ON REUSING EPHEMERAL KEYS IN DIFFIE-HELLMAN KEY AGREEMENT PROTOCOLS

ALFRED MENEZES AND BERKANT USTAOGLU

Abstract. A party may choose to reuse ephemeral public keys in a Diffie-Hellman key agreement protocol in order to reduce its computational workload or to mitigate against denial-of-service attacks. In this note we highlight the danger of reusing ephemeral keys if domain parameters are not appropriately selected or if public keys are not appropriately validated.

1. Introduction

Let G be a multiplicatively-written group of prime order q, and let g be a generator of G. In the classic Diffie-Hellman key agreement protocol, two parties $\hat{A}$ and $\hat{B}$ exchange public keys $X = g^x$ and $Y = g^y$, where $x \in_R [1, q-1]$ and $y \in_R [1, q-1]$ are their respective private keys, and thereafter compute the shared secret $\sigma = X^y = Y^x = g^{xy}$. A session key is then derived from σ, usually by hashing.

A public key may either be *static* (long-lived) or *ephemeral* (short-lived). Ephemeral public keys are unauthenticated, whereas static public keys are authenticated by inclusion in certificates issued by a certification authority. The Diffie-Hellman protocol is called *static-static* if both X and Y are static; this protocol has the drawback that the shared secret σ is fixed for the lifetime of the static keys. The downside of the *ephemeral-ephemeral* Diffie-Hellman protocol, in which both X and Y are ephemeral, is that σ is unauthenticated. However, the ephemeral-ephemeral protocol is appealing because it achieves perfect forward secrecy — the leakage of any long-term secret keys does not give an attacker any advantage in learning session keys that were previously established. Another Diffie-Hellman variant is *ephemeral-static* where the initiator's public key is ephemeral and the responder's public key is static; this variant is useful in situations like email where the recipient may not be online and therefore cannot contribute an ephemeral public key. Many Diffie-Hellman protocols have also been proposed in which the communicating parties exchange both static and ephemeral public keys which are then combined to yield an authenticated session key; for example, see [19, 17, 14].

Some Diffie-Hellman protocols implicitly or explicitly require that an ephemeral key pair can be used in only one session. For example, the ANSI X9.42 [2] standard which specifies several Diffie-Hellman protocols

states that an ephemeral key is a "private or public key that is unique for each execution of a cryptographic scheme. An ephemeral private key is to be destroyed as soon as computational need for it is complete." Other protocols do not place any restrictions on the reuse of ephemeral keys. For example, the SIGMA protocol [13], which is the basis for the signature-based modes of the IKE (versions 1 and 2) protocols, allows for the reuse of an ephemeral public key "by the same party across different sessions", and this advice is followed in the IKEv2 standard [12, Section 2.12].

The primary reason for reusing ephemeral public keys is to increase efficiency by reducing the number of costly exponentiations a party has to perform. Another reason is to mitigate against denial-of-service (DoS) attacks. For example the JFKi protocol [1] allows a responder to reuse an ephemeral public key, whereby the responder does not have to do any expensive computations in its first response and only performs costly exponentiations after receiving the second message from the session's initiator. Since the initiator has to perform costly cryptographic operations when preparing the second message, this protocol offers the responder some resistance to DoS attacks. The authors of [1] go on to say that it is inadvisable *not* to reuse ephemeral public keys "in times of high load (or attack)."

In this paper we examine the vulnerability of some standardized Diffie-Hellman protocols to 'small-subgroup attacks' in the case where ephemeral keys are reused but public keys are not validated as being non-identity elements in the group G. A brief overview of small-subgroup attacks is given in §2. In §3 we show that in some scenarios the ephemeral-static Diffie-Hellman protocol in S/MIME is vulnerable to such attacks. In §4 we consider the version of the HMQV protocol that has been proposed for standardization by IEEE P1363. We conclude with some remarks in §5.

2. Small-subgroup attacks

Lim and Lee [18] showed that some discrete logarithm protocols may be vulnerable to small-subgroup attacks if the recipient of a public key does not *validate* it, i.e., verify that the public key is an element of the group $G = \langle g \rangle$ and is not equal to 1. In this section we describe a well-known small-subgroup attack on the static-static Diffie-Hellman protocol whereby parties $\hat{A}$ and $\hat{B}$ exchange their static public keys $A = g^a$ and $B = g^b$, and thereafter compute a session key $K = H(\sigma)$ where $\sigma = A^b = B^a = g^{ab}$ and H is a hash function.

