proof is identified. In §4 we present attacks on HMQV in the case where only ephemeral public keys are validated. In §5 it is shown that MQV is insecure if validation of ephemeral public keys is omitted. In §6 we describe the approach taken in [14] to guard against the kinds of attacks discussed in this paper. An example is presented in §7 to illustrate the potential pitfalls if public keys are not completely validated. The paper concludes with some remarks in §8.

2 The HMQV Key Agreement Protocol

Let $G = \langle g \rangle$ be a multiplicatively-written cyclic group of prime order q, and let 1 denote the identity element in G. Let H be a hash function, and let $\overline{H}$ be an l-bit hash function where $l = (\lfloor \log_2 q \rfloor + 1)/2$. Party $\hat{A}$'s static private key is an integer $a \in_R [0, q-1]$, while her static public key is the group element $A = g^a$. Similarly, party $\hat{B}$ has a static key pair (B, b) where $b \in_R [0, q-1]$ and $B = g^b$.

2.1 Description of HMQV

In the (two-pass) HMQV protocol as presented in [12,13], parties $\hat{A}$ and $\hat{B}$ establish a secret session key as follows:

1. $\hat{A}$ selects an ephemeral private key $x \in_R [0, q-1]$ and computes her ephemeral public key $X = g^x$. $\hat{A}$ then sends $(\hat{A}, \hat{B}, X)$ to $\hat{B}$.
2. Upon receiving $(\hat{A}, \hat{B}, X)$, $\hat{B}$ checks that $X \neq 0$,[1] selects an ephemeral key pair (Y, y), and sends $(\hat{B}, \hat{A}, Y)$ to $\hat{A}$. $\hat{B}$ proceeds to compute $s_B = y + eb \bmod q$ and $\sigma = (XA^d)^{s_B}$ where $d = \overline{H}(X, \hat{B})$ and $e = \overline{H}(Y, \hat{A})$.[2]
3. Upon receiving $(\hat{B}, \hat{A}, Y)$, $\hat{A}$ checks that $Y \neq 0$, and computes $s_A = x + da \bmod q$ and $\sigma = (YB^e)^{s_A}$ where again $d = \overline{H}(X, \hat{B})$ and $e = \overline{H}(Y, \hat{A})$.
4. The secret session key is $K = H(\sigma) = H(g^{s_A s_B})$.

The messages transmitted in steps (1) and (2) may include certificates for the static public keys A and B, respectively. Note that HMQV does not mandate that static and ephemeral public keys be validated, i.e., verified as being non-identity elements of G.

[1] Note that $0 \notin G$. The check $X \neq 0$ (and $Y \neq 0$) makes sense in some settings, e.g., when G is a multiplicative subgroup of a finite field; in this case 0 is the additive identity of the field.

[2] The HMQV papers [12,13] do not explicitly state that s_A (and s_B) should be computed modulo q. The attacks in this paper can still be launched if s_A (and s_B) are not reduced modulo q.

2.2 Description of MQV

The three essential differences between the MQV protocol (as standardized in [27]) and the HMQV protocol are the following:

1. Static and ephemeral public keys must be validated in MQV.[3]
2. In MQV, the integers d and e are derived from the group elements X and Y, respectively. For example, if G is a group of elliptic curve points, then d and e are derived from the l least significant bits of the x-coordinates of X and Y respectively.
3. The secret session key is $K = H(\sigma, \hat{A}, \hat{B})$.[4]

2.3 Security Proofs

Krawczyk [12,13] provided a very extensive analysis of HMQV. He proved that the protocol satisfies the Canetti-Krawczyk definition [6] for secure key agreement, under the assumptions that H and $\overline{H}$ are random oracles and that the computational Diffie-Hellman (CDH) problem[5] in G is intractable. The Canetti-Krawczyk security definition is a very strong one in that the adversary controls all communications between parties and its goal is very modest — distinguishing a target session key from a purely random key. The protocol remains secure even if the adversary is allowed to learn session keys different from the target session key. Krawczyk also proved that the protocol is resistant to attacks when the adversary is permitted to learn the ephemeral private keys of sessions; for this property the 'Gap Diffie-Hellman' and 'KEA1' assumptions about G are needed.

2.4 An Attack

We describe the attack on HMQV that was presented in [19]. The attack exploits the omission of public-key validation for ephemeral and static public keys, and also the ability of the adversary to learn the victim's ephemeral private keys.

We suppose that there is an algebraic structure R (e.g., a field, ring, or group) such that:

1. The elements of R are represented in the same format as elements of G (e.g., bitstrings of the same length).
2. The group operation for G is defined on elements of R.

For the attack in this section, we further assume that there is a subset G' of R such that:

3. G' is a cyclic group with respect to the operation defined on G.
4. G' has order t where $t = 2^r$ for some small r (e.g., $r = 4$).

