PFLASH - Secure Asymmetric Signatures on Smart Cards

Ming-Shing Chen[1], Bo-Yin Yang[1], and Daniel Smith-Tone[2]

[1]Academia Sinica,
Taipei, Taiwan
[2]National Institute of Standards and Technology,
Gaithersburg, Maryland, USA

mschen@crypto.tw, daniel.smith@nist.gov, by@crypto.tw

Abstract. We present PFLASH, an asymmetric digital signature scheme appropriate for smart card use. We present parameters for several security levels in this low resource environment and bootstrap many technical properties (including side-channel resistance) exposed in the evaluation of predecessors of this scheme.
PFLASH is a multivariate signature scheme with a specific set of parameters. Specifically, PFLASH is a pC^{*-} scheme which means that geometrically the scheme can be viewed as a morphism of a monomial permutation, restricting the domain and range to two subspaces of an n-dimensional vector space over a finite field $\mathbb{F}_q$.
PFLASH is a direct descendent of the SFLASH signature scheme which was recommended by NESSIE in 2003 and subsequently broken in 2007. Since that time we have developed a greater understanding of security for these so called "big field schemes." PFLASH provably resists a large class of attacks on multivariate cryptosystems, a class which includes all known attacks on multivariate cryptosystems. While this doesn't constitute a guarantee of the security of PFLASH, it does imply that any attack on the system will require a fundamental mathematical advance which the scientific community has not discovered in the nearly two decades since the first suggestion of pC^{*-} schemes.
The performance of PFLASH is comparable to that of its parent SFLASH, being roughly $\frac{q}{2}$ times slower. This level of efficiency still makes PFLASH faster than RSA and far easier to implement on a smart card without an arithmetic coprocessor. The public key size is far larger than RSA, but the scheme far outperforms RSA, does not suffer nearly as much to poor random number generation and still fits easily on the cheapest smart cards.
Optimization of this scheme and simulations in the smart card environment is a continuing project the results of which will be included in the full version of this manuscript.

1 Introduction

PFLASH is designed in the lineage of the C^* scheme of Imai and Matsumoto from Eurocrypt '88, see [1]. The immediate predecessor of PFLASH is SFLASH

which was recommended by the NESSIE consortium in 2003 as the fastest option for digital signatures on low cost smart cards, see [2]. Unfortunately, in 2007 SFLASH was famously broken in [3].

The field of multivariate public key cryptography has matured a great deal since that time. We now have a greater understanding of the techniques available for key recovery attacks, the complexity of direct algebraic inversion of public keys, and frameworks for proving security against a large class of attacks, see [4,5,6,7] for example. Moreover, we have known since 2007 that one of the original suggestions for a C^* family signature scheme preserves the fundamental efficient structure of SFLASH while avoiding the weaknesses. At this point we have the theoretical tools to justify implementing PFLASH, a scheme which balances great speed with security and small operating costs, ideal for the smart card environment.

One should note that many of the developments in the evolution of SFLASH carry over directly to PFLASH, such as resistance to differential power analysis, see [8]. In addition, PFLASH is ideally suited for small footprints, requiring only a minimal amount of volatile memory. PFLASH also inherits the speed of SFLASH, being as little as four times slower than its swift progenitor for constant-time implementations.

This manuscript is organized as follows. In Section 2 the PFLASH scheme is presented along with inversion and verification procedures. In the following section the security of PFLASH against a differential adversary, as well as a discussion of other measurable security criteria is offered. Section 4 provides some timings in terms of cycle counts on various platforms with different instruction sets.

2 PFLASH

PFLASH is descended from the original "big field" cryptosystem C^*, see [1]. The construction of C^* is as follows. Fix a finite field $\mathbb{F}_q$ with q elements. Fix a degree n extension k. We may identify (via a tacitly understood vector space isomorphism) k with $\mathbb{F}_q^n$. Thus n-dimensional vectors over $\mathbb{F}_q$ are simultaneously n-tuples of elements of $\mathbb{F}_q$ and individual elements of k.

Consider the C^* monomial map

$$f(x) = x^{q^\theta+1},$$

where $gcd(q^n - 1, q^\theta + 1) = 1$. Given the condition on θ, f is a permutation of k. Since raising to the power of q is simply a Frobenius automorphism of k over $\mathbb{F}_q$, we recognize that $x \mapsto x^{q^\theta}$ is $\mathbb{F}_q$-linear. Thus multiplying by a single extra copy of x results in a system of formulae quadratic in the unknown coefficients of x.