Let G' be a group such that:

(1) The elements of G' are represented in the same format as elements of G (e.g., bitstrings of the same length).
(2) A conventional implementation of the multiplication operation for elements of G can also be used to multiply elements of G'.
(3) G' has an element γ of order t, where t is small enough so that an attacker can feasibly perform t operations (e.g., $t < 2^{40}$).

Suppose that neither $\hat{A}$ nor the certification authority verify that static public keys are non-identity elements of G. The attacker, party $\hat{B}$, selects an invalid static public key $B = \gamma$. Upon receiving B, party $\hat{A}$ computes $K = H(\sigma)$ where $\sigma = B^a = \gamma^a$. Suppose now that $\hat{A}$ subsequently sends $\hat{B}$ an authenticated message (m, T) where $T = \mathrm{MAC}_K(m)$. Then $\hat{B}$ iteratively computes $K' = H(\gamma^c)$ and $T' = \mathrm{MAC}_{K'}(m)$ for $c = 0, 1, 2, \ldots$ until $T' = T$, in which case $c = a \bmod t$ with high probability. By repeating this procedure for several orders t that are pairwise relatively prime (and possibly different groups G'), $\hat{B}$ can efficiently determine $\hat{A}$'s static private key a by the Chinese Remainder Theorem.

We next consider the applicability of such small-subgroup attacks in three groups that are widely used in practice.

2.1. **DSA groups.** As in DSA, G is the subgroup of prime order q of the multiplicative group G' of a prime field $\mathbb{Z}_p$. The bitlength of q is substantially smaller than that of p. For example, the bitlengths of q and p may be 160 and 1024 respectively, or 256 and 3072 respectively. Since the group parameters are typically generated by randomly selecting a prime q, and then randomly selecting even integers k of the appropriate bitlength until $p = 1+kq$ is prime, one can expect with non-negligible probability (see [4]) that $(p-1)/q$ will have a smooth divisor greater than q. Hence DSA groups are vulnerable to small-subgroup attacks if the recipient of a public key B does not verify that $B \in G \setminus \{1\}$; this validation can be accomplished by checking that B is an integer in the interval $[2, p-1]$ and that $B^q = 1$. Another countermeasure against these attacks is to select domain parameters p and q so that $(p-1)/q$ does not have any small prime factors.

2.2. **Safe prime groups.** In these groups, G is the subgroup of prime order q of the multiplicative group of a prime field $\mathbb{Z}_p$, where $p = 2q + 1$. If $\hat{A}$ checks that the public key B is an integer in the interval $[2, p-2]$ then, since the multiplicative group of $\mathbb{Z}_p$ has order $2q$, attacks like the ones described above can reveal at most a single bit of a. Thus small-subgroup attacks are not effective in the case of safe prime groups.

2.3. **Elliptic curve groups.** Let $E : V^2 = U^3 + \alpha U + \beta$ be an elliptic curve of prime order q defined over a prime field $\mathbb{F}_p$, and let $G = E(\mathbb{F}_p)$. In [5] (see also [3]) it was observed that the usual formulae for adding points in $E(\mathbb{F}_p)$ do not explicitly depend on the coefficient β. Hence $\hat{B}$ could launch the small-subgroup attack described above by selecting $B \in G' = E'(\mathbb{F}_p)$, where $E' : V^2 = U^3 + \alpha U + \beta'$ is an elliptic curve whose order is divisible by a relatively small prime factor t. $\hat{A}$ can easily thwart the attack by checking that B is a point in $E(\mathbb{F}_p)$ (and is not the point at infinity).

3. S/MIME

S/MIME (version 3.1) is an IETF standard for securing email [20, 11, 10]. S/MIME can be used to provide several security services including confidentiality, data integrity, data origin authentication, and non-repudiation, and allows the use of various public-key cryptographic algorithms including RSA, DSA and Diffie-Hellman. In this section we examine the vulnerability to small-subgroup attacks of an encryption-only mode in S/MIME (called "Enveloped-only" in [20]) when ephemeral-static Diffie-Hellman (as described in [21]) is used for key agreement.