[3] Actually 'embedded' validation may be performed on ephemeral public keys. The details are not relevant to the attacks presented in this paper.

[4] The identities $\hat{A}$, $\hat{B}$ are included in the derivation of K from σ in order to thwart Kaliski's unknown-key share attack [10].

[5] The CDH problem in $G = \langle g \rangle$ (with respect to g) is that of computing $\mathrm{CDH}(X, Y) = X^y = Y^x$ given g, $X = g^x$ and $Y = g^y$.

For example, if G is amultiplicative subgroup of order q of $\mathbb{F}_p^*$ (the integers modulo p) and $t \mid (p-1)/q$, then we can take $R = \mathbb{F}_p$ and G' to be the unique subgroup of $\mathbb{F}_p^*$ of order t. Note that elements of G and R have the same representation (integers modulo p), and the common operation is multiplication modulo p. As a second example, suppose that $G = E(\mathbb{F}_p)$ where $E : V^2 = U^3 + \alpha U + \beta$ is an (additively-written) elliptic curve defined over $\mathbb{F}_p$, and let $E' : V^2 = U^3 + \alpha U + \beta'$ be another elliptic curve defined over $\mathbb{F}_p$ such that $t \mid \#E'(\mathbb{F}_p)$. Then we can take $R = E'(\mathbb{F}_p)$ and G' to be a subgroup of $E'(\mathbb{F}_p)$ of order t. Again, the elements of G and R have the same representation (pairs of integers modulo p), and the group law for E and E' are the same since the usual chord-and-tangent laws for E and E' do not (explicitly) use the coefficients β and β' (see §4).

The attack proceeds as follows. The adversary $\hat{A}$ chooses an element $\gamma \in G'$ of order $t = 2^r$, selects $a, x \in [1, t-1]$, computes $A = \gamma^a$ and $X = \gamma^x$, and sends $(\hat{A}, \hat{B}, X)$ to $\hat{B}$. While $\hat{B}$ is computing the session key $K = H((XA^d)^{s_B})$, the adversary learns $\hat{B}$'s ephemeral private key y. Let $\beta = XA^d = \gamma^{x+da}$, so $K = H(\beta^{s_B})$. $\hat{A}$ then learns the session key K.[6] Now $\hat{A}$ computes $K' = H(\beta^c)$ for $c = 0, 1, 2, \ldots, t-1$ until $K' = K$, in which case $\hat{A}$ has determined $s_B \bmod t$. After repeating this procedure a few times, $\hat{A}$ can use the Leadbitter-Smart lattice attack [17] to find the l most significant bits of b. The remaining l bits of b can thereafter be determined in $O(q^{1/4})$ time using Pollard's lambda method [24].

3 No Ephemeral Private Key Leakage

The adversary in the attack of §2.4 requires knowledge of the victim's ephemeral private keys. While resistance to ephemeral private key leakage is a desirable attribute of a key establishment protocol[7], it is arguably not a fundamental security requirement. In [13] it is claimed that the HMQV protocol is *provably secure* if the adversary does not learn any ephemeral private keys. In this section we demonstrate that this claim is false.

3.1 A New Attack

Suppose that $G = \langle g \rangle$ is a multiplicatively-written group of prime order q, and suppose that the CDH problem in G is intractable. We further assume that R is a ring such that:

1. The elements of R are represented in the same format as elements of G (e.g., bitstrings of the same length).
2. The multiplication operation for R is defined in the same way as the operation for G. In particular, G is a subgroup of the group of units $U(R)$ of R.

[6] Suppose, for example, that $\hat{B}$ sends $\hat{A}$ an authenticated message $(m, \tau = \mathrm{MAC}_K(m))$. Then $\hat{A}$ can learn K by computing $\tau' = \mathrm{MAC}_{K'}(m)$ where $K' = H(\beta^c)$ for $c = 0, 1, 2, \ldots, t-1$ until $\tau' = \tau$.

[7] In [12,13], resistance of Diffie-Hellman protocols to damage from the disclosure of ephemeral private keys is described as a 'prime security concern'.

3. There exists an element $T \in R$, $T \neq 0$, such that $T^2 = 0$ (where “0” denotes the additive identity element in R).

The new attack on HMQV assumes that parties do not validate ephemeral public keys. The adversary $\hat{C}$ intercepts the message $(\hat{A}, \hat{B}, X)$ sent by $\hat{A}$ and replaces it with $(\hat{A}, \hat{B}, T)$. Similarly, $\hat{C}$ intercepts $\hat{B}$'s response $(\hat{B}, \hat{A}, Y)$ and forwards $(\hat{B}, \hat{A}, T)$ to $\hat{A}$. If R is commutative then, assuming that $s_A \geq 2$ and $s_B \geq 2$, it is easy to see that both $\hat{A}$ and $\hat{B}$ compute the session key $K = H(0)$. Of course, $\hat{C}$ can also compute this key.

