The TriTon Transformation

Daniel Smith-Tone[1]

[1]Department of Mathematics, University of Louisville,
Louisville, Kentucky, USA
[1]National Institute of Standards and Technology,
Gaithersburg, Maryland, USA

daniel.smith@nist.gov

Abstract. Many new systems have been proposed which hide an easily invertible multivariate quadratic map in a larger structure by adding more variables and introducing some mixing of a random component to the structured system. While many systems which have been formed by attempting to hide the hidden structure of equations have been broken by observing symmetric properties of the differential of the public key, the dichotomy between the roles of the different types of variables, or even the different types of monomials in the systems, have given rise to differential invariant attacks which distinguish between subspaces corresponding to one type of variable or the other. In this monologue, we take a general approach, and describe a basic construction, TriTon, of which several of the above types of systems are special cases. We analyse this system, and conclude that such constructions are weak with naive choices of parameters.

Key words: multivariate public key cryptography, differential invariant

1 Introduction

Since 1994, when Peter Shor discovered the key to factoring large composite integers and computing discrete logarithms in polynomial time on a quantum computer, see [1], there has been an ongoing challenge to develop a secure and practical public key replacement for RSA and Diffie-Hellman. This quest to find quantum-resistant mechanisms to replace the current public key infrastructure is wraught with difficulties. In addition to the challenges of designing asymmetric schemes which are immune to classical attack, the task of the post-quantum cryptographer is to create cryptographic tools which are invulerable in a computational model, the understanding of which is constantly evolving.

As a result of such difficulties, the main approach is to design public key cryptosystems in the classical model of computing which do not admit efficient analysis by known quantum techniques. This process often results in cryptosystems which suffer from massive public keys. In light of Grover's search algorithm, see [2], it is entirely possible that we may have no other option in this matter. What we can do, however, is construct schemes which are extremely fast.

Speed is one of the motivating factors for the development of a secure Multivariate Public Key Cryptosystem (MPKC). In addition to its other virtues— such as extreme parametrizability, the NP-completeness of the fundamental problem of inverting a system of multivariate equations, and the fact that empirically this problem seems difficult in the average case— multivariate systems, and in particular the "big field" schemes, are extremely efficient, often having speeds dozens of times faster than RSA, [3–5].

The big question about these MPKCs is whether we can be assured of the security of a system while retaining such desirable properties. Many schemes, such as C^*, SFLASH, PMI, ℓIC-, Oil and Vinegar, and the various Square variants, have been broken by uncovering some of the structure inherent to the public key. See [6–11]. Although there are some general theoretical results about the security of such cryptosystems, see [12, 13], the resistance of these systems against structural attack is not well understood.

In this paper, we analyze an approach to the construction of schemes which involve variables of multiple types. We call such schemes "TriTon," because they contain three colors, or flavors, of monomials— the structure monomials, the obfuscation monomials, and the mixing monomials. We endeavor to reveal some fundamental structural weaknesses of such schemes to further the development of security theory; in particular, we break some instances with naive parameters.

The paper is organized as follows. In the next section we present the TriTon transformation of a multivariate cryptosystem and describe why such a modification might seem beneficial. In the subsequent section, we express several well-known schemes as TriTon transformations of more basic systems. The following sections describes an attack against certain TriTon schemes with poorly chosen parameters. Finally, we draw conclusions about the trustworthiness of systems derived from such a design philosophy.

2 TriTon Construction

Let q be a prime power, and let $\mathbb{F}_q$ be a finite field with q elements. Given an effectively invertible quadratic function, $f : \mathbb{F}_q^n \to \mathbb{F}_q^m$, a quadratic function, $g : \mathbb{F}_q^l \to \mathbb{F}_q^m$, and $A : \mathbb{F}_q^{n+l} \to \mathbb{F}_q^m$ bilinear, the TriTon construction produces the function $\tilde{f} : \mathbb{F}_q^{n+l} \to \mathbb{F}_q^m$ as follows:

$$\tilde{f}(x, y) = f(x) + g(y) + A(x, y),$$

where $x \in \mathbb{F}_q^n$ and $y \in \mathbb{F}_q^l$.

To complete the scheme, we compose two affine transformations, $T : \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $U : \mathbb{F}_q^{n+l} \to \mathbb{F}_q^{n+l}$, to produce:

$$P(x) = T \circ \tilde{f} \circ U(x),$$

where $x \in \mathbb{F}_q^{n+l}$.

