Efficient Mutual Data Authentication Using Manually Authenticated Strings

Sven Laur[2] and Kaisa Nyberg[1,2]

[1] Nokia Research Center, Finland
kaisa.nyberg@nokia.com
[2] Helsinki University of Technology, Finland
{slaur, knyberg}@tcs.hut.fi

Abstract. Solutions for an easy and secure setup of a wireless connection between two devices are urgently needed for WLAN, Wireless USB, Bluetooth and similar standards for short range wireless communication. All such key exchange protocols employ data authentication as an unavoidable subtask. As a solution, we propose an asymptotically optimal protocol family for data authentication that uses short manually authenticated out-of-band messages. Compared to previous articles by Vaudenay and Pasini the results of this paper are more general and based on weaker security assumptions. In addition to providing security proofs for our protocols, we focus also on implementation details and propose practically secure and efficient sub-primitives for applications.

1 Introduction

In this paper we consider the problem of setting up a shared secret key in an ad hoc manner, that is, in isolation from any key management facility and without pre-shared secrets. Consider two parties Alice and Bob who want to establish a shared secret key over an insecure network without any prior authenticated information. If adversaries are passive, that is, no malicious messages are sent to the network and all messages are delivered unaltered, then exchanging public keys for Diffie-Hellman or similar public key based key exchange protocol is sufficient. However, an active adversary, Charlie, can launch an man-in-the-middle attack. Namely, Charlie can replace a desired secure channel from Alice to Bob by a pair of secure channels, one from Alice to Charlie and one from Charlie to Bob. The attack is transparent to legitimate users without prior authenticated information. Thus secure key exchange is impossible without authenticated channels. The main question is how much information, *authentic out-of-band messages (OOB messages)*, must be sent over the authenticated channel to achieve reasonable security level. We are aiming at an application, where keys are exchanged between various electronic devices and authentic communication is done by an ordinary user who either enters messages into devices or compares output displays. The latter severely limits a plausible size of OOB messages: one could consider 4–6 decimal digits as optimal and 16 hexadecimal characters as an absolute limit. Other possible OOB channels include various visual or audible signals like blinking lights, images, phone calls etc.

Most urgently such a solution is needed for WLAN: the current use of pre-shared keys degrades both practical usability and security. The home users should have a clear

D. Pointcheval, Y. Mu, and K. Chen (Eds.): CANS 2006, LNCS 4301, pp. 90–107, 2006.

and manageable procedure to set up a secure wireless network so that it is easy to add and remove devices from the network. Hence, the WiFi Alliance is working on a better solution. Recently, manual data authentication using short authenticated strings received practical applications in ad hoc key agreement. Phil Zimmermann released a software called Zfone and an Internet draft to offer security to Voice over IP [ZJC06]. A similar protocol (See Protocol 3) was adopted by USB-IF for Wireless USB devices [WUS06] and manual data authentication is going to be incorporated into Bluetooth [BT06].

A formal security model for such protocols consists of three bidirectional asynchronous channels, where messages can be arbitrarily delayed. In-band communication is routed from Alice to Bob via an active adversary Charlie, who can drop, modify or insert messages. The out-of-band channel between Alice and Bob is authentic but has low bandwidth and Charlie can arbitrarily delay[1] OOB messages. The model captures nicely all threats in wireless environment, as malicious adversary with a proper equipment can indeed change the network topology and thus reroute, drop, insert and modify messages. However, security is not the only objective. User-friendliness, low resource consumption and simple setup assumptions are equally important. There should be no public key infrastructure, as it is almost impossible to guarantee authenticity and availability of public keys to the humongous number of electronic devices. Also, protocols should use only symmetric primitives if possible.

All currently known user-aided key exchange and data authentication protocols can be divided into two different groups: protocols with authenticated but public OOB messages [Hoe05, CCH06, Vau05, LAN05, PV06a, PV06b, NSS06] and protocols with confidential passwords. Password-protected key exchange, see [BM92, KOY01] and Mana III in [GMN04], is needed when a user wants to establish a secure connection between devices that have input only, for example, devices with keyboards but no display. The main application for the manual data authentication is also a cryptographically secure but still user-friendly ad hoc key agreement between two or more network devices.

Our contribution. In this paper, we clarify and extend our preliminary results [LAN05]. In particular, we show that the previously presented manual cross authentication protocols [LAN05, PV06b] are indeed instantiations of the same protocol family that uses a commitment scheme to temporarily hide a secret key needed for data authentication. Compared to the results by Pasini and Vaudenay [PV06b], our security proofs (Sec. 4) are more modular and assumptions on used primitives are weaker and geared towards practice. We explicitly consider implementation details, that is, how to choose practical primitives (Sec. 5). Given a data authentication protocol it can be combined with the Diffie-Hellman key agreement in a secure way by taking the Diffie-Hellman key, or the pair of the public keys, as the data to be authenticated. But the designers of the practical protocols from [ZJC06, WUS06] have taken a different approach by using the Diffie-Hellman key shares as the source of randomness. In Sec. 3, we extend our proof of security also for such a case. In App. A, we consider security in any computational context and show that, under reasonable assumptions, security is not abruptly degraded if several protocols are executed in parallel. As an important theoretical result, we show

[1] For example, the adversary can distract the user who compares the output of two devices.