FLASH, a fast multivariate signature algorithm*

http://www.minrank.org/flash/

Jacques Patarin, Nicolas Courtois and Louis Goubin

Bull CP8
68 route de Versailles – BP45
78431 Louveciennes Cedex
France
J.Patarin@frlv.bull.fr,courtois@minrank.org, Louis.Goubin@bull.net

Abstract. This article describes the particular parameter choice and implementation details of one of the rare published, but not broken signature schemes, that allow signatures to be computed and checked by a low-cost smart card. The security is controversial, since we have no proof of security, but the best known attacks require more than 2^{80} computations. We called FLASH our algorithm and we also proposed SFLASH, a version that has a smaller public key and faster verification though one should be even more careful about it's security.
FLASH and SFLASH have been accepted as submissions to NESSIE (New European Schemes for Signatures, Integrity, and Encryption), a project within the Information Societies Technology (IST) Programme of the European Commission.

1 Introduction

In the present document, we describe the FLASH public key signature scheme.

FLASH is a C^{*--} algorithm (see [4]) with a special choice of the parameters. FLASH belongs to the family of "multivariate" public key schemes, *i.e.* each signature and each hash of the messages to sign are represented by some elements of a small finite field K.

FLASH is designed to be a very fast signature scheme, both for signature generation and signature verification. It is much faster in signature than RSA and much easier to implement on smart cards without any arithmetic coprocessor for example. However its public key size is larger than the public key size of RSA. Nevertheless this public key size can fit in current smart cards. It may also be noticed that, with the secret key, it is possible to sign AND to check the signature (generated with this particular secret key) without the need of the public key (in some applications this may be useful).

As a result, the parameters of FLASH have been chosen in order to satisfy an extreme property that very few public key scheme have reached so far: efficiency

* Part of this work is an output of project "Turbo-signatures", supported by the french Ministry of Research.

on low-price smart cards. FLASH has been specially designed for this specific application because we thought that for all the classical applications of signature schemes, the classical algorithms (RSA, Fiat-Shamir, Elliptic Curves, DSA, etc) are very nice, but when we need some very specific properties these algorithms just cannot satisfy them, and it creates a real practical need for algorithms such as FLASH.

FLASH was designed to have a security level of 2^{80} with the present state of the art in Cryptanalysis.

2 FLASH: the basic ideas

(This paragraph is here to help the understanding of FLASH. FLASH will then be described in details in the next paragraphs.)

Let $K = \mathbf{F}_q = \mathrm{GF}(q)$ be a small finite field (in FLASH we will choose $K = \mathbf{F}_{256}$ and in SFLASH we will choose $K = \mathbf{F}_{128}$).

Let n and α be two integers (in FLASH and SFLASH we will have $n = 29$ and $\alpha = 11$).

Let F be the following function:

$$F : \begin{cases} \mathbf{F}_{q^n} \to \mathbf{F}_{q^n} \\ x \mapsto x^{1+q^\alpha} \end{cases}$$

This function F can be seen in two different ways:

1. It can be seen as a monomial function with only one variable $x \in \mathbf{F}_{q^n}$, of degree $1 + q^\alpha$.
2. Or, if we write this function F as a function from $\mathbf{F}_{q^n}$ to $\mathbf{F}_{q^n}$, it can be seen as a multivariate function from n variables $(x_1, \ldots, x_n) \in K^n$ to n variables $(y_1, \ldots, y_n) \in K^n$, of total degree 2.

From the univariate representation (1), it is easy to invert F when $1 + q^\alpha$ is coprime to $q^n - 1$ (we will always choose q, n and α such that this condition is satisfied). In this case, it can be proven that the inverse function F^{-1} of F is also a monomial function:

$$F^{-1}(x) = x^h$$

where h is an integer such that

$$h \cdot (1 + q^\alpha) = 1 \bmod (q^n - 1).$$

From the multivariate representation (2), we will be able to "hide" this function F by introducing two secret bijective affine transformations s and t from K^n to K^n, and we will compute $G' = t \circ F \circ s$ and keep F secret. This function G' is a quadratic function from K^n to K^n.