If R is not commutative, then the value of the session key depends on the particular exponentiation method used by the parties. Suppose that $\hat{A}$ determines the session key by first calculating $t_A = es_A \bmod q$ and then using simultaneous multiple exponentiation [20, Algorithm 14.88] to compute $\sigma = T^{s_A} B^{t_A}$ and $K = H(\sigma)$. This algorithm first computes TB and initializes an accumulator to 1. It then repeatedly examines the bits of s_A and t_A from left to right. During each iteration, either 1, T, B or TB is multiplied into the accumulator which is then squared. Now, if the most significant bits of s_A and t_A are 1 and 0, respectively, then the accumulator takes on the values $1, T, T^2, \ldots$. Hence $\hat{A}$ computes $\sigma = 0$. Similarly, $\hat{B}$ may compute $\sigma = 0$, in which case $\hat{C}$ also learns the session key $K = H(0)$.

3.2 Examples of Groups

We give two examples of groups that satisfy the conditions of §3.1. These examples do not have any immediate practical relevance since such groups are not being deployed in practice. Nonetheless, they serve to refute the claim made in [12,13] that HMQV is provably secure regardless of the representation used for the elements of the G (subject to the constraint that the CDH problem in G be intractable).

A Commutative Example. Let p be a 1024-bit prime such that $p-1$ has a 160-bit prime divisor q. Consider the commutative ring $R = \mathbb{Z}_{p^2}$. Then $U(R)$ is cyclic and $\#U(R) = p(p-1)$. Let G be the order-q cyclic subgroup of $U(R)$. The CDH problem in G is believed to be intractable. The element $T = p \in \mathbb{Z}_{p^2}$ satisfies $T \neq 0$ and $T^2 = 0$.

A Non-commutative Example. Again, let p be a 1024-bit prime such that $p-1$ has a 160-bit prime divisor q. Consider the non-commutative ring R of 2×2 matrices over $\mathbb{F}_p$. Then $\#U(R) = (p^2-1)(p^2-p)$. Let $g \in U(R)$ be an element of order q, and let $G = \langle g \rangle$. The CDH problem in G is equivalent to the CDH problem in the order-q subgroup of $\mathbb{F}_p^*$ (see [21]) and is therefore believed to be intractable. The element

$$T = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

satisfies $T \neq 0$ and $T^2 = 0$.

3.3 Flaw in the HMQV Proof

The HMQV security proof in [13] has two main steps. First, an 'exponential challenge-response' signature scheme XCR is defined and proven secure in the random oracle model under the assumption that the CDH problem in G is intractable. Second, the security of XCR (actually a 'dual' version of XCR) is proven to imply the security of HMQV.

In the XCR signature scheme, a verifier $\hat{A}$ selects $x \in_R [0, q-1]$ and sends the challenge $X = g^x$ and a message m to the signer $\hat{B}$. $\hat{B}$ responds by selecting $y \in_R [0, q-1]$ and sending the signature $(Y = g^y, \sigma = X^{s_B})$ to $\hat{A}$ where $s_B = y + eb \bmod q$, $e = \overline{H}(Y, m)$, and (B, b) is $\hat{B}$'s static key pair. The signature is accepted by $\hat{A}$ provided that $Y \neq 0$ and $\sigma = (YB^e)^x$. XCR signatures are different from ordinary digital signatures — $\hat{A}$ cannot convince a third party that $\hat{B}$ generated a signature (Y, σ) for message m and challenge X because $\hat{A}$ could have generated this signature herself.

The XCR security proof in [12,13] uses the forking lemma of Pointcheval and Stern [23]. The proof hypothesizes the existence of a forger who, on input $B, X_0 \in_R G$ and a signing oracle for $\hat{B}$, produces a message m_0 and a valid signature (Y_0, σ) for m_0;[8] that is $Y_0 \neq 0$ and $\sigma = (Y_0 B^e)^{x_0}$ where $e = \overline{H}(Y_0, m_0)$ and $X_0 = g^{x_0}$.[9] Now, in order to compute CDH(B, X_0), the forger is run twice with input B, X_0. The forger's hash function and signature queries are suitably answered so that the two invocations of the forger eventually produce valid forgeries (m_0, Y_0, σ) and (m_0, Y_0, σ') where $e = \overline{H}(Y_0, m_0)$, $e' = \overline{H}'(Y_0, m_0)$, and $e \not\equiv e' \pmod{q}$. To conclude the argument, one notes that

$$\frac{\sigma}{\sigma'} = \frac{(Y_0 B^e)^{x_0}}{(Y_0 B^{e'})^{x_0}} = (B^{x_0})^{e-e'} \tag{1}$$

whence CDH$(B, X_0) = (\sigma/\sigma')^{(e-e')^{-1}}$ can be efficiently computed.

