Breaking the Simple Authenticated Key Agreement (SAKA) protocol

Chris J. Mitchell

Technical Report
RHUL–MA–2001–2
18 August 2001

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
http://www.rhul.ac.uk/mathematics/techreports

Abstract

An active attack against a key agreement protocol based on a shared password is described[1]. If poorly chosen, as passwords often are, the password can be compromised by a simple brute force search.

1 Introduction

A recent paper by Seo and Sweeney, [3], describes *SAKA*, a method for 'authenticated key agreement' based on a shared password. The scheme is presented as an alternative to the schemes of Bellovin and Merritt, [2], and Anderson and Lomas, [1]. These latter schemes also enable a key to be set up using a shared secret password and have been carefully designed to prevent exhaustive searches for poorly chosen passwords.

Unfortunately, unlike the protocols of [1, 2], the SAKA protocol does nothing to protect the password against an active guessing attack. A cryptanalyst can engage in the protocol, masquerading as a genuine party, and then guess the password by attempting to decrypt a message subsequently sent with the agreed key.

2 Details of attack

In the protocol as described, A and B generate a shared secret key as follows. A chooses a random a and sends B the value $g^{aQ} \bmod n$, where g is a public Diffie-Hellman 'base' modulo n (a public prime), and where Q is the shared password. B chooses a random b and sends A the value $g^{bQ} \bmod n$. A and B can then both compute $g^{ab} \bmod n$ (using knowledge of $Q^{-1} \bmod n-1$).

Suppose C impersonates A to B in the above protocol and sends $X = g^c \bmod n$ to B (for a random c). B then sends $Y = g^{bQ} \bmod n$ to C (thinking he is talking to A). B computes the shared secret key as $K = X^{bQ^{-1}} \bmod n = g^{bcQ^{-1}} \bmod n$. C cannot compute the shared key but knows it will equal $Y^{cQ^{-2}} \bmod n$.

Now suppose B encrypts the message M using K, and suppose also that C knows part of M (knowing M consists of a string of 8-bit ASCII characters will typically be sufficient). If C knows the password Q is poorly chosen, then C simply works though all possible passwords Q, computes $K^* = Y^{cQ^{-2}} \bmod n$ for each candidate, and uses K^* to decrypt the encrypted message. If the

[1] This report was originally written in July 1999

result has elements which match the parts of M known by C, then C has discovered the password Q.

3 Conclusion

Contrary to the main purpose of the SAKA protocol, we have shown that it is subject to a simple password search if the attacker can conduct an active attack.

References

[1] R.J. Anderson and T.M.A. Lomas. Fortifying key negotiation schemes with poorly chosen passwords. *Electronics Letters*, **30**:1040–1041, 1994.

[2] S.M. Bellovin and M. Merritt. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In *Proceedings: 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 72–84. IEEE Computer Society Press, Los Alamitos, California, May 1992.

[3] D.H. Seo and P. Sweeney. Simple authenticated key agreement protocol. *Electronics Letters*, **35**:1073–1074, 1999.