To hide this easily invertible structure two affine transformations T, U are composed with f as follows, $P = T \circ f \circ U$. This basic construction represents a few different multivariate cryptosystems depending on the properties of T and U.

If both T and U are invertible, the scheme is known as C^*. This scheme was broken in 1995 by Patarin [9]. If T is singular, the scheme becomes a C^{*-}, where the $(-)$ modifier refers to the fact that the public key's range is a proper $\mathbb{F}_q$-subspace of k. SFLASH is an example of such a scheme. If U is singular, the scheme becomes a pC^* scheme, where the (p) modifier refers to the fact that the plaintext space is projected onto a subspace. The SQUARE scheme is an example. PFLASH combines both of these modifiers and is thus a pC^{*-} scheme.

Any pC^{*-} scheme can be described entirely by a few parameters. First, we require q and n to build $\mathbb{F}_q$ and k. Next we require r, the corank of T, or equivalently, the number of equations removed from the public key. We require d, the corank of U.

Considering the state of the art and what we can prove, our initial parameter sets have $q = 16$, $n = 62$, $r = 22$, and $d = 1$ for 80-bit security, $q = 16$, $n = 74$, $r = 22$, and $d = 1$ for 104-bit security, and $q = 16$, $n = 94$, $r = 30$ and $d = 1$ for 128-bit security. These values are suggested to maintain the size of q^r, q^{n-r}, and $\binom{n-r+d+1}{n-r}$, as discussed in Section 3. In addition, the corresponding author suggests the parameter set $q = 16$, $n = 51$, $r = 19$, and $d = 1$ for 64-bit short-term security on a card with an 8-bit multiplier. We believe each of these parameter sets to be immune from key recovery attacks at least to the 120-bit level, with security against forgery attacks listed above. These data are summarized in Table 1.

Scheme	Pub. Key	Digest	Security	Key Recovery
PFLASH(GF16,62,22,1)	39,040B	160b	80b	120+b
PFLASH(GF16,74,22,1)	72,124B	208b	104b	120+b
PFLASH(GF16,94,30,1)	142,848B	256b	128b	120+b
PFLASH(GF16,51,19,1)	21,200B	128b	64b	120+b

Table 1. Suggested parameter sets for PFLASH at various security levels. The entry below the double-bar is a short-term secure scheme.

To generate a signature, one selects a preimage under T, inverts f, and selects a preimage under U. This task entails computing several sums, products, and squares in k. For a fixed private key, the most efficient method of inversion involves determining a way to minimize the number of field multiplications to perform— see [8], for example— as the standard square and multiply is much slower. We chose in our analysis to be more conservative and to be more concerned with side-channel resistant and generic constant time implementations.

Verification is accomplished by simply plugging the signature coefficients into the public key to recover the image.

3 Security

Many new tools have been developed since the attack on SFLASH for measuring the security of multivariate cryptosystems. In particular, [4,5,6,7] offers a model for proving the resistance of a scheme to differential attacks, the family of attacks exploiting properties of the discrete differential of the public key. The discrete differential is given by $Df(a, x) = f(a + x) - f(a) - f(x) + f(0)$.

In [5] it is shown that given proper parameter choices PSFLASH provably has no differential symmetry. Thus PFLASH is immune from the attack which broke SFLASH. This proof of security is information theoretic; therefore, no advance in techniques can render PFLASH vulnerable to a differential symmetric attack, even as yet undiscovered attack methods.

Similarly, in [6] PFLASH is shown to have no nontrivial differential invariant structure, and thus cannot be attacked by any method exploiting an unexpected linear action of the public key on some large subspace of the plaintext space. An example of such an attack, presented with different terminology, was discovered on the oil and vinegar signature scheme, see [10]. Thus PFLASH has been proven secure against a differential adversary.

PFLASH is also provably resistant to rank attacks. Each public formula can be written as a quadratic form over $\mathbb{F}_q$. The corank of the matrix representation of the quadratic forms is expected to be low with high probability. Moreover, it is expected for the maximum corank of any nontrivial matrix in the span of the public quadratic forms to be small with high probability. Thus PFLASH has no anomalous rank structure to be exploited, and the scheme is secure against rank attacks.

An interesting attack is presented in [11] which is able to remove the $(-)$ modifier and to remove the (p) modifier individually on HFE schemes, which replace f with a slightly more general polynomial. These attacks rely on Q-rank, which is the rank of the entire public key considered as a quadratic form from an n-dimensional representation of k over itself. These attacks would certainly be applicable to PFLASH since PFLASH has Q-rank 1, however, the (p) modifier is only able to be removed when the $(-)$ modifier has been removed, and the $(-)$ modifier can only be removed when T is of corank 1. Thus PFLASH is out of range for this attack.