The flaw in this argument is the assumption that the Y_0 terms in (1) can be cancelled under the sole condition that $Y_0 \neq 0$. While the cancellation in (1) is valid if $Y_0 \in G$ (which is the case if Y_0 has been validated), in general one needs to make additional assumptions including that Y_0 is invertible. Thus, since the description of XCR does not mandate that the verifier validate Y, the XCR security proof in [12,13] is incorrect.

This flaw in the XCR security proof accounts for the following attack on XCR. Let R and T be as defined in §3.1, and suppose for the sake of concreteness that R is commutative. A forger can respond to $\hat{A}$'s challenge (X, m) with the signature $(Y = T, \sigma = 0)$. The signature is accepted by $\hat{A}$ since $T \neq 0$ and $(TB^e)^x = 0$. This attack on XCR in turn explains why the attack described in §3.1 can be launched on HMQV.

[8] There is also the requirement that (m_0, Y_0) did not appear in any of $\hat{B}$'s responses to the forger's signature queries.

[9] The XCR security definition in [12,13] incorrectly states that the forger's output (m_0, Y_0, σ) should satisfy $Y_0 \neq 0$ and $\sigma = X_0^{y_0+eb}$ where $Y = g^{y_0}$. The latter condition is *not* equivalent to the condition $\sigma = (Y_0 B^e)^{x_0}$ in the case where $Y_0 \notin G$ — indeed y_0 is not even defined when $Y_0 \notin G$.

4 No Static Public-Key Validation

We describe an attack on HMQV in the hypothetical situation where ephemeral public keys are validated but static public keys aren't. As mentioned in the Introduction, this situation is worth investigating because validation for static public keys may be omitted if a certification authority is not configured to perform such tests. We describe attacks that can be mounted in the realistic setting where G is a DSA-type group or an elliptic curve group.

4.1 DSA-Type Groups

We suppose that G is the order-q subgroup of $\mathbb{F}_p^*$, and that $t = 2^r$ is a divisor of $(p-1)/q$. Let $\gamma \in \mathbb{F}_p^*$ be an element of order t. Using the notation introduced in §2.4, we have $R = \mathbb{F}_p$ and $G' = \langle\gamma\rangle$.

The adversary $\hat{A}$ selects a valid $X \in G$ and computes $d = (\overline{H}(X, \hat{B}))^{-1} \bmod q$ and $A = \gamma X^{-d^{-1} \bmod q}$. She certifies A as her (invalid) static public key and sends X to $\hat{B}$ who computes $\beta = XA^d = \gamma^d$ and $K = H(\beta^{s_B})$. As in the attack described in §2.4, $\hat{A}$ learns y, K, and $s_B \bmod t$; repeating this procedure yields half the bits of b.

4.2 Elliptic Curves Groups

We suppose that $G = E(\mathbb{F}_p)$ where $E : V^2 = U^3 + \alpha U + \beta$ is an elliptic curve of prime order defined over the prime field $\mathbb{F}_p$. Let $P_1 = (u_1, v_1)$ and $P_2 = (u_2, v_2)$ be two finite points in $E(\mathbb{F}_p)$ with $P_1 \neq -P_2$, and let $P_3 = (u_3, v_3) = P_1 + P_2$. The usual formulae for computing P_3 are:

$$u_3 = \lambda^2 - u_1 - u_2, \tag{2}$$
$$v_3 = \lambda(u_1 - u_3) - v_1, \tag{3}$$

where

$$\lambda = \frac{v_2 - v_1}{u_2 - u_1} \quad \text{or} \quad \lambda = \frac{3u_1^2 + \alpha}{2v_1},$$

depending on whether $P_1 \neq P_2$ or $P_1 = P_2$. Note that the formulae do not (explicitly) depend on the coefficient β.

The adversary $\hat{A}$'s goal is to select two points $A, X \in \mathbb{F}_p \times \mathbb{F}_p$ such that (i) X is valid, i.e., $X \in E(\mathbb{F}_p)$, $X \neq \infty$; and (ii) $T = X + dA$ is a point of order 16 on some curve $E' : V^2 = U^3 + \alpha U + \beta'$ defined over $\mathbb{F}_p$, where $d = \overline{H}(X, \hat{B})$ and $X + dA$ is computed using the formulas for $E(\mathbb{F}_p)$. Using the notation introduced in §2.4, we have $R = E'(\mathbb{F}_p)$, $G' = \langle T\rangle$, and $t = 16$. The adversary then certifies A as her (invalid) static public key and sends X to $\hat{B}$, who computes $K = H(s_B T)$. As in the attack described in §2.4, $\hat{A}$ learns y, K, and $s_B \bmod t$; repeating this procedure yields half the bits of b.