We first give a description of the S/MIME encryption-only mode. Let $G = \langle g \rangle$ be a DSA subgroup, i.e., G is a subgroup of order q of $G' = \mathbb{Z}_p^*$, where the bitlength of q is substantially smaller than that of p. Let Enc and Dec denote the encryption and decryption functions for a symmetric-key encryption scheme such as Triple-DES, and let H be a hash function. Party $\hat{A}$ encrypts an email m for $\hat{B}$ as follows:

(1) Obtain an authentic copy of $\hat{B}$'s static public key $B = g^b$.
(2) Select an ephemeral private key $x \in_R [1, q-1]$ and compute the ephemeral public key $X = g^x$ and the session key $K = H(\sigma)$ where $\sigma = B^x = g^{bx}$.
(3) Select at random a content-encryption key k for the symmetric-key encryption scheme.
(4) Compute a 64-bit checksum v for k, and compute $c_1 = \mathrm{Enc}_K(k \,\|\, v)$.[1]
(5) Compute $c_2 = \mathrm{Enc}_k(m)$.
(6) Send X, c_1, c_2 to $\hat{B}$.

Upon receiving (X, c_1, c_2), party $\hat{B}$ computes $K = H(X^b)$, decrypts c_1, and verifies that the checksum for the recovered key k is correct. If so, then $\hat{B}$ decrypts c_2 to obtain m.

The S/MIME standards allow a party $\hat{A}$ to reuse an ephemeral key pair (x, X) for an unspecified period of time. For example, [21, Section 2.3] (see also [10, Section 4.1.1]) advises that some additional data, such as a counter, be appended to σ prior to hashing to ensure that a different session key is generated even if the ephemeral key is being reused: "If, however, the same ephemeral sender key is used for multiple messages (e.g. it is cached as a performance optimization) then a separate partyAInfo MUST be used for each message." However, while [21] recommends that the recipient $\hat{B}$ validate the received public key X in order to protect against small-subgroup attacks that aim to learn its static private key b, there is no requirement that the sender $\hat{A}$ validate the recipient's static public key B. Section 2.1 of [24] explains that this protection is not necessary because a dishonest party $\hat{B}$ who learns x via a small-subgroup attack isn't able to mount any

[1] More precisely, the checksum v consists of the 64 most significant bits of SHA-1(k), and $k \,\|\, v$ is encrypted twice in succession using the CBC mode of encryption [9].

interesting attacks because "the key is ephemeral and only associated with a message that the recipient can already decrypt..."

Our observation is that interesting small-subgroup attacks *can* be mounted in the situation where $\hat{A}$ reuses (x, X) to encrypt different messages to more than one user. A dishonest party $\hat{B}$ chooses its static public key $B = \gamma$ to be an element of small order t in $\mathbb{Z}_p^*$. Upon receiving (X, c_1, c_2), $\hat{B}$ iteratively computes $K' = H(\gamma^c)$ and decrypts c_1 using K' for $c = 0, 1, 2, \ldots$ until he obtains a valid checksum. He then learns that $c = x \bmod t$, and can repeat this procedure for pairwise relatively prime t to recover x. Hereafter, $\hat{B}$ is able to decrypt messages that $\hat{A}$ encrypts using x for other (honest) parties.

As mentioned in §2.1 such attacks can be thwarted by validating static public keys, or by selecting DSA parameters p, q so that $(p-1)/q$ does not have any small prime factors.

4. HMQV

HMQV [14, 15] is an authenticated Diffie-Hellman key agreement protocol. In this section, we consider the version of the HMQV protocol that has been proposed for standardization by P1363 [16].

Let $G = \langle g \rangle$ be a DSA subgroup, i.e., G is a subgroup of order q of $G' = \mathbb{Z}_p^*$, where the bitlength of q is substantially smaller than that of p. Let H be a hash function, and let $\overline{H}$ be an l-bit hash function where $l = (\lfloor \log_2 q \rfloor + 1)/2$. Let $A = g^a$ and $B = g^b$ denote the static public keys of parties $\hat{A}$ and $\hat{B}$ respectively.