Now, we use another important idea: we will not publish <u>all</u> the n quadratic equations of G', but only $n - r$ of these equations (*ie* r equations will be kept secret). (In FLASH and SFLASH, $r = 11$.)

Let G be the public function from K^n to K^{n-r} obtained like this. Then G will be the public key and t, s and the r equations removed are the secret key. As we will see below from G, we will be able to design the very efficient signature schemes FLASH and SFLASH.

Remark: FLASH and SFLASH are very similar to the scheme C^* published in 1988 by T. Matsumoto and H. Imai (cf [?]). However, there are two major changes:

1. In FLASH and SFLASH, there is only "one branch".
2. In FLASH and SFLASH, r equations of the composition $G' = t \circ F \circ s$ are kept secret, where $q^r \geq 2^{80}$.

Without these changes, the schemes can be broken (see [3] and [4]).

3 Notations and Parameters of the Algorithm

In all the present document, $||$ will denote the "concatenation" operation. More precisely, if $\lambda = (\lambda_0, \ldots, \lambda_m)$ and $\mu = (\mu_0, \ldots, \mu_n)$ are two strings of elements (in a given field), then $\lambda||\mu$ denotes the string of elements (in the given field) defined by:

$$\lambda||\mu = (\lambda_0, \ldots, \lambda_m, \mu_0, \ldots, \mu_n).$$

For a given string $\lambda = (\lambda_0, \ldots, \lambda_m)$ of bits and two integers r, s, such that $0 \leq r \leq s \leq m$, we denote by $[\lambda]_{r \to s}$ the string of bits defined by:

$$[\lambda]_{r \to s} = (\lambda_r, \lambda_{r+1}, \ldots, \lambda_{s-1}, \lambda_s).$$

The FLASH algorithm uses two finite fields.

- The first one, $K = \mathbf{F}_{256}$ is precisely defined as $K = \mathbf{F}_2[X]/(X^8 + X^6 + X^5 + X + 1)$. We will denote by π the bijection between $\{0,1\}^8$ and K defined by:

$$\forall b = (b_0, \ldots, b_7) \in \{0,1\}^8,$$

$$\pi(b) = b_7 X^7 + \ldots + b_1 X + b_0 \pmod{X^8 + X^6 + X^5 + X + 1}.$$

- The second one is $\mathcal{L} = K[X]/(X^{37} + X^{12} + X^{10} + X^2 + 1)$. We will denote by φ the bijection between K^{37} and $\mathcal{L}$ defined by:

$$\forall \omega = (\omega_0, \ldots, \omega_{36}) \in K^{37}$$

$$\varphi(\omega) = \omega_{36} X^{36} + \ldots + \omega_1 X + \omega_0 \pmod{X^{37} + X^{12} + X^{10} + X^2 + 1}.$$

3.1 Secret Parameters

1. An affine secret bijection s from K^{37} to K^{37}. Equivalently, this parameter can be described by the 37×37 square matrix and the 37×1 column matrix over K of the transformation s with respect to the canonical basis of K^{37}.
2. An affine secret bijection t from K^{37} to K^{37}. Equivalently, this parameter can be described by the 37×37 square matrix and the 37×1 column matrix over K of the transformation s with respect to the canonical basis of K^{37}.
3. A 80-bit secret string denoted by Δ.

3.2 Public Parameters

The public key consists in the function G from K^{37} to K^{26} defined by:

$$G(X) = (Y_0, Y_1, \ldots, Y_{25}),$$

where

$$Y = (Y_0, Y_1, \ldots, Y_{37}) = t\Big(\varphi^{-1}\big(F(\varphi(s(X)))\big)\Big).$$

Here F is the function from $\mathcal{L}$ to $\mathcal{L}$ defined by:

$$\forall A \in \mathcal{L},\ F(A) = A^{256^{11}+1}.$$

By construction of the algorithm, G is a quadratic transformation over K, *i.e.* $(Y_0, \ldots, Y_{25}) = G(X_0, \ldots, X_{36})$ can be written, equivalently:

$$\begin{cases} Y_0 = P_0(X_0, \ldots, X_{36}) \\ \quad \vdots \\ Y_{25} = P_{25}(X_0, \ldots, X_{36}) \end{cases}$$

with each P_i being a quadratic polynomial of the form

$$P_i(X_0, \ldots, X_{36}) = \sum_{0 \le j < k < 37} \zeta_{i,j,k} X_j X_k + \sum_{0 \le j < 37} \nu_{i,j} X_j + \rho_i,$$

all the elements $\zeta_{i,j,k}$, $\nu_{i,j}$ and ρ being in K.