This construction has a great deal of algebraic structure, as can be seen by determining its differential. The discrete differential of an univariate function, f, is the bivariate function $Df(a, x) = f(a+x)-f(x)-f(a)+f(0)$. Since we are only interested in encryption functions which are quadratic, the differential will always be bilinear, and therefore each coordinate of the differential is a bilinear form. The differential of each coordinate of the core map, $\tilde{f}$, has the following structure:

$$D\tilde{f}_i = \begin{bmatrix} Df_i & A_i \\ A_i^T & Dg_i \end{bmatrix}.$$

The motivating force behind this transformation strategy is to hide any structure present in f without producing any new invariants or rank weaknesses. In addition, the ability to make A, or g, or both maps random may provide effective means of hiding the structure of f, and potentially enhance the security of the scheme.

3 Well-known TriTon Systems

While any system of multivariate equations can be defined using two sets of variables and separating the monomials into three categories, it is only reasonable to consider the system a TriTon construction if the system relies on this delegation of monomials into the three categories, structure, obfuscation, and mixing, for the effective inversion or analysis of the system. Several schemes have been proposed over the years fitting this description. Here we express a few well-known schemes which fit the above description, and give an example of a scheme which cannot effectively be considered in such a context.

3.1 Oil and Vinegar

The prototypical scheme differentiating between two types of variables is Oil and Vinegar, see [14]. In this scheme, the central map is defined in such a way that quadratic monomials in one type of variable, the oil variables, never occur. Thus the structured component is zero, the obfuscation component is comprised of monomials with random coefficients which are quadratic in the vinegar variables, and the mixing component is similarly random. Once the values of the vinegar variables have been fixed, the system is linear in the oil variables and they can be uniquely determined.

The differential of each single core map formula has the following form:

$$Df_i = \begin{bmatrix} 0 & Df_{i1} \\ Df_{i1}^T & Df_{i2} \end{bmatrix}.$$

Clearly, any vector of the form:

$$\begin{bmatrix} * \\ 0 \end{bmatrix},$$

that is, in the oil subspace, is mapped by Df_i to a vector of the form:

$$\begin{matrix} 0 \\ * \end{matrix},$$

in the vinegar subspace. Therefore, the product of a matrix in the span of the differential coordinate forms with the inverse of another such matrix will leave the oil subspace invariant, a fact which was exploited to break the balanced oil and vinegar scheme, see [10].

One may note that the unbalanced oil and vinegar scheme similarly admits a TriTon structure, as do several other vinegar variants of multivariate schemes. The main distinction between such systems and the balanced oil and vinegar scheme, is that they never have a trivial quadratic component of such a high, detectable dimension.

3.2 PMI

The C^* cryptosystem, developed by Matsumoto and Imai in [15], is the prototypical multivariate public key cryptosystem based on the structure of a large extension field. Given a degree $n+l$ extension, k, of our scalar field, the scheme expressed the composition of a hidden monomial map, $f : \mathbb{F}_q^{n+l} \to \mathbb{F}_q^{n+l}$, of the form $f(x) = x^{q^\theta+1}$, where $gcd(n+l, \theta) = 1$, and two affine transformations, $U, T : \mathbb{F}_q^{n+l} \to \mathbb{F}_q^{n+l}$, as a system of multivariate equations over the base field. The scheme, however, was later broken by Patarin, see [16], by virtue of a bilinear relation in the input and output of f.

The internally perturbed C^* scheme, PMI, see [17], uses the idea of adding a random summand of low dimensional support to the core map. Given the standard parameters of C^*, internal perturbation augments the core map, f, with a summand $g \circ L$, where $g : \mathbb{F}_q^l \to \mathbb{F}_q^{n+l}$ is a random quadratic map and $L : \mathbb{F}_q^{n+l} \to \mathbb{F}_q^l$ is a random linear map. Thus the entire encryption map is given by:

$$P(x) = T \circ f \circ U(x) + T \circ g \circ L \circ U(x).$$

The strategy here is to randomize the obfuscation monomials while retaining structure in the majority of the function. Once the randomized component is removed, the structure of the entire remaining map is utilized to find a preimage.

Specifically, the map $y = P(x)$ can be "inverted" by computing all possible outputs, z, of the random quadratic, g, subtracting Tz from $P(x)$, and applying the decryption routine of C^* to the result. If the output, x, of this procedure matches a preimage of z under $g \circ L \circ U$, then $P(x) = y$ and x is legitimately an inverse of y. If none of the q^l values of z share such a preimage with the C^* portion of the map, then y is not in the image of P.