The HMQV protocol is the following:

(1) The initiator $\hat{A}$ selects an ephemeral public key $X = g^x$ and sends A and X to $\hat{B}$.
(2) Upon receiving (A, X), $\hat{B}$ does the following:
   (a) Verify that $A, X \in G'$ and $A, X \neq 1$.
   (b) Select an ephemeral public key $Y = g^y$.
   (c) Compute $d = \overline{H}(X, \hat{B})$, $e = \overline{H}(Y, \hat{A})$, and $r = y + eb \bmod q$.
   (d) Compute $\sigma = (XA^d)^r$ and $K = H(\sigma)$.
   (e) Send B and Y to $\hat{A}$.
(3) Upon receiving (B, Y), $\hat{A}$ does the following:
   (a) Verify that $B, Y \in G'$ and $B, Y \neq 1$.
   (b) Compute $d = \overline{H}(X, \hat{B})$, $e = \overline{H}(Y, \hat{A})$, and $s = x + da \bmod q$.
   (c) Compute $\sigma = (YB^e)^s$ and $K = H(\sigma)$.

If all checks pass, then $\hat{A}$ and $\hat{B}$ share the session key $K = H(\sigma)$ where $\sigma = g^{sr} = g^{(x+da)(y+eb)}$.

HMQV has been proven secure in the Canetti-Krawczyk model for key exchange [7] under the assumptions that H and $\overline{H}$ are random functions, and the computational Diffie-Hellman problem in G is intractable [14, 15]. The security proof assumes that ephemeral keys are used only once, and that ephemeral private keys are securely destroyed after they have been used. However, in practice it may be tempting (in the absence of explicit

warnings) for an implementer to allow for the reuse of ephemeral public keys in order to improve performance. We next show that if party $\hat{A}$ decides to reuse an ephemeral key pair (x, X), then party $\hat{B}$ may be able to mount a devastating attack — one that reveals $\hat{A}$'s static private key a.

Our attack assumes that $(p-1)/q$ has several small (e.g. less than 2^{40}) pairwise relatively prime factors whose product is greater than q; let t be one such factor. The attacker $\hat{B}$ obtains a certificate for his (valid) static public key $B = g^b$. Upon receiving (A, X) from $\hat{A}$, the attacker selects an ephemeral public key $Y \in G'$ of order t, and sends (B, Y) to $\hat{A}$. The public keys B and Y satisfy the checks in step 3(a), so $\hat{A}$ computes $\sigma = (YB^e)^s$ and $K = H(\sigma)$ where $s = x + da \bmod q$, $d = \overline{H}(X, \hat{B})$, and $e = \overline{H}(Y, \hat{A})$. Suppose now that $\hat{A}$ sends $\hat{B}$ an authenticated message $(m, T = \mathrm{MAC}_K(m))$. Note that

$$\sigma = (YB^e)^s = Y^s B^{es} = Y^s (g^s)^{be} = Y^s (XA^d)^{be}.$$

Hence $\hat{B}$ iteratively computes $K' = H(Y^c (XA^d)^{be})$ and $T' = \mathrm{MAC}_{K'}(m)$ for $c = 0, 1, 2, \ldots$ until $T' = T$ in which case $c = s \bmod t$. After repeating this procedure for several pairwise relatively prime orders t, $\hat{B}$ can determine s by the Chinese Remainder Theorem.

Party $\hat{B}$ now repeats the attack using a different identifier $\hat{B}'$ (or perhaps by colluding with a third party $\hat{C}$). Consequently $\hat{B}$ learns $s' = x + d'a \bmod q$, where $d' = \overline{H}(X, \hat{B}')$. Since $d' \neq d$ with very high probability, $\hat{B}$ can then compute $a = (s - s')(d - d')^{-1} \bmod q$.

Our attack is possible because $\hat{A}$ is not required to validate public keys. Omission of validation is intentional in HMQV in order to improve performance — the only checks required on B and Y are that they belong to $G' \setminus \{1\}$. However, we emphasize that the attack *cannot* be mounted on the version of HMQV as described in [16], since that version of the protocol assumes that ephemeral public keys are never reused. Rather, the attack highlights the danger of reusing ephemeral public keys in a key agreement protocol for which the security analysis assumed that ephemeral public keys are never reused.