The adversary can proceed to determine A and X as follows. She first selects an arbitrary finite point $X = (u_2, v_2) \in E(\mathbb{F}_p)$ such that $d = \overline{H}(X, \hat{B})$ is odd.

Now let $A = (z, 0)$, where $z \in \mathbb{F}_p$ is an indeterminate whose value will be specified later. Since d is odd, application of the group law for E yields $dA = A$. The coordinates (u_3, v_3) of $T = X + dA$ are then derived using (2) and (3):

$$u_3 = \left(\frac{v_2}{u_2 - z}\right)^2 - z - u_2 \quad \text{and} \quad v_3 = \frac{v_2}{u_2 - z}(z - u_3). \tag{4}$$

Define

$$\beta' = v_3^2 - u_3^3 - \alpha u_3, \tag{5}$$

so that $T = (u_3, v_3)$ is an $\mathbb{F}_p$-point on the elliptic curve

$$E' : V^2 = U^3 + \alpha U + \beta'. \tag{6}$$

We next show how division polynomials can be used to select $z \in \mathbb{F}_p$ so that T has order 16. The following result is well known (e.g., see [25]).

Theorem 1. Consider the division polynomials $\Psi_k(U, V) \in \mathbb{F}_p[U, V]$ associated with an elliptic curve $E/\mathbb{F}_p : V^2 = U^3 + \alpha U + \beta$ and defined recursively as follows:

$$\begin{aligned}
\Psi_1(U, V) &= 1 \\
\Psi_2(U, V) &= 2V \\
\Psi_3(U, V) &= 3U^4 + 6\alpha U^2 + 12\beta U - \alpha^2 \\
\Psi_4(U, V) &= 4V(U^6 + 5\alpha U^4 + 20\beta U^3 - 5\alpha^2 U^2 - 4\alpha\beta U - 8\beta^2 - \alpha^3) \\
\Psi_{2k+1}(U, V) &= \Psi_{k+2}\Psi_k^3 - \Psi_{k+1}^3\Psi_{k-1} \text{ for } k \geq 2 \\
\Psi_{2k}(U, V) &= \Psi_k(\Psi_{k+2}\Psi_{k-1}^2 - \Psi_{k-2}\Psi_{k+1}^2)/2V \text{ for } k \geq 3.
\end{aligned}$$

Let Ψ_k' be the polynomial obtained by repeatedly replacing occurrences of V^2 in Ψ_k by $U^3 + \alpha U + \beta$, and define

$$f_k = \begin{cases} \Psi_k'(U, V), & \text{if } k \text{ is odd}, \\ \Psi_k'(U, V)/V, & \text{if } k \text{ is even}. \end{cases}$$

Then in fact $f_k \in \mathbb{F}_p[U]$. Moreover, if $P = (u, v) \in E(\overline{\mathbb{F}}_p)$ such that $2P \neq \infty$, then $kP = \infty$ if and only if $f_k(u) = 0$.

It follows from Theorem 1 that the roots of the polynomial

$$g(U) = \frac{f_{16}(U)}{f_8(U)}$$

are precisely the U-coordinates of points of order 16 in $E(\overline{\mathbb{F}}_p)$, and hence $\deg(g) = 96$.

Now to determine T, the adversary computes $h(z) = g(u_3)$, where $g(U)$ is associated with $E' : V^2 = U^3 + \alpha U + \beta'$, and where u_3 and β' are defined in (4) and (5). It can be seen that $h(z) = h_1(z)/h_2(z)$, where $h_1, h_2 \in \mathbb{F}_p[z]$ and

$\deg(h_1) = 288$.[10] If the polynomial h_1 has a root z in $\mathbb{F}_p$, then the associated point T is guaranteed to have order 16 in $E'(\mathbb{F}_p)$. Since X can be chosen uniformly at random from $E(\mathbb{F}_p)$, it is reasonable to make the heuristic assumption that h_1 is a "random" degree-288 polynomial over $\mathbb{F}_p$. The following result ensures that there is a very good chance that h_1 will indeed have a root in $\mathbb{F}_p$. The result is well known (e.g., see Exercise 1 in §4.6.2 of [11]), but we include its proof for the sake of completeness.

Lemma 1. For $p \gg n \geq 10$, the proportion of degree-n polynomials over $\mathbb{F}_p$ that have at least one root in $\mathbb{F}_p$ is approximately $(1 - \frac{1}{e}) \approx 0.632$.

Proof. It suffices to consider monic polynomials over $\mathbb{F}_p$.