4 Signing a message

In the present section, we describe the signature of a message M by the FLASH algorithm.

4.1 The signing algorithm

The message M is given by a string of bits. Its signature S is obtained by applying successively the following operations (see figure 1):

1. Let M_1 and M_2 be the three 160-bit strings defined by:

$$M_1 = \text{SHA-1}(M),$$

$$M_2 = \text{SHA-1}(M_1).$$

2. Let V be the 208-bit string defined by:

$$V = [M_1]_{0 \to 159} || [M_2]_{0 \to 47}.$$

3. Let W be the 88-bit string defined by:

$$W = [\text{SHA-1}(V || \Delta)]_{0 \to 87}.$$

4. Let Y be the string of 26 elements of K defined by:

$$Y = \Big(\pi([V]_{0\to 7}), \pi([V]_{8\to 15}), \ldots, \pi([V]_{200\to 207})\Big).$$

5. Let R be the string of 11 elements of K defined by:

$$R = \Big(\pi([W]_{0\to 7}), \pi([W]_{8\to 15}), \ldots, \pi([V]_{80\to 87})\Big).$$

6. Let B be the element of $\mathcal{L}$ defined by:

$$B = \varphi\Big(t^{-1}(Y||R)\Big).$$

7. Let A be the element of $\mathcal{L}$ defined by:

$$A = F^{-1}(B),$$

F being the function from $\mathcal{L}$ to $\mathcal{L}$ defined by:

$$\forall A \in \mathcal{L},\ F(A) = A^{256^{11}+1}.$$

8. Let $X = (X_0, \ldots, X_{36})$ be the string of 37 elements of K defined by:

$$X = (X_0, \ldots, X_{36}) = s^{-1}\Big(\varphi^{-1}(A)\Big).$$

9. The signature S is the 296-bit string given by:

$$S = \pi^{-1}(X_0)|| \ldots ||\pi^{-1}(X_{36}).$$

4.2 Computing $A = F^{-1}(B)$

The function F, from $\mathcal{L}$ to $\mathcal{L}$, is defined by:

$$\forall A \in \mathcal{L},\ F(A) = A^{256^{11}+1}.$$

As a consequence, $A = F^{-1}(B)$ can be obtained by the following formula:

$$A = B^h,$$

the value of the exponent h being the inverse of $256^{11} + 1$ modulo $256^{37} - 1$. In fact, h can be explicitly given by:

$$h = 2^{295} + \sum_{i=0}^{17} \sum_{j=176i+87}^{176i+174} 2^j.$$

Three methods can be used to compute $A = B^h$:

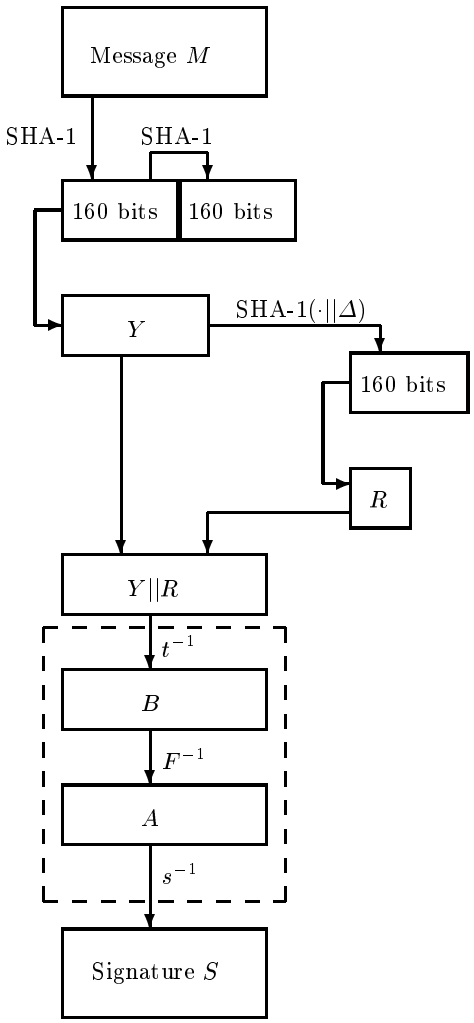

Fig. 1. Signature generation with FLASH

1. Directly compute the exponentiation B^h by using the "square-and-multiply" principle.
2. Use the following algorithm:
   (a) Initialize A to:
   $$A = B^{2^{87}} \quad \left(= B^{256^{10}} \cdot B^{128} \right).$$
   Note that $B \mapsto B^{256^{10}}$ is a linear transformation of $\mathcal{L}$ if we consider $\mathcal{L}$ as a vector space over K and can thus be easily computed.
   (b) Compute
   $$u = A^{256^{11}-1}.$$
   This value can be computed either by using the "square-and-multiply" principle or by noticing that we also have
   $$u \cdot A = A^{256^{11}}$$
   with $A \mapsto A^{256^{11}}$ being a linear transformation of $\mathcal{L}$ if we consider $\mathcal{L}$ as a vector space over K. We can thus easily find A by solving a system of linear equations over K.
   (c) Apply 18 times the following transformation: replace A by $u \cdot A^{256^{22}}$. This is also practical, since $A \mapsto A^{256^{22}}$ is a linear transformation of $\mathcal{L}$ (considered as a vector space over K).
3. Finally, we can also use the fact that
   $$A \cdot B^{256^{11}} = A^{256^{22}} \cdot B.$$
   Since $B \mapsto B^{256^{11}}$ and $A \mapsto A^{256^{22}}$ are two linear transformations of $\mathcal{L}$ (considered as a vector space over K), A can be found by solving a system of linear equations over K.

5 Verifying a signature

Given a message M (*i.e.* a string of bits) and a signature S (a 296-bit string), the following algorithm is used to decide whether S is a valid signature of M or not:

1. Let M_1 and M_2 be the three 160-bit strings defined by:
   $$M_1 = \text{SHA-1}(M),$$
   $$M_2 = \text{SHA-1}(M_1).$$
2. Let V be the 208-bit string defined by:
   $$V = [M_1]_{0 \to 159} || [M_2]_{0 \to 47}.$$

3. Let Y be the string of 26 elements of K defined by:

$$Y = \Big(\pi([V]_{0\to 7}), \pi([V]_{8\to 15}), \ldots, \pi([V]_{200\to 207})\Big).$$

4. Let Y' be the string of 26 elements of K defined by:

$$Y' = G\Big(\pi([S]_{0\to 7}), \pi([S]_{8\to 15}), \ldots, \pi([S]_{288\to 295})\Big).$$

5. – If Y equals Y', accept the signature.
   – Else reject the signature.

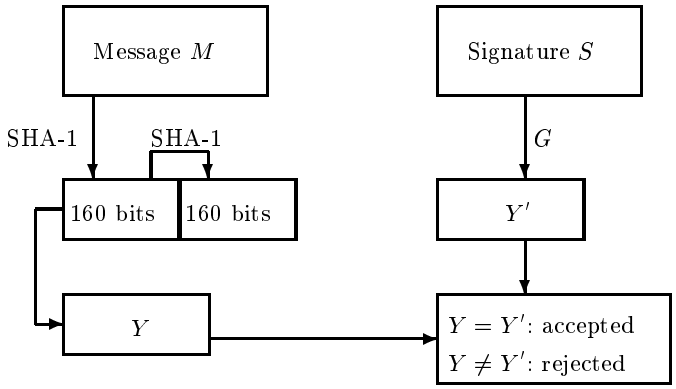

Fig. 2. Signature verification with FLASH

6 Security of the FLASH algorithm

FLASH is a C^{*--} scheme with a special choice of the parameters.

The security of such schemes has been studied in [4].