One might also consider the security of PFLASH against attacks trying to recover f from P by finding T' and U' such that $P = T' \circ f \circ U'$. Since there are so few possibilities for f, one may consider f to be given to an adversary. However, this problem is a known complexity theoretic problem called the $\mathcal{MP}$ or morphism of polynomials problem. This problem is NP-complete, and there is no evidence that the systems arising from PFLASH form easy instances.

Direct algebraic attacks involve computing a Grobner basis for the system of equations arising from setting $P(x) = y$. While this technique does not in practice provide a key recovery attack, it does in general pose a threat for message recovery for multivariate schemes. To analyse this we note that algebraically PFLASH is an HFE^- scheme. Thus we can use the derived estimate of the degree of regularity of HFE^- schemes from [12] to determine the complexity

of direct inversion. Given the suggested parameters, we compute the expected degree of regularity over $GF(2)$ of PFLASH to be $\frac{r+4}{2}$ where r is the number of equations removed when the scheme is considered over $GF(2)$. For the suggested parameters we compute the complexity of algebraic inversion to be greater than brute force search.

In addition to the theoretical security, PFLASH can also bootstrap methods derived in the development process for SFLASH to resist side-channel attacks as well. Timing attacks can be easily avoided for these schemes by making signature generation a nearly constant time process. In addition, there are known masking techniques for preventing SPA and DPA attacks on the hardware, see [8], for example.

In summary, for the suggested parameters we achieve 120+ bits of security against key recovery attacks. The limiting factor in the security of these schemes is the digest size. For these schemes over a 4-bit field, the security level against a forgery attack is half of the digest, hence the claimed security levels in Table 1.

4 Performance

Due to time constraints, we have only implemented and optimized those versions of PFLASH which we deem to be of the most interest to the intended audience. We are still in the process of implementing and optimizing PFLASH in other environments of interest. We intend to post to eprint a full survey of PFLASH schemes relevant to manifold utilizări. We here present figures for the performance of side-channel resistant constant-time versions of PFLASH at the 80, 104, and 128-bit security levels.

The data are summarized in Tables 2 and 3. Table 2 displays our new side-channel resistant constant-time optimized SSE implementation of PFLASH on an Intel Xeon E3-1245 v3 3.40 GHz along with the eBATS figures of some comparable schemes on a different machine with the same architecture. Table 3 shows data for an implementation without vector instructions on several platforms.

We must note here several facts which should be taken into consideration in analyzing these data. First, the SSE data uses vector instructions which are likely unavailable in a low power environment. Still, these data are useful for comparing with other relevant schemes since benchmarking data are available for (and usually the most care in optimization given for) such platforms. One can check eBATS, see [13], for comparison with other schemes in various environments.

Second, these data are for side-channel resistant schemes; therefore, we may tweak parameters for better performance if, for example, the application is in an environment in which timing attacks are of no concern. Generic estimates of the cost of side-channel resistance include roughly a factor of two for SPA and DPA security via masking, see [8] for a reasonable reference, and also roughly a factor of two for constant time code.

Thirdly, PFLASH stands out along with a few of its surviving cousins in the C^* family tree having the properties that it can be implemented on a low power

Scheme	Security	Pub. Key	Sec. Key	Signature	Digest	Signing	Verifying
PFLASH(GF16,62,22,1)	80b	39,040B	3,937B	244b	160b	288,093c	17,007c
PFLASH(GF16,74,22,1)	104b	72,124B	5,587B	292b	208b	509,355c	23,829c
PFLASH(GF16,94,30,1)	128b	142,848B	8,977B	372b	256b	634,051c	38,044c
Rainbow(gf16,24,20,20)		94,384B	102,912B	256b	160b	24,616c	14,708c
Rainbow(gf31,24,20,20)		57,600B	150,512B	296b		42,700c	46,520c
Rainbow(gf256,18,12,12)		30,240B	23,408B	336b	192b	14,016c	10,560c
ed25519		32B	64B	512b		61,976c	184,992c
ec p256		64B	96B	512b		381,696c	913,848c
RSA 1024		128B	1024B	344b		1,186,912c	33,676c
RSA 2048		256B	2048B	344b		5,134,876c	67,916c

Table 2. Constant time implementation data for PFLASH with SSE instructions on Intel Xeon E3-1245 v3 3.40 GHz, avg. for 1000 trials. Also listed are comparable data from eBATS http://bench.cr.yp.to/results-sign.html on an Intel Xeon E3-1275 v3 3.50 GHz (same architecture).