The generating function for the number of monic polynomials over $\mathbb{F}_p$ with respect to degree is

$$\Phi(x) = \sum_{i \geq 0} p^i x^i = \frac{1}{1 - px}. \tag{7}$$

Let $L(n, p)$ denote the number of degree-n monic irreducible polynomials over $\mathbb{F}_p$. Since every monic polynomial can be written as a product of monic irreducible polynomials, the generating function $\Phi(x)$ can be written as

$$\Phi(x) = \prod_{i \geq 1} \left(\frac{1}{1 - x^i} \right)^{L(i,p)}. \tag{8}$$

Now, the generating function for monic polynomials with no linear factors (i.e., no roots in $\mathbb{F}_p$) is

$$\tilde{\Phi}(x) = \prod_{i \geq 2} \left(\frac{1}{1 - x^i} \right)^{L(i,p)}. \tag{9}$$

Multiplying (8) by $(1 - x)^{L(1,p)} = (1 - x)^p$ yields

$$\tilde{\Phi}(x) = \frac{(1 - x)^p}{1 - px}. \tag{10}$$

Letting $[\cdot]$ denote the coefficient operator, it follows from (10) that the number $R(n, p)$ of monic polynomials of degree n over $\mathbb{F}_p$ that have at least one root in $\mathbb{F}_p$ is

$$R(n, p) = p^n - [x^n]\tilde{\Phi}(x) = p^n - \sum_{i=0}^{n} \binom{p}{i} (-1)^i p^{n-i}.$$

For $p \gg n \geq 10$, we have

$$R(n, p) \approx p^n \sum_{i=1}^{n} \frac{(-1)^{i-1}}{i!} \approx p^n \sum_{i \geq 1} \frac{(-1)^{i-1}}{i!} = p^n \left(1 - \frac{1}{e} \right).$$

[10] More generally, if $t = 2^r$ then $\deg(g) = 3 \cdot 2^{2r-3}$ and $\deg(h_1) = 9 \cdot 2^{2r-3}$.

Example 1. (determination of A, X, T and E') Consider the NIST-recommended elliptic curve [8] defined by the equation $E : V^2 = U^3 - 3U + \beta$ over $\mathbb{F}_p$, where $p = 2^{192} - 2^{64} - 1$ and

$$\beta = 2455155546008943817740293915197451784769108058161191238065.$$

Suppose that we select

$$X = (602046282375688656758213480587526111916698976636884684818,\\ 174050332293622031404857552280219410364023488927386650641)$$

in $E(\mathbb{F}_p)$, and

$$A = (2664590514587922359853612565516270937783866981812798250851,\ 0).$$

Then the point $T = X + A$ computed using the group law for $E(\mathbb{F}_p)$ is

$$T = (5350077178842604929587851454217201721791103389533004256989,\\ 4170329249603673452251890924513609385018269372344921771517).$$

T is a point of order 16 on $E' : y^2 = x^3 - 3x + \beta'$, where

$$\beta' = 2271835836669632292423953498680460143165540922751246538627.$$

5 No Ephemeral Public-Key Validation

In this section we consider attacks in the hypothetical situation where static public keys are validated but ephemeral public keys aren't. We don't know of any attacks on HMQV in the case where the underlying group G is a DSA-type group or an elliptic curve group (cf. §4.1 and §4.2). In particular, we don't know how to extend the attacks described in §4.1 and §4.2 to this setting. The difficulty is in part because of the complicated relationship between X and $d = \overline{H}(X, \hat{A})$ whereby d is not determined until X has been fixed.

However, we observe that attacks can be launched on MQV if ephemeral public keys are not validated. Suppose that $G = E(\mathbb{F}_p)$ where $E/\mathbb{F}_p : V^2 = U^3 + \alpha U + \beta$ is an elliptic curve of prime order. The adversary $\hat{C}$, who wishes to impersonate $\hat{A}$ to $\hat{B}$, selects $u_1 \in_R \mathbb{F}_p$ and sets $X = (u_1, z)$ where z is an indeterminate. Since in MQV d depends only on u_1, $\hat{C}$ can then compute $\tilde{A} = dA$, where A is $\hat{A}$'s (valid) static public key. Using the method of §4.2, $\hat{C}$ can use the t-th division polynomial (for some small t) to determine $z, \beta' \in \mathbb{F}_p$ so that $T = X + \tilde{A}$ has order t on $E' : V^2 = U^3 + \alpha U + \beta'$. The adversary sends X to $\hat{B}$ who computes the session key $K = H(T^{s_B}, \hat{A}, \hat{B})$. Now $\hat{C}$ can guess the session key with non-negligible success probability $\frac{1}{t}$. Alternatively, if $\hat{C}$ can learn $\hat{B}$ ephemeral's private keys y, then $\hat{C}$ can determine $\hat{B}$'s static private key b as in §2.4.

