Building Secure Tame-like Multivariate Public-Key Cryptosystems: The New TTS

Bo-Yin Yang[1] and Jiun-Ming Chen[2]

[1] Dept. of Mathematics, Tamkang University, Tamsui, Taiwan*
by@moscito.org
[2] Chinese Data Security, Inc., & Nat'l Taiwan U., Taipei
jmchen@ntu.edu.tw

Abstract. Multivariate public-key cryptosystems (sometimes polynomial-based PKC's or just multivariates) handle polynomials of many variables over relatively small fields instead of elements of a large ring or group. The "tame-like" or "sparse" class of multivariates are distinguished by the relatively few terms that they have per central equation. We explain how they differ from the "big-field" type of multivariates, represented by derivatives of C^* and HFE, how they are better, and give basic security criteria for them. The last is shown to be satisfied by efficient schemes called "Enhanced TTS" which is built on a combination of the Oil-and-Vinegar and Triangular ideas. Their security levels are estimated. In this process we summarize and in some cases, improve *rank-based attacks*, which seek linear combinations of certain matrices at given ranks. These attacks are responsible for breaking many prior multivariate designs.

1 Introduction: Multivariate and Tame-like PKC's

Multivariate public-key cryptosystems (sometimes just multivariates[3]) operate on long vectors over small fields, in contrast to the huge rings and groups of better-known schemes. A typical multivariate PKC over the base field K has a public map comprising three portions. In the notations of [3, 35], we write it as $V = \phi_3 \circ \phi_2 \circ \phi_1 : K^n \to K^m$. The maps $\phi_1 : \mathbf{w} \mapsto \mathbf{x} = \mathsf{M}_1\mathbf{w} + \mathbf{c}_1$ and $\phi_3 : \mathbf{y} \mapsto \mathbf{z} = \mathsf{M}_3\mathbf{y} + \mathbf{c}_3$ are affine in K^n and K^m respectively and usually invertible. We call ϕ_2 the *central map* and the equations giving each y_j in the x_i's the *central equations. The security of the scheme is then based on the NP-hardness [15] in solving a large system of quadratics and difficulty in decomposing V into the components ϕ_i.* The speed of the public map and the size of the keys depend only on m and n. The speed of the private map depends on how fast a preimage for $\phi_2 : \mathbf{x} \mapsto \mathbf{y}$ can be obtained, and key generation on the complexity of ϕ_2. A good quick reference on various multivariates can be found in [33].

* Research supported in part by Taiwan's National Science Council via TWISC (Taiwan Information Security Center at NTUST) project and grant NSC93-M-2115-032-008

[3] also "polynomial-based" along with lattice-based NTU, which differs fundamentally.

C. Boyd and J.M. González Nieto (Eds.): ACISP 2005, LNCS 3574, pp. 518–531, 2005.

Recently there has been renewed interest in multivariate PKC's, and we will

- Characterize tame-like PKC's, a subset of multivariates (Sec. 1). Show that they are efficient and possibly very useful in low-resource deployments.
- Review the security concerns of tame-like PKC's including linear-algebra related attacks (collectively, "rank attacks", Sec. 5), in some cases generalized and improved viz. Sec. 9, modern Gröbner Bases related methods, and others (Sec. 10).
- Give basic criteria for proper design of a tame-like multivariate scheme (Sec. 10). Build (Sec. 4) a scalable sequence of schemes satisfying these conditions using a combination of the triangular and oil-and-vinegar themes.

Note: old version at e-Print archive report 2004/061; full version to be up later.

2 Pros and Cons for Multivariates

For a long time, cryptologists were not very interested in multivariates because traditional PKC's are considered "good enough". The large keys of multivariates also causes problems in key storage, management, and generation for PKI setup and maintenance. Furthermore, the last two decades saw many proposed multivariates broken, so there is some general distrust of multivariates. But multivariate are getting another look because

1. The relative slowness of RSA does affect deployment (e.g., co-processors cost) and some environments are simply too real-time-oriented or resource-poor for RSA (i.e. lower-cost RFID). A multivariate-like structure may do better [14].
2. In some multivariate schemes, keys can be generated blockwise possible in real time on a smart card, which ameliorates the on-card storage problem.
3. Quantum computing may become reality in two decades, bringing a sea change.

The slowness of current progress [30] belies the lack of recent advances in factoring technique, but at CHES 2004, Dr. Issac Chuang reported on QC and estimated less than 2 decades to practicality. RSA and discrete-log based schemes will then be broken by Shor's Algorithm [28], but multivariates are more resistant[4]. Quantum physics can also accomplish a secure key exchange, but so far lacks the functionality of digital signatures. Thus alternative digital signature schemes are being sought.

3 Tame-like Multivariates

In one type of multivariates including the HFE [26] and $C^\star$ families [22, 27], ϕ_2 represents a function in a huge field. They are termed *big-field* or *two-field* [33], and generate keys via an interpolation-based procedure in $\sim n^6$ time [31].

[4] A QC attack with Grover's Algorithm [17] only halves the log-complexity [24].

