A Key Substitution Attack on SFLASH^{v3}

Willi Geiselmann and Rainer Steinwandt

IAKS, Arbeitsgruppe Systemsicherheit, Prof. Dr. Th. Beth
Fakultät für Informatik, Universität Karlsruhe,
Am Fasanengarten 5, 76 131 Karlsruhe, Germany

Abstract

A practical key substitution attack on SFLASH^{v3} is described: Given a valid (message, signature) pair (m, σ) for some public key v_0, one can derive another public key v_1 (along with matching secret data) such that (m, σ) is also valid for v_1. The computational effort needed for finding such a 'duplicate' key is comparable to the effort needed for ordinary key generation.

1 Introduction

SFLASH^{v2} is one of the asymmetric signature algorithms that are part of the NESSIE Portfolio of recommended cryptographic primitives [4]. The successor SFLASH^{v3} introduces several changes in the algorithm: E. g., the way of using SHA-1 [8] during signing has been modified and—reflecting a comment [7] on an earlier version of the specification [5]—the at the time of writing latest specification [6] also makes use of a so-called *semi-public key*.

This contribution shows that SFLASH^{v3} is vulnerable to a so-called key substitution attack, which can be of interest in multi-user settings (see [1, 2]): Given a valid (message, signature) pair (m, σ) for a verification key v_0, one can efficiently derive another verification key v_1 (along with a matching secret key) such that (m, σ) is valid for v_1, too. After recalling the basic set-up of SFLASH^{v3} in the next section, we show that for this scheme the computational effort needed for deriving such a 'duplicate' key is comparable to the effort needed for creating an 'ordinary' key.

2 Signing and verifying in SFLASH^{v3}

For our purposes, it is not necessary to recall the detailed structure of SFLASH^{v3}, and we therefore give only a rough summary of the scheme; a complete specification can be found in [6].

SFLASH^{v3} makes use of two fields along with corresponding bijections

- $K := \mathbb{F}_2[X]/(X^7 + X + 1)$ along with the bijection

$$\begin{array}{rccl} \pi : & \{0,1\}^7 & \longrightarrow & K \\ & (b_0, \dots, b_6) & \longmapsto & \sum_{i=0}^{6} b_i X^i \pmod{X^7 + X + 1} \end{array}$$

- $L := K[X]/(X^{67} + X^5 + X^2 + X + 1)$ along with the bijection

$$\begin{array}{rccl} \varphi : & K^{67} & \longrightarrow & L \\ & (b_0, \dots, b_{66}) & \longmapsto & \sum_{i=0}^{66} b_i X^i \pmod{X^{67} + X^5 + X^2 + X + 1} \end{array}$$

2.1 Secret and semi-public key

The non-public part of the key is comprised of three parts:

- $\Delta \in \{0,1\}^{80}$: a secret 80-bit string
- $s = (S_L, S_C)$: an affine bijection $K^{67} \longrightarrow K^{67}$ given by a secret 67×67 matrix $S_L \in K^{67 \times 67}$ and a semi-public column vector $S_C \in K^{67}$
- $t = (T_L, T_C)$: an affine bijection $K^{67} \longrightarrow K^{67}$ given by a secret 67×67 matrix $T_L \in K^{67 \times 67}$ and a semi-public column vector $T_C \in K^{67}$

For deriving the corresponding public key, the function

$$\begin{array}{rccl} F : & L & \longrightarrow & L \\ & \alpha & \longmapsto & \alpha^{128^{33}+1} \end{array}$$

is used.

2.2 Public verification key

The public verification key is the function

$$G(x) = [(t \circ \varphi^{-1} \circ F \circ \varphi \circ s)(x)]_{0 \to 7 \cdot 56 - 1}.$$

Here the notation $[\cdot]_{0 \to 7 \cdot 56 - 1}$ means that only the first 56 (out of 67) rows are published,[1] and $\circ$ denotes functional composition, i. e., $(f \circ g)(x) := f(g(x))$.

[1] As one K-element corresponds to 7 bits, $[\cdot]_{0 \to 7 \cdot 56 - 1}$ translates into selecting the first 56 K-elements.

By construction, $(Y_0, \ldots, Y_{55}) = G(X_0, \ldots, X_{66})$ can be expressed in the form

$$\begin{aligned} Y_0 &= P_0(X_0, \ldots, X_{66}) \\ &\vdots \\ Y_{55} &= P_{55}(X_0, \ldots, X_{66}) \end{aligned}$$

where each P_i is a polynomial of total degree ≤ 2 with coefficients in K.

2.3 Computing and verifying signatures

Essentially, to sign a bitstring m, the following steps are performed:

1. Without involving any secret or semi-public data, a 392-bit string V is derived from m by means of SHA-1.

2. Via $Y := (\pi([V]_{0\to 6}), \pi([V]_{7\to 13}), \ldots, \pi([V]_{385\to 391}))$ the bitstring V is translated into a vector $Y \in K^{56}$, where the notation $[\cdot]_{a\to b}$ is to be understood as selecting the bits no. a–b.

