An Elliptic Curve Backdoor Algorithm for RSASSA

Adam Young[1] and Moti Yung[2]

[1] Cryptovirology Labs
aly@cryptovirology.com
[2] RSA Labs and Columbia University
moti@cs.columbia.edu

Abstract. We present the first (1,2)-SETUP algorithm for the RSA digital signature scheme with appendix. A SETUP algorithm C' is an algorithmic modification of algorithm C that (1) contains an asymmetric backdoor that can only be used by the designer, even if the backdoor algorithm is fully public, and (2) ensures that the public outputs of C and C' are computationally indistinguishable under black-box queries. The SETUP is presented in RSASSA-PSS and it transmits the RSA private key within two w.l.o.g consecutive digital signatures. This problem has been solved for DSA and other discrete-log based digital signature algorithms, but not RSA. We therefore solve a long-standing problem in kleptography.

1 Introduction

There has been a lot of research into designing backdoors into key generation algorithms, encryption algorithms, key exchanges, and digital signature schemes. Such backdoors are dual-edged in nature. When deployed by an honest key recovery agent, they can be used to enable an organization to have timely access to private keys. However, when deployed by dishonest recovery agents, they can be used to surreptitiously access private information.

In this paper we continue the line of research that seeks to design asymmetric backdoors in digital signing algorithms. More specifically, we present the first high-bandwidth asymmetric backdoor for the RSASSA-PSS digital signature scheme that is defined in PKCS #1 [21]. An asymmetric backdoor is a covert backdoor that can only be utilized by the designer that deploys it, even when the entire backdoor algorithm is made public. RSASSA-PSS is based on the RSA signature algorithm [23], except that it is probabilistic and incorporates a nonce in each digital signature that is output. RSASSA has a formal proof of security and it appears in [4,5].

The backdoor that we present is non-trivial since the subliminal channel in RSASSA is rather small. For example, when SHA-1 [11,12] is used and the salt length is 20 bytes, the subliminal channel is also 20 bytes (the channel is the salt). The RSA function is a deterministic permutation and it is therefore challenging

J. Camenisch et al. (Eds.): IH 2006, LNCS 4437, pp. 355–374, 2007.

to conduct subliminal communication using RSA after the RSA key pair has already been generated.

Previous work on designing a backdoor into RSA key generation includes Anderson's construction [2]. Later the notion of an asymmetric backdoor (that can only be used by the designer even when the device is fully reverse-engineered and all "secrets" are learned) was introduced [26,27]. Related work includes [9]. It is worth pointing out that efforts have primarily focused on designing backdoors in RSA key generation as opposed to RSA signing.

The notion of (1,2)-leakage bandwidth in kleptographic attacks was put forth in [27]. A (m, n)-leakage scheme is an asymmetric backdoor (SETUP mechanism) that leaks m keys/secret messages over n keys/messages that are output by the cryptographic device ($m \leq n$). A (1,2)-leakage scheme was presented in DSA [28] that leaks the signing private key over the course of two w.l.o.g consecutive DSA signatures. Other work includes [29] that presents a method for transmitting a 20-bit asymmetrically encrypted message covertly over a single RSASSA-PSS signature. However, no RSA digital signature backdoors to date achieve anywhere near the capacity of what we achieve, namely, the secure (asymmetric) and subliminal transmission of the RSA private key over two RSA signatures. Work that is related to this includes a study of subliminal-freeness in the nonce-devoid version of RSA-PSS [6].

The significance of our high-bandwidth backdoor in RSA signing is as follows. Our backdoor algorithm makes it possible to generate an RSA key pair normally (i.e., no backdoor involved), load it into a smartcard, and then leak the RSA private key securely and subliminally through two RSA signatures. This is not possible in any of the previous RSA backdoor designs (this includes key generation backdoors and backdoors in RSA signing algorithms).

Our backdoor exploits in a constructive and forward-engineering fashion the recent cryptanalysis of both MD5 and SHA-1 [24,25]. In short, we argue that the collapse of these primitives is a boon for *information hiding* since the traditional security parameter of $k = 1024$ for RSA *has stayed the same* yet the range of hash functions that are deemed secure has increased significantly (i.e., the move from SHA-1 to SHA-224 and higher). The backdoor we present exploits this shift in the size of contemporary security constants. Just how this shift is exploited is covered in Section 8.

2 Definition of a (1,2)-SETUP

The notion of a secretly embedded trapdoor with universal protection (SETUP) was put forth in [26]. In short, a SETUP is an algorithmic modification made to cryptosystem C to derive cryptosystem C' that leaks secret key bits to the cryptosystem designer. A SETUP has the properties that: (1) even if the entire implementation of C' becomes public, the backdoor can still only be used by the designer, and (2) the public outputs of C and C' are polynomially indistinguishable under black-box queries. The exclusive use of the backdoor by the designer