Lower-powered systems, especially low-end embedded ones, needs to do better. **A multivariate is termed tame-like if its central equations average a small number (vs. $\sim n^2/2$ terms for a random quadratic) of terms** — say $\leq 2n$ each — and **can be inverted quickly**, e.g., faster than evaluating the public map. Since a tame-like map takes only $O(n^2)$ instead of $O(n^3)$ time to evaluate, key generation via interpolation would take at most $O(n^5)$ time. However, we can do even better than that:

Proposition 1. *Keys can be generated for a tame-like multivariate in time $O(n^4)$.*

Proof. Following Imai and Matsumoto [22], we divide the coefficients involved in each public key polynomial into linear, square, and crossterm portions thus:

$$z_k = \sum_i P_{ik} w_i + \sum_i Q_{ik} w_i^2 + \sum_{i<j} R_{ijk} w_i w_j = \sum_i w_i \left[P_{ik} + Q_{ik} w_i + \sum_{i<j} R_{ijk} w_j \right].$$

R_{ijk}, which comprise most of the public key, may be computed as follows (as in [35]):

$$R_{ijk} = \sum_{\ell=n-m}^{n-1} \left[(\mathsf{M}_3)_{k,(\ell-n+m)} \left(\sum_{p\, x_\alpha x_\beta \text{ in } y_\ell} p\ \left((\mathsf{M}_1)_{\alpha i} (\mathsf{M}_1)_{\beta j} + (\mathsf{M}_1)_{\alpha j} (\mathsf{M}_1)_{\beta i} \right) \right) \right] \quad (1)$$

The second sum is over all cross-terms $p\ x_\alpha x_\beta$ in the central equation for y_ℓ. For every pair $i < j$, we can compute at once R_{ijk} for every k in $O(n^2)$ totalling $O(n^4)$. Similar computations for P_{ik} and Q_{ik} take even less time.

Therefore set-up times for a tame-like multivariate be two-orders-of-magnitudes faster than non-tame-like ones. On a low-cost smartcard, on-demand public-key generation from private info ($O(n^2)$ storage) can be done in real time (cf. Tab. 1).

4 Triangular Maps, Tame Maps and the TTS Family

The prototype of tame-like ϕ_2 is the *tame transformation* from algebraic geometry. With dimensions $m \geq n$ over the *base field* K, this is a polynomial map $\phi : K^n \to K^m$, taking $\mathbf{x}$ to $\mathbf{y}$ either affinely ($\mathbf{y} = \mathsf{M}\mathbf{x} + \mathbf{c}$) or in *de Jonquiere* form with $y_1 = x_1$; $y_j = x_j + q_j(x_1, \ldots, x_{j-1})$, $j = 2 \cdots n$;
$y_j = q_j(x_1, \ldots, x_n)$, $j = n+1 \cdots m$. If bijective, it is a *tame automorphism* over K, in which case obviously $m = n$.

On tame transformations, sometimes called *triangular maps*, is based the public-key encryption scheme TTM [23]. This concept was adapted and extended the concept [3] to include **all polynomial maps without a low degree explicit inverse for which an inverse can be found without solving anything higher than linear equations.** We will call term such maps *tame*, and

([3]) **TTS is defined as a multivariate DSS with a tame central map.** For example, with $n = 28$, $m = 20$, ϕ_2 :

$$y_k = x_k + a_k x_{k-8} x_{k-1} + b_k x_{k-7} x_{k-2} + c_k x_{k-6} x_{k-3} + d_k x_{k-5} x_{k-4},\ 8 \leq k \leq 26;$$
$$y_{27} = x_{27} + a_{27} x_{19} x_{26} + b_{27} x_{20} x_{25} + c_{27} x_{21} x_{24} + d_{27} x_{\mathbf{0}} x_{\mathbf{27}};$$

is tame since we can assign any $x_1, \ldots, x_7$ and $x_0 \neq -d_{27}^{-1}$ and find a preimage. We will illustrate with the multivariate signature scheme TTS/2′ that has this central map.

5 Rank-Based Attacks Against Tame-like Multivariates

Many tame-like PKC's were broken through finding linear combinations associated matrices at some given rank. There are three distinct types of these *rank-based attacks*:

The Rank (or Low Rank) Attack [21] seems well-known in other circles before introduced to cryptography by Shamir and Kipnis against HFE.

The Dual (or High) Rank Attack [5] likely first invented by Coppersmith *et al.* Goubin and Courtois somewhat simplified the procedures of the above two attacks against an instance of the encryption scheme TTM [23]. They further expanded their scope to all "TPM" (triangular-plus-minus) systems [16].

Oil-and-Vinegar attacks invented by Kipnis *et al* [19, 20] against OV/UOV schemes.

6 The Rank or Low Rank Attack

Let $q = |K|$, and r be the smallest rank in linear combinations of central equations, which without loss of generality we take to be the first central equation itself. Goubin and Courtois [16] outline how to break TPM in expected time $O(q^{\lceil \frac{m}{n} \rceil r} m^3)$:

1. Take $P = \sum_{i=1}^{m} \lambda_i H_i$, an undetermined linear combination of the symmetric matrices representing the homogeneous quadratic portions of the public keys. [16] did not mention this, but when char $K = 2$ the quadratic portion of z_i cannot be written as $\mathbf{w}^T Q_i \mathbf{w}$, with the matrices Q_i symmetric. However there is still a unique symmetric matrix that can represent z_i, namely $H_i = Q_i + Q_i^T$ [5].

   A quadratic $C_{ab} x_a x_b + C_{cd} x_c x_d + \cdots$ with all indices distinct will have a corresponding symmetric matrix with kernel $\{\mathbf{x} :\ 0 = x_a = x_b = x_c = x_d = \cdots\}$. We will call this the kernel of the quadratic and use the shorthand $\ker y_i$ (or $\ker_{\mathbf{x}} y_i$ to specify what space). With ℓ cross-terms with distinct indices, the rank of the matrix is 2ℓ. Hence $\ker_{\mathbf{x}} y_k = \{\mathbf{x} :\ x_{k-8} = \cdots = x_{k-1} = 0\}$ for TTS/2′.

2. Guess at a random k-tuple $(\mathbf{w}_1, \ldots, \mathbf{w}_k)$ of vectors in K^n, where $k = \lceil \frac{m}{n} \rceil$. Set $P\mathbf{w}_1 = \cdots = P\mathbf{w}_k = \mathbf{0}$ and solve for λ_i via Gaussian elimination. When this is uniquely solvable P is likely the quadratic part of y_1, the first central equation.
3. Assume the matrix corresponding to y_1 has a rank of r, then its kernel (the inverse image $H_1^{-1}(\mathbf{0})$) has dimension $n - r$, hence when we guess at $(\mathbf{w}_1, \ldots, \mathbf{w}_k)$ randomly, they have a probability of at least q^{-kr} to be all in $H_1^{-1}(\mathbf{0})$. This P is the quadratic portion of y_1 and the coefficients λ_i the row of M_3^{-1} (up to a factor).