With a change of basis we can express L as:

$$\tilde{L} = \begin{matrix} 0 & 0 \\ 0 & I \end{matrix}.$$

We then have:

$$P(x) = \tilde{T} \circ \tilde{f} \circ \tilde{U}(x) + \tilde{T} \circ \tilde{g} \circ \tilde{L} \circ \tilde{U}(x),$$

and in this basis the differential of each formula in the central map has the form:

$$D\tilde{f}_i + D(\tilde{g}_i\tilde{L})_i = \begin{matrix} D\tilde{f}_{i1} & D\tilde{f}_{i2} \\ D\tilde{f}_{i2}^T & D\tilde{g}_i + D\tilde{f}_{i3} \end{matrix}.$$

One may note that for $n+l$ odd, without the g component, each differential coordinate form has corank 1. If g is truely randomly selected, then often when LUx is nonzero, the rank of the differential coordinate form will be smaller. An equivalent observation involving the associated bilinear form of each public equation, along with some additional probabilistic methods resulted in an attack discovering the "noise kernel," effectively removing the obfuscation, see [18]. Notice that for $\begin{bmatrix} x \; y \end{bmatrix}^T \in \cap_i ker(D\tilde{g}_i)$ we have for all i:

$$\begin{matrix} D\tilde{f}_{i1} & D\tilde{f}_{i2} \\ D\tilde{f}_{i2}^T & D\tilde{g}_i + D\tilde{f}_{i3} \end{matrix} \begin{matrix} x \\ y \end{matrix} = \begin{matrix} D\tilde{f}_{i1} & D\tilde{f}_{i2} \\ D\tilde{f}_{i2}^T & D\tilde{f}_{i3} \end{matrix} \begin{matrix} x \\ y \end{matrix}.$$

3.3 pSFLASH - A Non-Example

pSFLASH is another scheme based on the original C^* scheme of Matsumoto and Imai, see [15]. After the discovery of Patarin's linearization attack, see [16], a new modification, the idea of discarding public equations, was suggested, [19]. This method was later shown to be weak in an attack exploiting a multiplicative symmetry exhibited by the differential of the public key by Dubois et al. from [9]. The results of this paper, and the subsequent generalization of the attack to other schemes, see [20], for example, further popularized differential methods in multivariate cryptanalysis and inspired several theoretical veins of inquiry, see [21, 12, 13].

The practical suggestion was proposed by Ding et al. in [22], that using the projection modifier, which is equivalent to making the affine transformation U singular, may prevent the attack using multiplicative symmetry. The resulting scheme is known as pSFLASH. The encryption map is formed as follows:

$$P(x) = T \circ f \circ S(x),$$

where f is a C^* monomial, and both S and T are singular with corank 1 and r, respectively.

The system is inverted by choosing a nonsingular map which agrees with T on the range of T, applying the inverse of this map, inverting f, and finding a preimage of S. Each of these operations is efficient for anyone with the knowledge of T, f, and S.

We may attempt to view this system as a TriTon scheme by choosing a change of basis which maps the image of S to the first $n-1$ basis vectors. The resulting scheme looks like:

$$P(x) = \tilde{T} \circ \tilde{f} \circ \tilde{S},$$

where $\tilde{S}$ is of the form:

$$\tilde{S} = \begin{matrix} \tilde{S}_1 & \tilde{S}_2 \\ 0 & 0 \end{matrix} .$$

As a result, the input of the hidden monomial map always has zero as the last coordinate, and we can equivalently regard the core map as including a projection onto the first $n-1$ coordinates, in which case the differential of the ith core coordinate formula has the form:

$$\begin{matrix} \widetilde{D(f)}_{i1} & 0 \\ 0 & 0 \end{matrix} .$$

In light of this fact, one may choose such a basis and consider the system as having one fewer variables. This has the effect of allowing a marginally smaller public key, and since an adversary can easily complete this computation there is no reason not to take advantage of this benefit. As a result, however, there is no advantage to considering this scheme as a TriTon construction.

4 Trivial Mixing Method and Analysis

In the previous section, we witness the strategies of adding a random component for obfuscation and of making the structured component trivial so that it does not interfere with the inversion of the mixing component. In this section, we describe another strategy called the Hidden Pair of Bijections scheme which has been proposed recently by Gotaishi, see [23], and present a cryptanalysis. The approach Gotaishi advocated requires the obfuscation component, g, to be invertible, and for the mixing component, A, to be of full rank. The resulting function defines a signature scheme analogous to the oil and vinegar scheme, in that one fixes the values of a set of variables, rendering the mixing component trivial, and inverts the resultant expression. The exposition of the scheme mentions that any form of structured quadratic components f and g could be used; for example, both f and g could be C^* monomials.

Specifically, to sign a message m, one begins by seting $z = H(m)$, a hash of the message. One then flips a coin determining which of x and y to fix to zero, and solves either $z = f(x)+g(0)+A(x, 0) = f(x)$, or $z = f(0) + g(y) + A(0, y) = g(y)$.