6 Partial Validation

It may be possible to circumvent the attacks described in the preceding sections without performing (full) public-key validation on static and ephemeral public keys. For example, consider the version of HMQV that has recently been proposed for standardization by the IEEE 1363 standards group [14]. This proposal specifies HMQV in the concrete setting of a DSA-type group G, i.e., G is the order-q subgroup of the multiplicative group $\mathbb{F}_p^*$ of a prime field. The only checks required on ephemeral and static public keys is that they be integers in the interval $[2, p-1]$. In [14] it is claimed that this instantiation of HMQV is provably secure (under the assumptions that CDH in G is intractable, and that the employed hash functions are random functions) as long as ephemeral private keys are never leaked. Moreover, in order to resist attacks that may be mounted in the face of ephemeral private key leakage, the recipient of an ephemeral key X and static key A only needs to verify that $Z^q = 1$ and $Z \neq 1$ where $Z = XA^d$. Such a check is more efficient that separately verifying that $A^q = 1$ and $X^q = 1$. Again, [14] claims that this version of HMQV is provably secure even if the adversary is able to learn some ephemeral private keys.

7 Almost Validation

A public key X is said to have been *almost validated* if it has been verified that $X \in G$ but not necessarily that $X \neq 1$. Protocol descriptions sometimes inadvertently omit the condition $X \neq 1$; see for example the 'G-tests' in [13]. Performing almost validation instead of full validation of public keys may lead to new vulnerabilities. This section gives an example of this likelihood.

In the one-pass HMQV protocol [13], only the initiator contributes an ephemeral public key. The initiator $\hat{A}$ sends $(\hat{A}, \hat{B}, X)$ to $\hat{B}$ and computes the session key $K = H(B^{s_A})$ where $s_A = x + da \bmod q$ and $d = \overline{H}(X, \hat{A}, \hat{B})$. The receiver $\hat{B}$ verifies that $X \neq 0$ and computes $K = H((XA^d)^b)$.

In [19] it was shown that the one-pass HMQV protocol succumbs to a Kaliski-style unknown-key share attack [10] even if public keys are (fully) validated. The attack is 'on-line' in the sense that the adversary needs to have her static public key certified during the attack. We next present an 'off-line' Kaliski-style attack on the one-pass HMQV protocol which succeeds if ephemeral public are (fully) validated but static public keys are only almost validated.

The adversary $\hat{C}$ registers in advance the static public key $C = 1$ with the certification authority. Now, when $\hat{A}$ sends $(\hat{A}, \hat{B}, X)$, $\hat{C}$ replaces this message with $(\hat{A}, \hat{C}, T)$ where $T = XA^d$ and $d = \overline{H}(X, \hat{A}, \hat{B})$. Note that T is valid, whereas C is only partially valid. The recipient $\hat{B}$ computes $d' = \overline{H}(T, \hat{C}, \hat{B})$ and

$$K = H((TC^{d'})^b) = H(T^b) = H((XA^d)^b).$$

Thus $\hat{A}$ and $\hat{B}$ have computed the same session key, but $\hat{B}$ mistakenly believes that the key is shared with $\hat{C}$.

8 Concluding Remarks

The attacks on HMQV presented in §2.4, §3.1 and §4 are also effective on MQV if validation of static or ephemeral public keys is omitted. The attacks are summarized in Table 1. While these attacks are not necessarily practical and may not be a threat in real-world settings, they nonetheless illustrate the importance of performing some form of validation for static and ephemeral public keys in Diffie-Hellman key agreement protocols. Furthermore, the attacks highlight the danger of relying on security proofs for discrete-logarithm protocols where a concrete representation for the underlying group is not specified. In particular, since public keys in HMQV are not necessarily valid, the security of HMQV depends on several aspects of the representation for the underlying group G including the manner in which the group operation is performed, and the particular algorithm chosen for computing $(XA^d)^{s_B}$ and $(YB^e)^{s_A}$. For other examples of the pitfalls when relying on security proofs where a concrete representation of the underlying group is not specified, see [22] and [26].

Table 1. Attacks on HMQV (and MQV without validation) described in this paper. † The attack of §5 applies to MQV only.

Static public keys validated?	Ephemeral public keys validated?	Ephemeral private keys secure?	Attacks
√	√	√	No attack known
√	√	×	No attack known
×	√	√	No attack known
√	×	√	§3.1, §5†
√	×	×	§3.1, §5†
×	×	√	§3.1, §5†
×	√	×	§4.1, §4.2
×	×	×	§2.4, §3.1, §4.1, §4.2, §5†

Acknowledgements

We would like to thank Daniel Panario for his help with the proof of Lemma 1. We also thank Darrel Hankerson for his comments on earlier drafts of this paper.

