Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol

Simon Blake-Wilson[1] and Alfred Menezes[2]

[1] Certicom Research, 200 Matheson Blvd. W., Suite 103
Mississauga, Ontario, Canada L5R 3L7
sblakewi@certicom.com
[2] Department of Combinatorics & Optimization
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
ajmeneze@cacr.math.uwaterloo.ca

Abstract. This paper presents some new unknown key-share attacks on STS-MAC, the version of the STS key agreement protocol which uses a MAC algorithm to provide key confirmation. Various methods are considered for preventing the attacks.

1 Introduction

Key establishment is the process by which two (or more) entities establish a shared secret key. The key may subsequently be used to achieve some cryptographic goal, such as confidentiality or data integrity. Ideally, the established key should have precisely the same attributes as a key established face-to-face — for example, it should be shared by the (two) specified entities, it should be distributed uniformly at random from the key space, and no unauthorized (and computationally bounded) entity should learn anything about the key.

Key establishment protocols come in various flavors. In *key transport* protocols, a key is created by one entity and securely transmitted to the second entity, while in *key agreement* protocols both parties contribute information which is used to derive the shared secret key. In *symmetric* protocols the two entities a priori possess common secret information, while in *asymmetric* protocols the two entities share only public information that has been authenticated. This paper is concerned with two-party key agreement protocols in the asymmetric setting.

Unfortunately, the requirement that key agreement protocols have the same properties as face-to-face key establishment is too vague to be much help to protocol designers, who instead focus on designing protocols to meet more explicit requirements. Implicit key authentication and key confirmation are two explicit requirements that are often considered essential.

Let A and B be two honest entities, i.e., legitimate entities who execute the steps of a protocol correctly. Informally speaking, a key agreement protocol is said to provide *implicit key authentication* (of B to A) if entity A is assured that no other entity aside from a specifically identified second entity B can possibly learn the value of a particular secret key. Note that the property of

H. Imai and Y. Zheng (Eds.): PKC'99, LNCS 1560, pp. 154–170, 1999.

implicit key authentication does not necessarily mean that A is assured of B actually possessing the key. A key agreement protocol which provides implicit key authentication to both participating entities is called an *authenticated key agreement (AK)* protocol.

Informally speaking, a key agreement protocol is said to provide *explicit key confirmation* (of B to A) if entity A is assured that the second entity B *has actually* computed the agreed key. The protocol provides *implicit key confirmation* if A is assured that B *can* compute the agreed key. While explicit key confirmation appears to provide stronger assurances to A than implicit key confirmation (in particular, the former implies the latter), it appears that, for all practical purposes, the assurances are in fact the same. That is, the assurance that A requires in practice is merely that B can compute the key rather than that B has actually computed the key. Indeed in practice, even if a protocol does provide explicit key confirmation, it cannot guarantee to A that B will not lose the key between key establishment and key use. Thus it would indeed seem that implicit key confirmation and explicit key confirmation are in practice very similar.

If both implicit key authentication and (implicit or explicit) key confirmation (of B to A) are provided, then the key establishment protocol is said to provide *explicit key authentication* (of B to A). A key agreement protocol which provides explicit key authentication to both participating entities is called an *authenticated key agreement with key confirmation (AKC)* protocol.

In addition to implicit key authentication and key confirmation, a number of other desirable security attributes of key agreement protocols have been identified including known-key security, forward secrecy, key-compromise impersonation, and unknown key-share. These are typically properties possessed by face-to-face key establishment which may be more or less important when a key establishment protocol is used to provide security in real-life applications.

An *unknown key-share* (UKS) attack on an AK or AKC protocol is an attack whereby an entity A ends up believing she shares a key with B, and although this is in fact the case, B mistakenly believes the key is instead shared with an entity $E \neq A$. The significance of UKS attacks on AK and AKC protocols is further discussed in §3.

This paper presents some new on-line UKS attacks on STS-MAC, the variant of the station-to-station (STS) [11] AKC protocol which uses a MAC to provide key confirmation. For an extensive survey on key establishment, see Chapter 12 of [25]. For a recent survey on authenticated Diffie-Hellman key agreement protocols, see [10]. Formal definitions of authenticated key agreement can be found for the symmetric setting in [7] and for the asymmetric setting in [9].

The remainder of this paper is organized as follows. The STS protocol is described in §2. In §3 we present the new on-line UKS attacks on STS-MAC, and consider ways of preventing the attacks. In §4, we examine the plausibility of an assumption regarding signature schemes that is required in order for the attacks to succeed. §5 makes concluding remarks.