# 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.
© Springer-Verlag Berlin Heidelberg 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).