References

1. ANSI X9.42, *Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography*, American National Standards Institute, 2003.
2. ANSI X9.63, *Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography*, American National Standards Institute, 2001.

3. A. Antipa, D. Brown, A. Menezes, R. Struik and S. Vanstone, “Validation of elliptic curve public keys”, *Public Key Cryptography – PKC 2003*, Lecture Notes in Computer Science, 2567 (2003), 211-223.
4. E. Bangerter, J. Camenisch and U. Maurer, “Efficient proofs of knowledge of discrete logarithms and representations in groups with hidden order”, *Public Key Cryptography – PKC 2005*, Lecture Notes in Computer Science, 3386 (2005), 154-171.
5. I. Biehl, B. Meyer and V. Müller, “Differential fault analysis on elliptic curve cryptosystems”, *Advances in Cryptology – CRYPTO 2000*, Lecture Notes in Computer Science, 1880 (2000), 131-146.
6. R. Canetti and H. Krawczyk, “Analysis of key-exchange protocols and their use for building secure channels”, *Advances in Cryptology – EUROCRYPT 2001*, Lecture Notes in Computer Science, 2045 (2001), 453-474. Full version available at http://eprint.iacr.org/2001/040/.
7. L. Chen, Z. Cheng and N. Smart, “Identity-based key agreement protocols from pairings”, Cryptology ePrint Archive: Report 2006/199. Available at http://eprint.iacr.org/2006/199.
8. FIPS 186-2, *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication 186-2, National Institute of Standards and Technology, 2000.
9. IEEE Std 1363-2000, *Standard Specifications for Public-Key Cryptography*, 2000.
10. B. Kaliski, “An unknown key-share attack on the MQV key agreement protocol”, *ACM Transactions on Information and System Security*, 4 (2001), 275-288.
11. D. Knuth, *Seminumerical Algorithms*, vol. 2 of *Art of Computer Programming*, 3rd ed., Addison-Wesley, 1997.
12. H. Krawczyk, “HMQV: A high-performance secure Diffie-Hellman protocol”, *Advances in Cryptology – CRYPTO 2005*, Lecture Notes in Computer Science, 3621 (2005), 546-566.
13. H. Krawczyk, “HMQV: A high-performance secure Diffie-Hellman protocol”, Full version of [12], available at http://eprint.iacr.org/2005/176/.
14. H. Krawczyk, “HMQV in IEEE P1363”, submission to the IEEE P1363 working group, July 7 2006. Available at http://grouper.ieee.org/groups/1363/P1363-Reaffirm/submissions/krawczyk-hmqv-spec.pdf.
15. S. Kunz-Jacques, G. Martinet, G. Poupard and J. Stern, “Cryptanalysis of an efficient proof of knowledge of discrete logarithm”, *Public Key Cryptography – PKC 2006*, Lecture Notes in Computer Science, 3958 (2006), 27-43.
16. L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, “An efficient protocol for authenticated key agreement”, *Designs, Codes and Cryptography*, 28 (2003), 119-134.
17. P. Leadbitter and N. Smart, “Analysis of the insecurity of ECMQV with partially known nonces”, *Information Security – ISC 2003*, Lecture Notes in Computer Science, 2851 (2003), 240-251.
18. C. Lim and P. Lee, “A key recovery attack on discrete log-based schemes using a prime order subgroup”, *Advances in Cryptology – CRYPTO ’97*, Lecture Notes in Computer Science, 1294 (1997), 249-263.
19. A. Menezes, “Another look at HMQV”, *Journal of Mathematical Cryptology*, to appear. Available at http://eprint.iacr.org/2005/205/.
20. A. Menezes, P. van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1996.
21. A. Menezes and Y.-H. Wu, “The discrete logarithm problem in $GL(n,q)$, *Ars Combinatoria*, 47 (1998), 23-32.

22. D. Naccache, N. Smart, and J. Stern, "Projective coordinates leak", *Advances in Cryptology – EUROCRYPT 2004*, Lecture Notes in Computer Science, 3027 (2004), 257-267.
23. D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures", *Journal of Cryptology*, 13 (2000), 361-396.
24. J. Pollard, "Monte Carlo methods for index computation mod p", *Mathematics of Computation*, 32 (1978), 918-924.
25. R. Schoof, "Elliptic curves over finite fields and the computation of square roots mod p", *Mathematics of Computation*, 44 (1985), 483-494.
26. N. Smart, "The exact security of ECIES in the generic group model", *Cryptography and Coding*, Lecture Notes in Computer Science, 2260 (2001), 73-84.
27. SP 800-56A *Special Publication 800-56A, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, National Institute of Standards and Technology, March 2006.