Rank Attacks and Defence in Tame-Like Multivariate PKC's

Bo-Yin Yang
Tamkang University
Tamsui, Taiwan
by@moscito.org

Jiun-Ming Chen
Chinese Data Security, Inc.
and Nat'l Taiwan U., Taipei, Taiwan
jmchen@math.ntu.edu.tw

February 23, 2004

Abstract

Partly in response to the slowness of traditional Public-Key Cryptosystems, multivariate PKC's were born. However, recent attention has mostly been focused on C^{*-} and HFE variants, in which the vector variables really represent an element in a much larger field. This phenomenon may be related to an article by Goubin and Courtois attacking "Triangular-Plus-Minus" (TPM) schemes, a class of multivariates with low expansion rates. They leave the perception that no fast "true" (with really independent variables) multivariate PKC should even be attempted. This impression is compounded by cryptanalysis of schemes with higher expansion rates (e.g. Oil-and-Vinegar).

We believe that TPM (and attacks thereupon) does not cover all pertinent true multivariate PKC's. We will term "tame-like" the multivariate PKC's whose central map has an easy inverse and relatively few terms per equation. Tame-like PKC's, a superset of TPM, have both fast private maps and short set-up times. Unfortunately, the same traits may also open them up to attacks relating to rank, what we will term "rank attacks". Here we study in detail the two attacks in the Goubin-Courtois paper — one may be called "high-rank" and one "low-rank". The former seems to have been first used by Coppersmith *et al*, the latter in a variety of earlier literature.

The TTS (Tame Transformation Signatures) family of digital signature schemes lies at this cusp of contention. Previous instances of TTS, (proposed at ICISC '03) claim good resistance to known attacks. FXL, the best previously-considered attack, cryptanalyzes TTS/$2'$ and TTS/4 in $> 2^{80}$ equivalent AES blocks under a very minimal estimate.

Inattention to rank creates vulnerabilities, however. We show that the innate structure of current TTS constructions (TTS/4 and TTS/$2'$) exacerbates the security concern of rank, and show two different cryptanalysis in $\leq 2^{57}$ AES units. A few other constructs also share the same liability.

A suitable equilibrium between speed and security must be struck. We suggest a generic way to craft tame-like PKC's more resistant to rank attacks. As an example, we build TTS variant in similar dimensions for which rank attack takes $> 2^{80}$ AES units, while remaining very fast and as resistant to other attacks. The proposed TTS variants can be scaled for better security.

In short: we show that Rank attacks can be used on the wider class of tame-like PKC's, sometimes even better than it was previously described. However, this is relativized by the realization that we can build tame-like multivariate PKC's that are adequately resistant, so the general theme still seem viable compared to more traditional alternatives.

1 Introduction

Of all public-key cryptosystems, RSA still "rules" all PKC some 30 years after public-key cryptography was invented ([17]). It is somewhat unfortunate, as due to current advances in cryptography like number field sieves ([7, 41]), secure applications of RSA requires ever-longer keys. This negatively affects the execution speed and cost of deployment. This paper describes an episode of the usual balancing act as a cryptologist veers between requirements in speed and security.

Multivariate PKC was introduced as an alternative to cryptosystems with large algebraic structures. A typical multivariate PKC (following notations of [9]) has a public map $V = \phi_3 \circ \phi_2 \circ \phi_1 : K^n \mapsto K^m$. 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 invertible and affine in K^n and K^m respectively, where K is the base field. *The security of the scheme is then based on the NP-hardness* ([21]) *in solving a large system of quadratic equations and the decomposition of V into components ϕ_1, ϕ_2, and ϕ_3.* Preimages for $\phi_2 : \mathbf{x} \mapsto \mathbf{y}$ are presumed available, but the speed of the private map depends on how fast this inversion can be. The speed of the public map and the size of the keys depends only on m and n, and key generation on how quickly ϕ_2 can be evaluated.