The Rank Attack should be at its most effective against a signature scheme, as $k = 1$. Obviously, not all multivariates are TPM. However, if a central equation has too few terms then the above works. Further remarks are due in Sec. 9.

Proposition 2 (Time to Find a Vector in any Given Kernel). *Regardless of the form of ϕ_2, if one unique linear combination $H = \sum_{i=1}^{m} \alpha_i H_i$ has the minimum rank r then the algorithm above will always find a vector in* ker$H$ *with an expected time of $\approx q^{kr}\left(m^2(nk/2 - m/6) + mn^2k\right)$ field multiplications.*

7 The Dual Rank or High Rank Attack

The Rank Attack finds a large kernel shared by a small subset of the space spanned by the matrices H_i. The converse, to find a small kernel shared by a many linear combinations of the H_i, may be called a Dual Rank attack or High Rank attack. It happens when a variable appears in too few central equations.

In Birational Permutation Schemes, the last central variables x_n appears the cross-terms of only one equation. This critical weakness [5] means we can find linear combinations $\sum_i \alpha_i z_i$ whose kernels share a non-empty intersection. Coppersmith, Stern, and Vaudenay [5] then construct an ascending chain of kernels in the matrix algebra over a ring without needing to search. In [16], a simpler version of the dual rank attack was run via searching, and we can describe this as follows:

Without loss of generality, let the fewest number of appearances of all variables in the cross-terms of the central equations be the last variable x_{n-1} appearing u times.

In TTS/2$'$, this is x_{27}, which only appears in y_{27}. So whenever $\alpha_{27} = 0$, the subspace $U = \{x_0 = \cdots = x_{26} = 0\} \subset \ker \sum_{i=8}^{27} \alpha_i Q_i$. (Here H_i and Q_i are as in Sec. 6.) If we denote by m_{ij} the (i,j)-entry of M_3, then almost every (H_i, H_j) pair has a linear combination with a kernel containing the same subset U. In general, with almost any $(u+1)$-subset picked from the H_i, a unique linear combination of these matrices has a kernel containing $U = \{\mathbf{x} : x_0 = \cdots = x_{n-2} = 0\}$. We try to find U.

1. Form an arbitrary linear combination $H = \sum_i \alpha_i H_i$. Find $V = \ker H$.
2. When $\dim V = 1$, set $(\sum_j \lambda_j H_j)V = \{\mathbf{0}\}$ and check if the solution set $\hat{V}$ of the (λ_i) form a subspace dimension $m - u$. Because a matrix in $K^{n \times n}$

can have at most n different eigenvalues, less than n/q of the time we would need to do this.

3. With probability q^{-u} we have $V = U$. The cost of one trial is bounded by one elimination plus possible testing, so total cost is $\left[mn^2 + \frac{n^3}{6} + \frac{n}{q}(m^3/3 + mn^2)\right] q^u$. We can do with a little more than $\left(un^2 + \frac{n^3}{6}\right) q^u$ field multiplications if we only consider linear combinations of $(u+1)$ of the matrices H_i, and are not too unlucky.

From this subspace, we can find bigger kernels. [5] does this through taking a sequence of derivatives. For TPM as for TTS/2', the next bigger kernel (which is $U' = \{x_0 = x_1 = \cdots = x_{25} = 0\}$) can be found by examing subspaces of V, which will get us U' with probability $1/q$. So for TTS/$2'$, the flaw is severe and cryptanalysis is swift.

8 Unbalanced Oil-and-Vinegar Attacks and a Simplification

An (Unbalanced) Oil-and-Vinegar attack [19, 21] on a multivariate takes place if we may partition the variables x_i into sets $\mathcal{O}$ and $\mathcal{V}$, such that there is no cross-term with both variable in $\mathcal{O}$. The two sets are called the *oil* and *vinegar* variables respectively.

Suppose a maximal set of vinegar variables is at least size v, then Kipnis *et al* find the oil subspace (the space spanned by the oil variables) by looking at certain linear combinations that become degenerate. The average time complexity is $q^{2v-n-1}(n-v)^4$.

TTS/$2'$ fits this description with $v = 14$ ($\mathcal{V}$ is the variables with even indices). An OV or UOV attack in essence let the attacker eliminate some variables. This often let the attacker get around whatever devices that defend against a rank attack. In [11], Ding and Yin cryptanalyze the instance of TTS given in [35] on such an oversight. They used a sequence of fairly intricate manuevers after the UOV stage. In this and certain other cases, we could make cryptanalysis using the UOV attack a little simpler, as below:

Proposition 3 (Unbalanced Oil-and-Vinegar with Guessing). *If a multivariate digital signature scheme with a public map $K^n \to K^m$ have minimal vinegar variable set size v, then a solution may be found in* $\max(q^{2v-n-1}(n-v)^4, q^{m+v-n}(n-v)^3/3)$ *multiplications, regardless of other structure.*

Proof. Follow the steps in [19] to distill the oil subspace. Now, if it were really an UOV scheme, we would be able to find a solution in time $(n-v)^3/3$ (i.e., time for one Gaussian elimination). However, this requires us to be able to guess at v variables. Since we can only fix $n-m$ variables and expect to find a solution, on average we rate $q^{v-(n-m)}$ random guesses during the the cryptanalysis.