The claim is that the scheme is secure because for any particular signature an attacker is unaware whether the first n variables, x, are set to zero, or the second n variables, y; therefore, given a large number of signatures, it cannot be known which ones were signed with x set to zero and which were signed with y set to zero. This claim is false.

Consider the collection of all possible signatures, $\mathcal{S}$. $\mathcal{S}$ consist of two components: $\mathcal{S}_1$, the collection of all signatures which were derived from setting $x = 0$, and $\mathcal{S}_2$, the collection of all signatures which were derived from setting $y = 0$. Both $\mathcal{S}_i$ have dimension n, and therefore we are guaranteed that once an adversary intercepts $2n + 1$ signatures, the last signature will be in the span of n of the previous signatures, identifying the domain of either f or g. Projecting the entire scheme onto this subspace reduces the encryption map to the composition of two affine maps with f or g. Thus the scheme is no more difficult to invert than f or g, and it is broken.

In the rump session of PQCRYPTO '11, Gotaishi suggested a modification to repair the scheme [24]. His suggestion was to add a third type of variable and a third quadratic map, h, which is invertible, but which has no mixing with the other types of variables. The problem with this method is that the domain of this third quadratic map is a differential invariant, i.e. the differential of the core map has the form:

$$\begin{bmatrix} Df_i & A_i & 0 \\ A_i^T & Dg_i & 0 \\ 0 & 0 & Dh_i \end{bmatrix}.$$

Therefore, we can attack the scheme by finding the n-dimensional invariant subspace, and projecting onto its cosummand, reducing the scheme to Gotaishi's original primitive.

5 Generalization of the Trivial Mixing Method

The system of the previous section suffers from another fatal flaw. The requirement that the value to which x or y is fixed is zero is very restrictive, so that there are only $2q^n$ possible signatures, while the domain contains q^{2n} elements. Therefore the proportion of used bits is only $\frac{2}{q^n}$, indicating that the scheme is extremely inefficient.

This strategy of fixing the values of some of the inputs of the core map to render the mixing component trivial can still be used while fixing the inefficiency problem and avoiding the above attack by making the mixing component, A, of low rank. Since randomly choosing which affine half-dimensional space on which to project did not enhance the security of the hidden pair of bijections scheme, we can remove this feature and allow the obfuscation component g to take an arbitrary form. Thus the generalized core map takes the form (2) with A of corank k and the quadratic function g of whichever form optimizes security.

To sign a message, one randomly selects an element $z \in \cap_x ker(A(x, *))$, and, given a hash y, returns $U^{-1} \begin{matrix} f^{-1}(T^{-1}y - g(z)) \\ z \end{matrix}$. One can easily check that $T(f(f^{-1}(T^{-1}y - g(z))) + g(z) + A(f^{-1}(T^{-1}y - g(z)), z)) = y$

Each coordinate of the differential of this core map admits the presentation:

$$\begin{matrix} Df_i & A_i \\ A_i^T & Dg_i \end{matrix}.$$

Now, as before, an adversary can collect a maximal collection of linearly independent signatures, revealing $\cap_x ker(A(x, *))$. Since no signature is contained in the cokernel, we may project onto this kernel to obtain an equivalent map with a smaller domain. In this manner, the the induced map on the differential produces the following bilinear form:

$$\begin{matrix} D\tilde{f_i} & 0 \\ 0 & D\tilde{g_i} \end{matrix}.$$

Now each of these differential coordinate forms share an n-dimensional invariant subspace and a k-dimensional invariant subspace. Since the n-dimensional subspace, V, corresponds to the input of f, we compose yet another projection with the system and recover a system of equations linearly equivalent to $T \circ f \circ U|_V$. At this point, the inversion of the entire scheme is reduced to an inversion of the hidden map, f, and thus the construction is broken.

6 Conclusion

The basic idea of the Triton construction is to combine two disparate quadratic systems, mixing the variables together in such a way that the distillation of a single component is difficult. In many instances, however, the division of variables into classes and the delegation of particular monomials into certain required structures has caused a detectable change in the rank, or invariant structure of the differential of the encryption map.

In particular, the trivial mixing methodology seems fundamentally flawed, in that we can effectively develop a distinguisher which can separate the types of variables based on the properties of each class of monomial, regardless of the dimension associated with each type of variable. In comparison to the case of oil and vinegar, which resists the standard cryptanalysis when sufficiently unbalanced, trivial mixing seems particularly weak.