The two best-known multivariate PKC's, SFLASH^{v2} ([38]) and QUARTZ ([37]) descend from Matsumoto-Imai's $C^\star$ ([27]) and Patarin's HFE ([36]) respectively. Both second-round NESSIE ([32]) digital signature scheme candidates were designed by the team of Patarin, Goubin, and Courtois. The former was eventually recommended for low-cost smart cards. Alas, the security of these candidates is under siege by newer developments,[1] and their speed and key sizes can still use some improvement.

In view of the above, it seems natural to investigate alternative possible fast PKC's. One broad class of candidates is what we term *Tame-Like Multivariate Public-Key Cryptosystems*. *Tame-like PKC's involve a central map with relatively few terms in each equation and readily and quickly invertible, usually through no more than serial substitution and solving linear systems.* They are extremely fast and suited for deployment in resource-poor PKI environments, but *are tame-like PKC's secure enough?*

Early tame-like multivariate PKC's had included Birational Permutation Schemes ([39]) and TTM ([28]). Coppersmith *et al* put paid the former ([11, 12]). Goubin and Courtois announced cryptanalysis of TTM in particular and of all "TPM" (triangular-plus-minus) PKC's, a much broader genre of similar systems, in general ([22]). They also conveyed the impression that the concept of a faster signature systems than $C^\star$-based ones is beyond redemption. The techniques they used were not new ([4, 11, 12, 42]), but they somewhat expanded the scope and simplified the procedures. Little attention has been paid to tame-like PKC since then until Chen and Yang proposed the TTS (Tame Transformation Signatures) family of digital signatures ([9]). As usual, the truth lies somewhere in between.

We will discuss why tame-like PKC's are desirable, and how the attacks of Goubin-Courtois (which can be succintly summarized as *"rank attacks"*) function and how well they work in general. We point out liabilities in current TTS instances, in particular, *the non-obvious vulnerability of having central equations with many linear combinations at the same rank*. We show how to cryptanalyze them on these vulnerabilities. Then we show how to construct tame-like PKC so as to account for such possible weaknesses. In line with our suggestions, we exhibit patched TTS instances resistant to all known attacks, still lightning fast in comparison and at least quite suitable for smart card implementation.

The result of our suggested repair work seems promising, as seen by Table 1. Compared to RSA, it has good security[2] levels against known attacks, and it signs 3 orders of magnitude faster. We did basic simulations to make sure that no estimate is out of line. It is hoped that our results can somewhat spur some renewed interest of cryptographers on multivariate PKCs.

m	n	PubKey	SecKey	Rank	Dual Rank	XL	RSA bits	ECC bits
20	28	8680 B	1399 B	2^{98}	2^{80}	2^{80}	≥ 1024	144
24	32	13440 B	1864 B	2^{98}	2^{88}	2^{91}	≥ 1536	160
28	38	21812 B	2594 B	2^{130}	2^{105}	2^{103}	≥ 2560	192
32	44	33088 B	3444 B	2^{152}	2^{121}	2^{114}	≥ 4096	224
36	50	47700 B	4414 B	2^{184}	2^{138}	2^{130}	≥ 6144	256

Table 1: Minimal Security Estimates of Enhanced TTS instances, $(m, n) =$ hash and signature sizes

[1] Patarin *et al* recently announced that SFLASH^{v2} is not secure enough ([16]). SFLASH^{v3}, its intended replacement, is supposedly still faster than RSA but has much bigger dimensions, signatures and keys. QUARTZ, slow to begin with, also has its security called into question ([13, 19]).