9 More About Rank-Based Attacks

Rank-based attacks are important considerations against tame-like (and perhaps other) multivariates. The various authors already did a fine job of presenting the methods. One notable correction we would like to make is the estimate for dual-rank attacks in [16] (unquestioned by later works) is given as $n^6 q^u$ when it should be $\left(un^2 + \frac{n^3}{6}\right) q^u$ (field multiplications) as given in Sec. 7. It is easy to fall to any of these three attacks if one is careless, e.g., in the RSE(2)PKC and RSSE(2)PKC schemes of Kasahara-Sakai that falls ([32]) to an almost verbatim attack from [16]. These are generalizations of TPM that C. Wolf *et al* call Stepwise Triangular Schemes (STS). As discussed in [33], the basic STS constructions cannot be used alone. We may also surmise that to depend fundamentally on guessing can be a very bad idea for non-big-field multivariates.

There is one potential improvement to the Rank Attack that has not been mentioned by previous investigators. In Sec. 6 we assume y_1 to have the smallest rank r; other y_i and even many linear combinations of the y_i (hence the H_i) with different kernels can also share the same minimum rank r. For example, in TTS/2′, for non-zero α, the ranks of $y_i + \alpha y_{i+1}$ and $y_i + \alpha y_{i+2}$ are both 8. So is $y_i + \alpha y_{i+1} + \beta y_{i+2}$ if $\alpha^2 a_{i+1} b_{i+1} d_{i+1} = \beta(c_i a_{i+1} d_{i+2} + b_i d_{i+1} a_{i+2})$. That is at least $10,000$ total combinations. We call this *interlinks*. When the largest kernels and equations interlink, the Rank Attack can be made faster by the *crawl* process below. Odds of finding a kernel vector in the [16] attack is then essentially multiplied by the number of distinct kernels.

Proposition 4 (Interlinked Kernels). *If there are c kernels of codimension r that interlink, then we can cryptanalyze in an expected $q^{kr} kmn(m+n)/c$ field multiplications.*

We take again as the example TTS/2′. For simplicity, let all coefficients be 1, then

$$\begin{aligned}
\ker y_8 &= \{\mathbf{x} : x_0 = x_1 = \cdots = x_7 = 0\};\\
\ker y_9 &= \{\mathbf{x} : x_1 = x_2 = \cdots = x_8 = 0\};\\
\ker y_{10} &= \{\mathbf{x} : x_2 = x_3 = \cdots = x_9 = 0\};\\
\ker(y_8 + \alpha y_9) &= \{\mathbf{x} : x_1 = x_3 = x_5 = x_7 = 0,\ x_0 : x_2 : x_4 : x_6 : x_8 = \alpha^4 : \alpha^3 : \alpha^2 : \alpha : 1\};\\
\ker(y_8 + \alpha y_{10}) &= \{\mathbf{x} : x_2 = x_3 = x_6 = x_7 = 0,\ x_0 : x_4 : x_8 = x_1 : x_5 : x_9 = \alpha^2 : \alpha : 1\};
\end{aligned}$$

With generic coefficients, there will be a three-term combination that has rank 8 exists (here it does not). Its kernel would be vectors $\mathbf{x}$ with $x_4 = x_5 = 0$ and $x_0 : x_2 : x_6 : x_8$ and $x_1 : x_3 : x_7 : x_9$ in fixed ratios. We now proceed along these steps:

1. Run the algorithm of Sec. 6 to find a kernel vector $\mathbf{u}$ and its associated quadratic $z = \sum_i \lambda_i z_i$ of rank 8. Verify $U = \ker z$ to be of codimension 8, and find a basis for U. Given any rank 8 kernel when $(m, n) = (20, 28)$, according to Prop. 2, we should need $256^8 \cdot \left[20^2 \cdot (28/2 - 20/6) + 20 \cdot 28^2\right] \approx 2^{78}$ field

multiplications (or $\approx 2^{71}$ 3DES units) to find a vector in that kernel. But kernels of the 10000+ rank-8 forms are mostly distinct, so we expect only $\sim 2^{65}$ multiplications. There being only 20 forms y_i and about 5000 forms $y_i + \alpha y_{i+1}$ (and almost as many $y_i + \alpha y_{i+2}$), the first vector yielding a codimension-8 kernel will likely come from a mixed form rather than from one of the y_i's, and we therefore need to isolate y_i's.

2. Repeat the same algorithm but we restrict test vectors $\mathbf{w}$ to U, and only accept a tested vector if it lies in more than one kernel, i.e., we solve $\sum_i \lambda_i H_i \mathbf{v} = 0$, finding a basis $(\hat{y}_i)_{i=1\cdots s}$ in quadratic forms, and keep $\mathbf{v}$ if the solution space is of dimension two or higher. Let this solution space be expressed in quadratic forms as $v \in \ker(\sum_{\ell=1}^{s} \alpha_\ell \hat{y}_\ell)$ for $s \geq 2$. We expect the dimension s to be 2 or 3. If we find two distinct sets of results ($\mathbf{v}$ and $(\hat{y}_i)$) in say 5000 tests, then we have just found a y_i for some $9 \leq i \leq 25$, and the results would match the forms $\mathrm{span}(y_i, y_{i\pm 1})$.

   If, as is normally the case, we find only one solution space for λ_i's, then that must be $\mathrm{span}(y_i, y_{i+1})$ or $\mathrm{span}(y_i, y_{i+1}, y_{i+2})$ depending on its dimension. As an example, assume that we initially hit a vector that lies in the kernel U of $y_8 + \alpha y_9$ and no other quadratic form. With probability 2^{-8} a random vector $\mathbf{v} \in U$ will lie in $\ker y_8 \cap \ker y_9 = \{\mathbf{x} : x_0 = x_1 = \cdots = x_8 = 0\}$. Ditto for any $z = y_i + \alpha y_{i+1}$.