3. Applying SHA-1 to the concatenation of V and Δ followed by reading off the first 77 bits of the hash value yields a bitstring $W =$ SHA-1$(V||\Delta)$. Via $R := (\pi([V]_{0\to 6}), \pi([V]_{7\to 13}), \ldots, \pi([V]_{70\to 76}))$ this bitstring is translated into an element $R \in K^{11}$.

4. By means of the secret and semi-public data now the value

$$X := (s^{-1} \circ \varphi^{-1} \circ F^{-1} \circ \varphi \circ t^{-1})(Y||R)$$

is computed, where $(Y||R) \in K^{67}$ denotes the concatenation of Y and R. Translating the 67 entries of X into a bitstring by means of π^{-1} yields the final (469-bit) signature σ of m.

To verify a signature σ' (of the correct length) of a bitstring m, one uses π to translate σ' into an element $X' \in K^{67}$. Evaluating the 56 public verification polynomials at X' yields an element $Y' \in K^{56}$. If Y' coincides with the value Y, that is derived from the bitstring m in the same manner as in the first two steps of the signing procedure, then the signature σ is accepted. Otherwise, σ is rejected.

3 A key substitution attack

Let (m, σ) be an arbitrary valid (message, signature) pair computed with some SFLASH^{v3} key. Then we can apply the following simple attack to derive another key which also yields the signature σ for m—knowing the 'original' verification key is not necessary:

- First generate an arbitrary private key (S_L, T_L, Δ) and an arbitrary semi-public $S_C \in K^{67}$. Let s be the affine bijection defined through S_L and S_C.
- Making use of Δ, now apply Step 1–3 of the signing procedure to the message m. Let $(Y||R) \in K^{67}$ be the concatenation of the resulting vectors Y and R.
- Next, as in the verification procedure, use π to translate the signature $\sigma \in \{0,1\}^{469}$ into a vector $X \in K^{67}$, and define

$$T_C := (Y||R) - T_L \cdot ((\varphi^{-1} \circ F \circ \varphi \circ s)(X)) \in K^{67}.$$

Denoting the affine bijection defined through T_L and T_C by t, by construction we now have

$$(t \circ \varphi^{-1} \circ F \circ \varphi \circ s)(X) = (Y||R).$$

In particular, (m, σ) is a valid (message, signature) pair for the public verification key corresponding to the secret/semi-public data (s, t, Δ). To derive this public verification key from (s, t, Δ) we can proceed as in the usual key generation.

4 Conclusion

The above discussion shows that the current specification of SFLASH^{v3} does not rule out a (practical) key substitution attack. Consequently, in multi-user settings where such attacks are of concern SFLASH^{v3} should not be used in the proposed form.

References

[1] A. Menezes ans N. Smart. Security of Signature Schemes in a Multi-User Setting. *Designs, Codes and Cryptography*, to appear. Available at `http://www.cacr.math.uwaterloo.ca/~ajmeneze/publications/signature.ps`.

[2] S. Blake-Wilson and A. Menezes. Unknown Key-Share Attacks on the Station-to-Station (STS) Protocol. In H. Imai and Y. Zheng, editors, *International Workshop on Practice and Theory in Public Key Cryptography—PKC '99*, volume 1560 of *Lecture Notes in Computer Science*, pages 154–170. Springer, 1999.

[3] R. Canetti. On Universally Composable Notions of Security for Signature, Certification and Authentication. Cryptology ePrint Archive: Report 2003/239, 2003. Available at http://eprint.iacr.org/2003/239/.

[4] NESSIE consortium. NESSIE Portfolio of recommended cryptographic primitives. Available at https://www.cosic.esat.kuleuven.ac.be/nessie/deliverables/decision-final.pdf, 2003.

[5] N. Courtois, L. Goubin, and J. Patarin. SFLASH^{v3}, a fast asymmetric signature scheme. Cryptology ePrint Archive: Report 2003/211, 2003. Revised Specification of SFLASH, version 3.0., October 2nd, 2003. Originally available at http://eprint.iacr.org/2003/211/.

[6] N. Courtois, L. Goubin, and J. Patarin. SFLASH^{v3}, a fast asymmetric signature scheme. Cryptology ePrint Archive: Report 2003/211, 2003. Revised Specification of SFLASH, version 3.0., October 17th, 2003. Available at http://eprint.iacr.org/2003/211/.

[7] W. Geiselmann and R. Steinwandt. A short comment on the affine parts of SFLASH^{v3}. Cryptology ePrint Archive: Report 2003/220, 2003. Available at http://eprint.iacr.org/2003/220/.

[8] U.S. Department of Commerce, National Institute of Standards and Technology. *FIPS PUB 180-1 SECURE HASH STANDARD*, April 1995. Available at http://csrc.nist.gov/publications/fips/fips180-1/fips180-1.pdf.