[2] Security Estimates for RSA and ECC taken from NESSIE ([33])

2 Tame-Like Multivariate PKC's

In $C^{\star}$ (resp. HFE), the central map ϕ_2 is really taking one (resp. sum of a few) given high powers. As a result, in HFE, ϕ_2^{-1} is painfully slow; $C^{\star}$ has a simpler and much faster ϕ_2^{-1}, but vulnerabilities of the $C^{\star -}$ family originate from its structure ([35]). In either family each y_i when written as a quadratic polynomial in the x_j has hundreds of terms, we cannot invert ϕ_2 without treating all of $\mathbf{y}$ as an element in a larger field. To avert the consequential time penalty, we should consider treating each x_i or y_j as *separate entities, rather than components of a big field element.* But ϕ_2^{-1} must remain doable quickly for private map evaluation. The most obvious and minimal requirement is to be able to find each component of $\mathbf{x}$ in some *mostly sequential order* when given $\mathbf{y}$. This is an approach that has been tried in some earlier attempts ([20, 39]).

We inch closer to TTS or at least *tame-like PKC's*. Lest we forget, *in the central map ϕ_2 of* tame-like *PKC's, each y_i written as a quadratic polynomial in the x_i has relatively few terms — as few terms as security would permit, and an easy inverse available through fast, simple means.*

Advantages of tame-like PKCs are speed, ease of implementation, and avoiding old attacks.

Fast Signing: In SFLASH^{v2}, the signing action include multiplying and raising to the 128-th power in $(\mathrm{GF}(2))^{37}$ many times. A *tame-like* PKC makes this stage faster.

Fast Setup: In SFLASH^{v2}, the set-up process is a complex and round-about affair, involving evaluating ϕ_2 (itself a complicated procedure) almost a thousand times. In a tame-like PKC, with few terms per equations, we can do this by brute-force. This is no problem on a modern PC, but setting up on-card for the SFLASH^{v2} takes a *long* time.

Avoidance of Previous Liabilities: There are many possible designs for tame-like maps, this means we can sidestep weaknesses that SFLASH^{v2} must design around.

Drawbacks of tame-like PKCs are (mostly) possible new vulnerabilities on rank.

2.1 Tame Transformations, Tame(-Like) Maps, and TTS

One type of map stands out as a candidate for ϕ_2. In algebraic geometry, one kind of map is called a *Tame Transformation*, which with *base field* K and dimensions $m \geq n$ is a polynomial map[3] $\phi : K^n \to K^m$, taking $\mathbf{x}$ to $\mathbf{y}$ either affinely ($\mathbf{y} = \mathsf{M}\mathbf{x} + \mathbf{c}$) or in *de Jonquiere* form:

$$y_1 = x_1; \quad y_j = x_j + q_j(x_1, x_2, \ldots, x_{j-1}),\ j = 2 \cdots n; \quad y_j = q_j(x_1, x_2, \ldots, x_n),\ j = n+1 \cdots m.$$

If bijective it is also called *a tame automorphism.* Obviously we must then have $m = n$.

A tame transformation can be inverted quickly, but its inverse has high degree and is hard to write out explicitly. This is a venerable concept — in two variables, all polynomial automorphisms can be decomposed into compositions of tame automorphisms ([31]). It is unknown, despite the efforts of a lot of algebraic geometers, whether a map in three or more variables is a composition of tame automorphisms, and if so how to decompose it.

Moh harnessed this basic idea in his public-key encryption scheme TTM ([28]). Chen and Yang adapted the underlying concept of TTM for digital signatures ([8]), and slightly extended it ([9]) to include the larger class of polynomial maps that we can easily find an inverse for using a sequence of substitutions and *solving for linear equations*, but without a low degree explicit inverse. As in [9], we will hence term such maps *tame*. For example, the map below

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

[3] Note that in a finite field just about any function can be represented as a polynomial.

is a tame map, because a preimage can be componentwise computed, straightforwardly and quickly, after assigning any $x_1, \dots, x_7$ and $x_0 \neq -d_{27}^{-1}$. Tame maps are the centerpiece of TTS ([9]):

> *The TTS (Tame Transformation Signatures) family of digital signature schemes are defined as "a multivariate scheme with a tame map as its central, non-linear portion ϕ_2".*

The middle map ϕ_2 was sometimes also called the *kernel,* but it is too confusing here, and we will use the name the *central map* instead. A TTS scheme clearly fits the *tame-like* concept in Sec. 1, if each equation in its central map giving a y_i (we call this one of its *central equations*) has relatively few terms involving the x_j's compared to the dimensions n and m.