   Similarly if $z = y_i + \alpha y_{i+2}$, or any three-term combination that has rank 8, the odds of finding a vector $\mathbf{v}$ in more than one kernel is 2^{-16}, and we would find $(\ker y_i) \cap (\ker y_{i+1}) \cap (\ker y_{i+2})$. *This step should take little time in this step, equivalent to trying 2^{16} random vectors $\mathbf{w}$ in Sec. 6.*
3. For all the materially different quadratic forms f_i that we locate we find the kernels U_i associated with them. There will be either 257 or $256^2 + 256 + 1 = 65793$ distinct linear combinations. Among the forms f_i we should have either two or three of the y_i's. Repeat the search in each U_i as above to find kernels corresponding to the y_i's. *Checking 2^{12} vectors from each of $\sim 2^{16}$ kernels U_i take $< 2^{42}$ multiplications.*
4. Say we have found y_9, since $y_9 = x_9 + a_9 x_1 x_8 + b_9 x_2 x_7 + c_9 x_3 x_6 + d_9 x_4 x_5$, we should be able to identify one linear combination of the w_i as x_9 and 8 others as $x_1, \ldots, x_8$. Indeed finding any y_i should give quickly all y_j and x_j where $j < i$. Incremental search will then locate all forms y_i and x_i, i.e. matrices M_1 and M_3.

In TTS/2′, a kernel vector should be found in $\approx 2^{63}$ multiplications ($\approx 2^{57}$ 3DES blocks[5]). Experiments on smaller analogues to TTS/2′ schemes was in reasonable accord with the cryptanalysis described above. There are other possibilities, viz.:

Proposition 5 (Accumulating Kernels). *With equations of rank 2 with sole cross-terms are $x_i x_{j_1}$, $x_i x_{j_2}, \ldots, x_i x_{j_s}$, then any vector with $x_i = 0$ becomes a kernel vector.*

[5] NESSIE [24] measures security in 3DES blocks and we count multiplications in $\mathrm{GF}(2^8)$. To calibrate, we use NESSIE's performance data [24], and compare against our runs. We find one 3DES block to be $\approx 2^6$ multiplications.

Remark: Such equations would effectively have a minrank of 1. A similar situation occurs in multi-term combinations. This implies that TTM is very hard to secure – there can only be rank-4+ equations and not too many interlinks.

10 Other Attacks and Security Criteria for a Multivariate

What non-rank-based attacks are there? There are no other generic attack aside from[6] Linearization-like Methods, i.e. XL [7] and Gröbner Bases Algorithms [12, 13]. There are also attacks tailored against specific schemes. The most important is Bilinear Relations [25], used against C^*. It only works if the central maps are rank-2 in some embedding field. Neither this nor any other specific attacks work against the tame-like systems that we will construct below (see [3]).

Proposition 6. *To build a tame-like Digital Signature Scheme needing a security of C:*

1. *If k linear combinations of central equations share a minimal rank r, then we need*
$$q^r \cdot \left(m^2(n/2 - m/6) + mn^2\right)/k \geq C. \tag{2}$$
Here usually $r = 2\ell$ where ℓ is the smallest number of cross-terms in an equation.
2. *If every central variable appears in at least u central equations, then*
$$q^u \left(un^2 + n^3/6\right) \geq C. \tag{3}$$
3. *Let v be the size of the smallest maximal set A of indices $0 \leq i < n$ such that every cross-term in the central map has at least one index in A, then we require*[7]
$$q^{2v-n-1}(n-v)^4 \geq C. \tag{4}$$
4. *Let $D_0 = D_0(m,k) := \min\{D : [t^D]\ \left((1-t)^{k-1}\,(1+t)^m\right)$, $T := \binom{m-k+D_0}{D_0}$, then ($c_0$, c_1, γ are constants, ω is the order of the equation-solver):*
$$\min_k q^k \cdot m^{\gamma_0} T^{\omega}(c_0 + c_1 \lg T) \geq C. \tag{5}$$
5. *There should not be any over-determined subsystems in the central equations.*[8]

The reader will need to refer to [1, 8, 34] to understand how Eq. 5 came about. The executive summary of the formula is the security when an attacker guesses at an optimal number of variables then runs either the Gröbner Bases algorithm

[6] The search methods of [6] is only useful against Signature Schemes over small fields [3].

[7] Within this restraint, lower v means higher FXL/F$\mathbf{F_5}$ complexity, see [34].

[8] Otherwise XL-type attacks can function at a much lower degree. This ([18]) breaks up more careless constructs like the latest version of TRMC [29].

$\mathbf{F_5}$ [12] or the FXL algorithm [7] using a sparse solver with speed comparable to Lanczos.

We do not know for sure what the parameters should be in Eq. 5. The theoretical best limit for $\mathbf{F_5}$ is given by ([34]) is roughly $c_0 = 4$, $c_1 = \frac{1}{4}$, $\gamma = 2$, $\omega = 2$ when counting field multiplications. To our knowledge no one comes close. Indeed, all commercially available software (including MAGMA, of the University of Sydney, which is reputed to be the best) have $\omega = 3$, according to many tests. A rough implementation of FXL with a sparse solver can currently do about $c_0 \approx 20$, $c_1 \approx 1$, $\gamma \approx 4$, $\omega = 2$.

11 Building Example Schemes: Enhanced TTS

What fast tame-like signature scheme would we come up with in full knowledge of what we now understand, to get to a complexity of 2^{80} 3DES blocks (2^{86} multiplications)?

1. The hash needs to be 160-bit, or $m \geq 20$ (birthday attacks), and $n \geq m$.
2. We need $m \geq 20$ for XL/$\mathbf{F_5}$ attacks (we would need $m \geq 22$ if $q = 2^7$).
3. We need $r > 8$, so there must be at least 5 independent cross-terms in each equation, probably 6 or 7 to account for the "crawl" of Sec. 9.
4. We do not want n too large because that adds to the key length and running times, and we may open ourselves to searching attacks cf. [3].
5. We need $u \geq 9$, so every variable must appear in at least 9 equations.