Xeon (Haswell)		
PFLASH(GF16,74,22,1)	1,253,068	201,598
RSA1024	1,186,912	33,676
Ed25519	61,976	184,992
ECDSAp256	381,696	913,848
ARM Cortex-A8		
PFLASH(GF16,74,22,1)	4,628,701	740,429
RSA1024	7,878,747	3,860,809
Ed25519	819,157	2,594,303
ECDSAp256	5,378,137	6,317,331
MIPS o32		
PFLASH(GF16,74,22,1)	5,710,020	1,105,242
RSA1024	17,756,132	385,956
Ed25519	2,612,848	8,762,140
ECDSAp256	14,586,352	17,535,264

Table 3. Implementations without vector instructions. Cycles are listed for the instruction sets of Xeon (Haswell), ARM Cortex-A8, and MIPS o32. The cycle-counts of these number theoretic schemes change more dramatically than PFLASH based on the width of the available multiplication instructions.

or even no-power device and having such a small private key. Often, a smart card need only have a copy of the private key in its storage, and this makes PFLASH an obvious candidate.

Finally, it is worth noting that the orders of magnitude for these data are changed very little even when implemented using a microprocessor with an 8-bit multiplier (though with a lower clock rate). The same fact cannot be said for many other schemes with larger fields or with integer arithmetic. For such schemes, being constrained by the very power-restricted environment of an 8-bit multiplier is catastrophic.

5 Summary

1. Signature Sizes: 244, 292, 372 bits
2. Length of public key: a few dozen Kbytes
3. Length of private key: a few Kbytes
4. Best known attack: brute force
5. Side-Channel Resistant
6. Time constant
7. Scalable under various security criteria
8. Appropriate for very low power devices

References

1. Matsumoto, T., Imai, H.: Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption. In: EUROCRYPT. (1988) 419–453
2. Preneel, B.: Nessie project announces final selection of crypto algorithms (2003) https://www.cosic.esat.kuleuven.be/nessie/deliverables/press_release_feb27.pdf.
3. Dubois, V., Fouque, P.A., Shamir, A., Stern, J.: Practical cryptanalysis of SFLASH. Advances in Cryptology - CRYPTO 2007, Springer **4622** (2007) 1–12
4. Smith-Tone, D.: Properties of the discrete differential with cryptographic applications. In Sendrier, N., ed.: PQCrypto. Volume 6061 of Lecture Notes in Computer Science., Springer (2010) 1–12
5. Smith-Tone, D.: On the differential security of multivariate public key cryptosystems. In Yang, B.Y., ed.: PQCrypto. Volume 7071 of Lecture Notes in Computer Science., Springer (2011) 130–142
6. Perlner, R.A., Smith-Tone, D.: A classification of differential invariants for multivariate post-quantum cryptosystems. In Gaborit, P., ed.: PQCrypto. Volume 7932 of Lecture Notes in Computer Science., Springer (2013) 165–173
7. Daniels, T., Smith-Tone, D.: Differential properties of the HFE cryptosystem. In Mosca, M., ed.: Post-Quantum Cryptography - 6th International Workshop, PQCrypto 2014, Waterloo, ON, Canada, October 1-3, 2014. Proceedings. Lecture Notes in Computer Science, Springer (2014) 59–75
8. Akkar, M., Courtois, N., Duteuil, R., Goubin, L.: A fast and secure implementation of sflash. In Desmedt, Y., ed.: Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings. Volume 2567 of Lecture Notes in Computer Science., Springer (2003) 267–278

9. Patarin, J.: Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt '88. Crypto 1995, Springer **963** (1995) 248–261
10. Shamir, A., Kipnis, A.: Cryptanalysis of the oil & vinegar signature scheme. CRYPTO 1998. LNCS **1462** (1998) 257–266
11. Bettale, L., Faugère, J.C., Perret, L.: Cryptanalysis of hfe, multi-hfe and variants for odd and even characteristic. Des. Codes Cryptography **69** (2013) 1–52
12. Ding, J., Hodges, T.J.: Inverting hfe systems is quasi-polynomial for all fields. In Rogaway, P., ed.: CRYPTO. Volume 6841 of Lecture Notes in Computer Science., Springer (2011) 724–742
13. : eBATS measurements of public key signature schemes, indexed by machine. (`http://bench.cr.yp.to/results-sign.html`) Accessed: 01-Jul-2015.