As a result of these facts, there is good reason to remain skeptical about techniques involving the division of variables into classes, or the introduction of intermediate variables, such as in the case of PMI. If rank methods and differential invariant methods continue to prove effective against such schemes, then none of these TriTon transformations of cryptosystems will be trusted.

References

1. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Sci. Stat. Comp. **26, 1484** (1997)
2. Grover, L.K.: A Fast quantum mechanical algorithm for database search. (1996) Proceedings STOC 1996, 212-219.
3. Chen, A.I.T., Chen, M.S., Chen, T.R., Cheng, C.M., Ding, J., Kuo, E.L.H., Lee, F.Y.S., Yang, B.Y.: Sse implementation of multivariate pkcs on modern x86 cpus. CHES 2009, LNCS, Springer, IACR **5747** (2009) 33–48
4. Chen, A.I.T., Chen, C.H.O., Chen, M.S., Cheng, C.M., Yang, B.Y.: Practical-sized instances of multivariate pkcs: Rainbow, tts, and ℓic-derivatives. Post-Quantum Crypto, LNCS **5299** (2008) 95–106
5. Yang, B.Y., Cheng, C.M., Chen, B.R., Chen, J.M.: Implementing minimized multivariate public-key cryptosystems on low-resource embedded systems. 3rd Security of Pervasive Computing Conference, LNCS **3934** (2006) 73–88
6. Clough, C., Baena, J., Ding, J., Yang, B.Y., Chen, M.S.: Square, a New Multivariate Encryption Scheme. In Fischlin, M., ed.: CT-RSA. Volume 5473 of Lecture Notes in Computer Science., Springer (2009) 252–264
7. Baena, J., Clough, C., Ding, J.: Square-vinegar signature scheme. PQCRYPTO 2008, LNCS **5299** (2008) 17–30
8. Billet, O., Macario-Rat, G.: Cryptanalysis of the square cryptosystems. ASIACRYPT 2009, LNCS **5912** (2009) 451–486
9. Dubois, V., Fouque, P.A., Shamir, A., Stern, J.: Practical Cryptanalysis of SFLASH. In Menezes, A., ed.: CRYPTO. Volume 4622 of Lecture Notes in Computer Science., Springer (2007) 1–12
10. Shamir, A., Kipnis, A.: Cryptanalysis of the oil & vinegar signature scheme. CRYPTO 1998. LNCS **1462** (1998) 257–266
11. Thomae, E., Wolf, C.: Roots of square: Cryptanalysis of double-layer square and square+. [25] 83–97
12. 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
13. Smith-Tone, D.: On the differential security of multivariate public key cryptosystems. [25] 130–142
14. Patarin, J.: The oil and vinegar algorithm for signatures. Presented at the Dagsthul Workshop on Cryptography (1997)
15. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature verification and message-encryption. Eurocrypt '88, Springer **330** (1988) 419–545
16. Patarin, J.: Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt 88. Crypto 1995, Springer **963** (1995) 248–261
17. Ding, J.: A new variant of the matsumoto-imai cryptosystem through perturbation. In: Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004. (2004) 305–318
18. Fouque, P.A., Granboulan, L., Stern, J.: Differential cryptanalysis for multivariate schemes. EUROCRYPT 2005, LNCS **3494** (2005) 341–353
19. Patarin, J., Goubin, L., Courtois, N.: C^{*}_{-+} and HM: Variations around two schemes of T.Matsumoto and H.Imai. Asiacrypt 1998, Springer **1514** (1998) 35–49
20. Fouque, P.A., Macario-Rat, G., Perret, L., Stern, J.: Total break of the ℓic- signature scheme. PKC 2008, LNCS **4939** (2008) 1–17
21. Ding, J., Dubois, V., Yang, B.Y., Chen, O.C.H., Cheng, C.M.: Could sflash be repaired? Automata, Languages and Programming **4450** (2009) 691–701
22. Ding, J., Yang, B.Y., Cheng, C.M., Chen, O., Dubois, V.: Breaking the Symmetry: a Way to Resist the New Differential Attack. Cryptology ePrint Archive, Report 2007/366 (2007) http://eprint.iacr.org/.
23. Gotaishi, M., Tsujii, S.: Hidden pair of bijection signature scheme. Cryptology ePrint Archive, Report 2011/353 (2011) http://eprint.iacr.org/.
24. Gotaishi, M.: Hidden pair of bijection signature (Part II). Presentation: Rump Session PQCRYPTO 2011 (2011) http://troll.iis.sinica.edu.tw/pqc11/recent.shtml.
25. Yang, B.Y., ed.: Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings. In Yang, B.Y., ed.: PQCrypto. Volume 7071 of Lecture Notes in Computer Science., Springer (2011)