5. Concluding remarks

As mentioned in §1, the IKEv2 protocol [12] allows for the reuse of ephemeral private keys. However, the protocol is immune to the small-subgroup attack described in §3 because IKEv2 requires the use of safe prime groups [22]. More recently, elliptic curve groups have been proposed for IKE and IKEv2 [8, 6]. To prevent invalid-curve attacks analogous to the small-subgroup attack described in §3, a sender who reuses an ephemeral key pair to encrypt different messages for more than one party should validate the recipients' static public keys.

The recent NIST SP 800-56A [23] standard for key agreement explicitly disallows the reuse of ephemeral keys with one exception — a sender may reuse an ephemeral key if the resulting session key is used to transport the

same keying material, and if all these transactions occur "simultaneously" (or within a short period of time). Since SP 800-56A mandates that all public keys be validated, this reuse of ephemeral public keys appears to be sound. It would be a useful exercise to capture the reuse of ephemeral keys in the Canetti-Krawczyk security model (as outlined in [1]) and formally verify the belief that reusing ephemeral keys as allowed in SP 800-56A does not introduce any security weaknesses.

References

[1] W. Aiello, S. Bellovin, M. Blaze, R. Canetti, J. Ionnidis, A. Keromytis and O. Reingold, "Just fast keying: key agreement in a hostile Internet", *ACM Transactions on Information and Systems Security*, 7/2 (2004), 1-30.

[2] ANSI X9.42, *Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm 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] W. Banks and I. Shparlinski, "Integers with a large smooth divisor", *Integers: Electronic Journal of Combinatorial Number Theory*, 7 (2007), #A17.

[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] D. Brown, "Additional ECC groups for IKE and IKEv2", Internet draft, IPSec working group, October 10 2006.

[7] 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/.

[8] D. Fu and J. Solinas, "ECP groups for IKE and IKEv2", RFC 4753, Internet Engineering Task Force, 2007.

[9] R. Housley, "Triple-DES and RC2 key wrapping", RFC 3217, Internet Engineering Task Force, 2001.

[10] R. Housley, "Cryptographic message syntax (CMS) algorithms", RFC 3370, Internet Engineering Task Force, 2002.

[11] R. Housley, "Cryptographic message syntax (CMS)", RFC 3852, Internet Engineering Task Force, 2004.

[12] C. Kaufman, "Internet key exchange (IKEv2) protocol", RFC 4306, Internet Engineering Task Force, 2005.

[13] H. Krawczyk, "SIGMA: the 'SIGn-and-MAc' approach to authenticated Diffie-Hellman and its use in the IKE protocols", *Advances in Cryptology – CRYPTO 2003*, Lecture Notes in Computer Science, 2729 (2003), 400-425.

[14] H. Krawczyk, "HMQV: A high-performance secure Diffie-Hellman protocol", *Advances in Cryptology – CRYPTO 2005*, Lecture Notes in Computer Science, 3621 (2005), 546-566.

[15] H. Krawczyk, "HMQV: A high-performance secure Diffie-Hellman protocol", Full version of [14], available at http://eprint.iacr.org/2005/176/.

[16] 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.

[17] 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.
[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] T. Matsumoto, Y. Takashima and H. Imai, "On seeking smart public-key distribution systems", *The Transactions of the IECE of Japan*, E69 (1986), 99-106.
[20] B. Ramsdell, editor,, "Secure/multipurpose internet mail extensions (S/MIME) version 3.1 – Message specification", RFC 3851, Internet Engineering Task Force, 2004.
[21] E. Rescorla, "Diffie-Hellman key agreement method", RFC 2631, Internet Engineering Task Force, 1999.
[22] J. Schiller, "Cryptographic algorithms for use in the Internet Key Exchange Version 2 (IKEv2)", RFC 4307, Internet Engineering Task Force, 2005.
[23] 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.
[24] R. Zuccherato, "Methods for avoiding the "small-subgroup" attacks on the Diffie-Hellman key agreement method for S/MIME", RFC 2785, Internet Engineering Task Force, 2000.

Department of Combinatorics & Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada
E-mail address: ajmeneze@uwaterloo.ca

Department of Combinatorics & Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada
E-mail address: bustaoglu@uwaterloo.ca