The following seems to be reasonable approaches to ensure the above:

- We choose not to search. Therefore we are restricted to an "Oil-and-Vinegar"-like approach of taking random values for some variables and solving for the rest.
- We need an initial segment with 6 or 7 cross-terms per equation. This will be solved as a linear system when the "vinegar-like" variables have been assigned.
- We need a final segment in at least the last 9 variables.
- One vinegar variable can provide one cross term per equation in the initial segment.
- If possible, the two systems we solve should be of equal dimension.

So we may do a signature scheme with the following central map ϕ_2:

$$y_i = x_i + \sum_{j=1}^{7} p_{ij} x_j x_{8+(i+j \bmod 9)},\ i = 8 \cdots 16;$$

$$\begin{aligned} y_{17} = x_{17} &+ p_{17,1} x_1 x_6 + p_{17,2} x_2 x_5 + p_{17,3} x_3 x_4 \\ &+ p_{17,4} x_9 x_{16} + p_{17,5} x_{10} x_{15} + p_{17,6} x_{11} x_{14} + p_{17,7} x_{12} x_{13}; \end{aligned}$$

$$\begin{aligned} y_{18} = x_{18} &+ p_{18,1} x_2 x_7 + p_{18,2} x_3 x_6 + p_{18,3} x_4 x_5 \\ &+ p_{18,4} x_{10} x_{17} + p_{18,5} x_{11} x_{16} + p_{18,6} x_{12} x_{15} + p_{18,7} x_{13} x_{14}; \end{aligned}$$

$$\begin{aligned} y_i = x_i &+ p_{i,0} x_{i-11} x_{i-9} + \sum_{j=19}^{i-1} p_{i,j-18}\ x_{2(i-j)-(i \bmod 2)}\ x_j + p_{i,i-18} x_0 x_i \\ &+ \sum_{j=i+1}^{27} p_{i,j-18}\ x_{i-j+19}\ x_j,\ i = 19 \cdots 27. \end{aligned}$$

This is the [35] central map modified to avoid the UOV attack. Of course, we need to show that the new variant can scale up if our estimate is somewhat off, or to meet future, higher security requirements. We will discuss this next in Sec. 12. Note that our ϕ_2 above can be inverted reliably as follows:

1. Assign $x_1, \ldots, x_7$ and try to solve the first nine equations for x_8 to x_{16}.
2. If we fail to solve the first system of equations, just redo everything from scratch. The probability is around 255/256 that this system can be solved. As the determinant of the first system (for any x_1 through x_6) is a degree-9 polynomial in x_1 there can only be at most 9 choices of x_1 to make the first system degenerate, so the odds to solve this system is at least 247/256 and we will eventually hit upon a solution.
3. Solve serially for x_{17} and x_{18} using the next two equations (y_{17} and y_{18}).
4. Assign a random x_0 and try to solve the second system for x_{19} through x_{27}. Again, there will be at most nine x_0 that makes the determinant of the second system zero. So, if the first attempt to solve it fails, try other x_0 until a solution is found.

We will call this TTS/5 or Enhanced TTS (20,28). Its operates as follows:

To Generate Keys: Assign non-zero random values in $K = \mathrm{GF}(2^8)$ to parameters p_{ij}; generate random nonsingular matrices $\mathsf{M}_1 \in K^{28\times 28}$ and $\mathsf{M}_3 \in K^{20\times 20}$ (usually via LU decomposition) and vector $\mathbf{c}_1 \in K^{28}$. Compose $V = \phi_3 \circ \phi_2 \circ \phi_1$; assign $\mathbf{c}_3 \in K^{20}$ so that V has no constant part. Save quadratic and linear coefficients of V as public key (8680 bytes). Find M_1^{-1}, M_3^{-1}; save them with $\mathbf{c}_1$, $\mathbf{c}_3$, and the parameters p_{ij} as the private key (1399 bytes).

To Sign: From the message M, first take its digest $\mathbf{z} = H(M) \in K^{20}$, then compute $\mathbf{y} = \mathsf{M}_3^{-1}(\mathbf{z} - \mathbf{c}_3)$, then compute a possible $\mathbf{x} \in \phi_2^{-1}(\mathbf{y})$ as above: Our desired signature is $\mathbf{w} = \mathsf{M}_1^{-1}(\mathbf{x} - \mathbf{c}_1)$. Release $(M, \mathbf{w})$.

To Verify: On receiving $(M, \mathbf{w})$, compute $\mathbf{z} = H(M)$ and match with $V(\mathbf{w})$.

Scheme	Signature	PublKey	SecrKey	Setup	Signing	Verifying
RSA-PSS	1024 bits	128 B	320 B	2.7 sec	84 ms	2.0 ms
ECDSA	326 bits	48 B	24 B	1.6 ms	1.9 ms	5.1 ms
ESIGN	1152 bits	145 B	96 B	0.21 sec	1.2 ms	0.74 ms
QUARTZ	128 bits	71.0 kB	3.9 kB	3.1 sec	11 sec	0.24 ms
SFLASH^{v2}	259 bits	15.4 kB	2.4 kB	1.5 sec	2.8 ms	0.39 ms
TTS(20,28)	224 bits	8.6 kB	1.3 kB	1.5 ms	51 μs	0.11 ms
TTS(24,32)	256 bits	13.4 kB	1.8 kB	2.5 ms	67 μs	0.18 ms