The security is not proven to be equivalent to a simple to describe and assumed difficult to solve problem. However, here are the present results on the two possible kinds of attacks :

6.1 Attacks that compute a valid signature from the public key as if it was a random set of quadratic equations (*i.e.* without using the fact that we have a C^{*--} scheme)

These attacks have to solve a MQ problem (MQ: Multivariate Quadratic equations), and the general MQ problem is NP-Hard. Moreover, when the parameters are well chosen, the known algorithms for solving such an MQ problem (such as XL, FXL or some Gröbner base algorithms) are efficient. With our choice of parameters for FLASH, they require more computations than the equivalent of 2^{80} operations.

6.2 Attacks that use the fact that the public key comes from a C^{*--} scheme (and is not a random set of quadratic equations)

All the known attacks on this family have a complexity in $\mathcal{O}(qr)$, where r is the number of removed equations ($r = 11$ in the FLASH algorithm), and where q is the number of elements of the finite field K used (so $q = 256 = 2^8$ for the FLASH algorithm). So these attacks will require more than the equivalent of 2^{80} operations for the FLASH algorithm.

7 Summary of the characteristics of FLASH

- Length of the signature: 296 bits.
- Length of the public key: 18 Kbytes.
- Length of the secret key: the secret key (2.75 Kbytes) is generated from a small seed of at least 128 bits.
- Time to sign a message[1]: less than 2.7 ms (maximum time).
- Time to verify a signature[2]: less than 0.8 ms (*i.e.* approximately $37 \times 37 \times 26$ multiplications and additions in K).
- Time to generate a pair of public key/secret key: less than 1 s.
- Best known attack: more than 2^{80} computations.

8 The SFLASH algorithm

In this chapter we introduce a modification of FLASH, that is made in order to have a smaller public key length. For this purpose, we will choose our parameters such that the coefficients of the public key lie in the prime subfield $\mathbf{F}_2$ of K. For this we need to satisfy two conditions.
First, the coefficients of the irreducible polynomial that defines $\mathcal{L}$ need to be in $\mathbf{F}_2$. Secondly, the affine invertible transformations s and t need to be defined over $\mathbf{F}_2$.

Moreover, in order to avoid a possible attack described in [1] (this attack uses the factorization of the extension degree 8 and the existence of an intermediate extension), we choose $K = \mathbf{F}_{128}$ in SFLASH.. Then the of [1] attack fails and we obtain an algorithm that is not broken with a difference from FLASH that the public key takes 2.2 Kbytes instead of 18. It is also sensibly faster in verification.

9 Summary of the characteristics of SFLASH

- Length of the signature: 259 bits.
- Length of the public key: 2.3 Kbytes.

[1] On a Pentium III 500 MHz. This part can be improved: the given software was not optimized.
[2] This part can be improved: the given software was not optimized.

- Length of the secret key: the secret key (0.35 Kbytes) is generated from a small seed of at least 128 bits.
- Time to sign a message[3]: less than 2.6 ms (maximum time).
- Time to verify a signature[4]: less than 0.4 ms (*i.e.* approximately $37 \times 37 \times 26 \times \frac{1}{2}$ multiplications and additions in K).
- Time to generate a pair of public key/secret key: less than 1 s.
- Best known attack: more than 2^{80} computations.

References

1. Nicolas Courtois, *Asymmetric cryptography with algebraic problems MQ, Min-Rank, IP and HFE.* PhD thesis, Paris 6 University, to appear soon.
2. Tsutomu Matsumoto and Hideki Imai, *Public Quadratic Polynomial-tuples for efficient signature-verification and message-encryption*, Proceedings of EUROCRYPT'88, Springer-Verlag, pp. 419-453.
3. Jacques Patarin, *Cryptanalysis of the Matsumoto and Imai public Key Scheme of Eurocrypt'88*, Proceedings of CRYPTO'95, Springer-Verlag, pp. 248-261.
4. Jacques Patarin, Louis Goubin, and Nicolas Courtois, *C^{*-+} and HM: Variations around two schemes of T. Matsumoto and H. Imai*, in Advances in Cryptology, Proceedings of ASIACRYPT'98, LNCS n° 1514, Springer Verlag, 1998, pp. 35-49.

[3] On a Pentium III 500 MHz. This part can be improved: the given software was not optimized.

[4] This part can be improved: the given software was not optimized.