2.2 Current Variants of TTS

The public (verification) map of TTS has the canonical decomposition (in the notation of [9]) of most multivariate PKC's, namely $V : \mathbf{w} \in K^n \overset{\phi_1}{\mapsto} \mathbf{x} \overset{\phi_2}{\mapsto} \mathbf{y} \overset{\phi_3}{\mapsto} \mathbf{z} \in K^m$. We will henceforth take the base field K to be $\mathrm{GF}(2^8)$. Its current[4] form of TTS ([9]) is "TTS/4", using 20-byte hashes and 28-byte signatures. Its central map $\phi_2 : \mathbf{x} = (x_0, x_1, \dots, x_{27}) \mapsto \mathbf{y} = (y_8, y_9, \dots, y_{27})$ is:

$$
\begin{aligned}
y_k &= x_k + a_k\, x_{k-8}\, x_{k-1} + b_k\, x_{k-7}\, x_{k-4} + c_k\, x_{k-6}\, x_{k-2} + d_k\, x_{k-5}\, x_{k-3},\ 8 \le k \le 23;\\
y_{24} &= x_{24} + a_{24}\, x_{16}\, x_{23} + b_{24}\, x_{17}\, x_{20} + c_{24}\, x_{18}\, x_{22} + d_{24}\, x_{\mathbf{4}}\, x_{\mathbf{24}};\\
y_{25} &= x_{25} + a_{25}\, x_{17}\, x_{24} + b_{25}\, x_{18}\, x_{21} + c_{25}\, x_{\mathbf{4}}\, x_{23} + d_{25}\, x_{\mathbf{5}}\, x_{\mathbf{25}};\\
y_{26} &= x_{26} + a_{26}\, x_{18}\, x_{25} + b_{26}\, x_{\mathbf{4}}\, x_{22} + c_{26}\, x_{\mathbf{5}}\, x_{24} + d_{26}\, x_{\mathbf{6}}\, x_{\mathbf{26}};\\
y_{27} &= x_{27} + a_{27}\, x_{\mathbf{4}}\, x_{26} + b_{27}\, x_{\mathbf{5}}\, x_{23} + c_{27}\, x_{\mathbf{6}}\, x_{25} + d_{27}\, x_{\mathbf{7}}\, x_{\mathbf{27}}.
\end{aligned}
$$

We see that this ϕ_2 is also *tame* because from any $\mathbf{y}$ we quickly compute one possible $\mathbf{x}$ by randomly assigning a value to $x_0, \dots, x_7$ (subject to the restrictions $x_i \neq d_{20+i}^{-1}$ for $i = 4 \cdots 7$) and solving sequentially for $x_8, \dots, x_{27}$. An alternative form, called TTS/2′ uses as ϕ_2 the map given in Sec. 2.1. Both TTS instances operate over $K = \mathrm{GF}(2^8)$ as follows (see [9]):

To Setup Keys: Generate random full-rank square matrices M_1 (of dimension 28) and M_3 (of dimension 20) over K. Similarly, generate random non-zero $a_i, b_i, c_i, d_i \in K$ for $i = 8 \cdots 27$, and a random vector $\mathbf{c}_1 \in K^{28}$. Find the composition $V = \phi_3 \circ \phi_2 \circ \phi_1$ and in the process compute the unique $\mathbf{c}_3$ such that V has no constant part. Save the 8680 coefficient of V as the public key. Save M_1^{-1}, M_3^{-1}, $\mathbf{c}_1$, $\mathbf{c}_3$, and parameters a_i, b_i, c_i, d_i as the private key (total 1312 bytes).

To Sign: Take the message M, find its 160-bit hash digest vector $\mathbf{z} = H(M)$. Do $\mathbf{y} = \mathsf{M}_3^{-1}(\mathbf{z} - \mathbf{c}_3)$, then $\mathbf{x} \in \phi_2^{-1}(\mathbf{y})$ as above, then $\mathbf{w} = \mathsf{M}_1^{-1}(\mathbf{x} - \mathbf{c}_1)$. Release $(M, \mathbf{w})$.