Table 1. TTS and NESSIE round 2 candidate signature schemes on a 500MHz P3

12 Scaling Up Enhanced TTS

We can scale up Enhanced TTS to provide for a security of $C \gtrsim 2^{16k}$. This sequence of TTS instances we will call the “odd sequence” because u is odd. We have (for $\ell \geq 4$) the $(m,n) = (4\ell, 6\ell-2)$, with security parameters $(u,r,v) = (2\ell-1, 4\ell-6, 4\ell-1)$

$$\begin{aligned}
y_i &= x_i + \textstyle\sum_{j=1}^{2\ell-3} p_{ij} x_j x_{2\ell-2+(i+j+1 \bmod 2\ell-1)}, \text{ for } 2\ell-2 \leq i \leq 4\ell-4;\\
y_i &= x_i + \textstyle\sum_{j=1}^{\ell-2} p_{ij} x_{i+j-(4\ell-3)} x_{i-j-2\ell)}\\
&\qquad + \textstyle\sum_{j=\ell-1}^{2\ell-3} p_{ij} x_{i+j-3\ell+6} x_{i+\ell-5-j},\ i = 4\ell-3 \text{ or } 4\ell-2;\\
y_i &= x_i + p_{i0} x_{i-2\ell+1} x_{i-2\ell-1} + \textstyle\sum_{j=4\ell-1}^{i-1} p_{i,j-(4\ell-2)} x_{2(i-j)-(i \bmod 2)} x_j\\
&\qquad + p_{i,i-(4\ell-2)} x_0 x_i + \textstyle\sum_{j=i+1}^{6\ell-3} p_{i,j-(4\ell-2)} x_{4\ell-1+i-j} x_j,\\
&\qquad \text{for } 4\ell-1 \leq i \leq 6\ell-3.
\end{aligned}$$

To account for more optimistic estimates for FXL/F$\mathbf{F_5}$, there is a different sequence of Enhanced TTS instances with the same Rank Attack estimates. These instances are called the "even sequence" because the parameter u is even. In ϕ_2 below, we have $(m,n) = (4\ell, 6\ell-4)$, with security parameters $(u,r,v) = (2\ell-2, 4\ell-10, 4\ell-2)$.

$$\begin{aligned}
y_i &= x_i + \textstyle\sum_{j=1}^{2\ell-5} p_{ij} x_j x_{2\ell-4+(i+j+1 \bmod 2\ell-2)}, \text{ for } 2\ell-4 \leq i \leq 4\ell-7;\\
y_i &= x_i + \textstyle\sum_{j=1}^{\ell-4} p_{ij} x_{i+j-(4\ell-6)} x_{i-j-(2\ell+1)}\\
&\qquad + \textstyle\sum_{j=\ell-3}^{2\ell-5} p_{ij} x_{i+j-3\ell+5} x_{i+\ell-4-j}, \text{ for } 4\ell-6 \leq i \leq 4\ell-3;\\
y_i &= x_i + p_{i0} x_{i-2(\ell+1)} x_{i-2(\ell-1)} + \textstyle\sum_{j=4\ell-2}^{i-1} p_{i,j-(4\ell-3)} x_{2(i-j)-(i \bmod 2)} x_j\\
&\qquad + p_{i,i-(4\ell-3)} x_0 x_i + \textstyle\sum_{j=i+1}^{6\ell-5} p_{i,j-(4\ell-3)} x_{4\ell-2+i-j} x_j,\\
&\qquad \text{for } 4\ell-2 \leq i \leq 6\ell-5.
\end{aligned}$$

This ϕ_2 gives about $2^{16}\times$ higher FXL/F$\mathbf{F_5}$ complexity for corresponding instances. The performance of Enhanced TTS $(24,32)$ is also given in Tab. 1.

Remark: A program for finding maximum cliques can verify that the UOV-attack parameter v is as given above. We have no space to explain the design.

We can estimate ϕ_2^{-1} to do $\approx 6k^2(k+2)$ multiplications for small k. This almost equals the work done in matrices M_1 and M_3 at $m = 20$, $n = 28$, and will overtake them when m increases. We further know that asymptotically as k increases, the dimensions n and m to build a TTS instance or another tame-like scheme with security level 2^{16k} both increase linearly (cf. [34]). Thus, time cost of a TTS-like signature scheme goes up roughly with k^ω, where $2 < \omega \leq 3$ is the order of an elimination. Private map timings for RSA and ECC also increase between the quadratic and cubic to size. So the Triangular+OV construction will remain hundreds of times faster than RSA at comparable security levels. Table 2 gives this comparison. Timings on an 8051-compatible is essentially the same as in [35] and maintains a good lead over comparable schemes.

13 Discussions and Conclusion

There is recently a small resurgence of interest in multivariates, with perturbed variations of HFE [10] and C^* and the non-big-field signature schemes TRMS

m	n	PubKey	SecKey	Rank	Dual Rank	FXL	RSA bits	ECC bits	*i*8051 keygen	*i*8051 sign	*i*8051 code
20	28	8680 B	1399 B	2^{120}	2^{80}	2^{81}	≥ 1024	144	78.5s	170ms	1.6kB
24	32	13440 B	1864 B	2^{122}	2^{88}	2^{93}	≥ 1536	160	134s	227ms	1.6kB
28	38	21812 B	2594 B	2^{154}	2^{105}	2^{104}	≥ 2560	192	$\sim$ 300s	$\sim$ 500ms	$\sim$ 2kB
32	44	33088 B	3444 B	2^{186}	2^{121}	2^{115}	≥ 4096	224			
36	50	47700 B	4414 B	2^{220}	2^{138}	2^{133}	≥ 6144	256			

Table 2. Security Estimates of TTS instances, (m, n) = hash and signature sizes

([4], this resembles a tame-like system) and Rainbow [9], essentially a pre-sparsified version of TTS. This is a welcome development, obviously.

At the moment there are no serious reductionist "proof of security" study for multivariates. In that context, We have explained how the central map can affect the security under rank-based attacks and showed how combining the oil-and-vinegar and triangular approaches leads to tame-like signature schemes that are less susceptible to attack on rank.

Tame-like schemes are very fast. The Enhanced TTS instances given here needs no co-processor to run on a really low-end smart card [35]. There is however much research to be done before sparse variants can gain wide currency and trust.

References

1. M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang, *Asymptotic Expansion of the Degree of Regularity for Semi-Regular Systems of Equations*, to be presented MEGA'05.
2. A. Braeken, C. Wolf and B. Preneel, *A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes*, CT-RSA'05, LNCS 3376, pp. 29–43.
3. J.-M. Chen and B.-Y. Yang, *A More Secure and Efficacious TTS Scheme*, ICISC'03, LNCS 2971, pp. 320-338.
4. C.-Y. Chou, Y.-H. Hu, F.-P. Lai, L.-C. Wang, and B.-Y. Yang, *Tractable Rational Map Signature*, PKC'05, LNCS 3386, pp. 244–257.
5. D. Coppersmith, J. Stern, and S. Vaudenay, *Attacks on the Birational Permutation Signature Schemes*, Crypto'93, LNCS 773, pp. 435–443.
6. N. Courtois, L. Goubin, W. Meier, and J. Tacier, *Solving Underdefined Systems of Multivariate Quadratic Equations*, PKC'02, LNCS 2274, pp. 211–227.
7. N. Courtois, A. Klimov, J. Patarin, and A. Shamir, *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations*, Eurocrypt'00, LNCS 1807, pp. 392–407.
8. C. Diem, *The XL-algorithm and a conjecture from commutative algebra*, Asiacrypt'04, LNCS 3329, pp. 338–353.
9. J. Ding and D. Schmidt, *Rainbow, a new multivariate polynomial signature system*, to appear at ACNS'05.
10. J. Ding and D. Schmidt, *Cryptanalysis of HFEv and Internal Perturbation of HFE*, PKC'05, LNCS 3386, pp. 288–301.
11. J. Ding and Z. Yin, *Cryptanalysis of TTS and Tame-like Multivariable Signature Schemes*, presentation at IWAP'04.

12. J.-C. Faugère, *A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5)*, Proceedings of ISSAC, ACM Press, 2002.
13. J.-C. Faugère and A. Joux, *Algebraic Cryptanalysis of Hidden Field Equations (HFE) Cryptosystems Using Gröbner Bases*, Crypto 2003, LNCS 2729, pp. 44-60.
14. M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, *Strong Authentication for RFID Systems Using the AES Algorithm*, CHES '04, LNCS 3156, pp. 357–370.
15. M. Garey and D. Johnson, *Computers and Intractability, A Guide to the Theory of NP-completeness*, Freeman and Co., 1979, p. 251.
16. L. Goubin and N. Courtois, *Cryptanalysis of the TTM Cryptosystem*, Asiacrypt'00, LNCS 1976, pp. 44–57.
17. L. K. Grover, *A fast quantum mechanical algorithm for database search*, Proc. 28th Annual ACM Symposium on the Theory of Computing, (May '96) pp. 212–220.
18. A. Joux, S. Kunz-Jacques, F. Muller, P.-M. Ricordel, *Cryptanalysis of the Tractable Rational Map Cryptosystem*, PKC'05, LNCS 3386, pp. 258–274.
19. A. Kipnis, J. Patarin, and L. Goubin, *Unbalanced Oil and Vinegar Signature Schemes*, Crypto'99, LNCS 1592, pp. 206–222.
20. A. Kipnis and A. Shamir, *Cryptanalysis of the Oil and Vinegar Signature Scheme*, Crypto'98, LNCS 1462, pp. 257–266.
21. A. Kipnis and A. Shamir, *Cryptanalysis of the HFE Public Key Cryptosystem*, Crypto'99, LNCS 1666, pp. 19–30.
22. T. Matsumoto and H. Imai, *Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption*, Eurocrypt'88, LNCS 330, pp. 419–453.
23. T. Moh, *A Public Key System with Signature and Master Key Functions*, Communications in Algebra, 27 (1999), pp. 2207–2222.
24. NESSIE project, `www.cryptonessie.org`.
25. J. Patarin, *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88*, Crypto'95, LNCS 963, pp. 248–261.
26. J. Patarin, *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms*, Eurocrypt'96, LNCS 1070, pp. 33–48.
27. J. Patarin, N. Courtois, and L. Goubin, *FLASH, a Fast Multivariate Signature Algorithm*, CT-RSA'01, LNCS 2020, pp. 298–307. Update at [24].
28. P. W. Shor, *Algorithms for quantum computation: Discrete logarithms and factoring*, Proc. 35th Ann. Symp. on Foundations of Comp. Sci., IEEE Comp. Soc. Press (1994), pp. 124-134.
29. L. Wang, *Tractable Rational Map Cryptosystem*, see ePrint 2004/046.
30. E. Weisstein, *RSA-576 Factored*, `mathworld.wolfram.com/news/2003-12-05/rsa`
31. C. Wolf, *Efficient Public Key Generation for Multivariate Cryptosystems*, Int'l Workshop on Cryptographic Algorithms and their Uses 2004, pp. 78–93, also ePrint 2003/089.
32. C. Wolf, A. Braeken, and B. Preneel, *Efficient Cryptanalysis of RSE(2)PKC and RSSE(2)PKC*, SCN '04, LNCS 3352, pp. 294–309.
33. C. Wolf and B. Preneel, *Taxonomy of Public-Key Schemes based on the Problem of Multivariate Quadratic Equations*, ePrint 2005/077.
34. B.-Y. Yang and J.-M. Chen, *All in the XL Family: Theory and Practice*, ICISC'04, LNCS 3506, pp. 67–86.
35. B.-Y. Yang, J.-M. Chen, and Y.-H. Chen, *TTS: High-Speed Signatures from Low-End Smartcards,*, CHES '04, LNCS 3156, pp. 371-385.