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

TTS/4 and TTS/2′ claim very fast execution times, short signatures, manageable key lengths, and reasonable security. Previous analysis ([9]) seems to show known attacks to be ineffective. The best attacks previously came from the XL family ([15]). TTS family schemes are well placed to resist XL-type attacks because it can be structured to have high-dimensional solution spaces at infinity ([9, 29]). Even giving the XL-wielding attacker all benefits of the doubt, TTS/4 and TTS/2′ still have a security level of 2^{80} AES blocks (about 2^{88} finite field multiplication operations). The other powerful general attack, the method of Gröbner Bases, is hard to obtain a tight timing for. But the same properties that guards against XL-methods also helps against Gröbner Basis attacks.

We will show however that there are design misjudgment in these TTS instances that leads to fast cryptanalysis and how to patch them effectively and generically.

[4] Boldface indices are irregularities in the pattern of indices made in TTS/4 for security improvements ([9]).

3 Rank Attacks vs. TPM and Other Tame-Like PKC's

Of the many known rank attacks, the most ambitious was probably that of [22], wherein the authors postulated a type of PKC called TPM (triangular plus-minus), and claimed that their general attack renders TPM (and strongly by implication, all tame-like or non-C^*-descended, non-HFE-derived PKC's) totally useless. *What they claimed was that the private keys can almost always be distilled from the public key of a tame-like PKC by seeking linear combinations of certain matrices at given ranks.* A TPM system is essentially just a simple multivariate PKC with a tame transformation as ϕ_2, and with some equations removed at the beginning (like with TTS). To evaluate how tame-like PKC's stand up to such attacks, we need to answer many questions:

- Does the TPM category really cover all the tame-like PKCs of interest?
  NO! In particular, TTS does not match what the authors of [22] describe as a TPM signature scheme. T. Moh ([30]) also maintains that the description does not match TTM.

  The biggest mismatches: A tame-like PKC may solve linear equations rather than search, and need not have a sequence of increasing kernels in the central equations.

- If not, can the attack be extended to other tame-like systems?
  Yes, while some objections of [30] seem valid, the ideas are meritorious and can be applied to decompose public maps from many PKC's, including badly designed TTS or TTM instances.

- Does the attack always work as described? Can it be faster or slower, and when?
  Yes, sort of. The attack should always conclude successfully if we can find kernels corresponding to all central equations. If the central equations are tangled somehow and finding the kernel to one central equation makes finding another one easier, we may cryptanalyze much faster. Otherwise the search can go on for a lot longer.

- Can we construct systems of a requisite complexity under Rank and other attacks?
  Yes, we can arrange for any desired complexity against rank attacks while retaining high practical speed and resilience against other attacks; that is the subject matter of Sec. 4.

3.1 The Rank Attack: Vulnerability on the Low-Rank Side

Let the public map take K^n to K^m, i.e. m and n be the number of equations and variables respectively. Also let $q = |K|$, and r be the smallest rank in linear combinations of central equations. It was claimed ([22]) that a tame-like PKC is broken in expected time $O(q^{\lceil \frac{m}{n} \rceil r} m^3)$. The steps are as outlined in [22]:

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. Here is a peculiarity when char $K = 2$ that [22] did not account for. *When the quadratic portion of z_i is written as $\mathbf{w}^T Q_i \mathbf{w}$, the matrices Q_i cannot be symmetric, and it can still be written in many ways.* However all is not lost, as there is still a unique symmetric matrix that can be said to represent z_i, namely $H_i = Q_i + Q_i^T$. We borrow an illustration from [9], showing the rank of $y_8 = x_8 + a_8\, x_0\, x_7 + b_8\, x_1\, x_6 + c_8\, x_2\, x_5 + d_8\, x_3\, x_4$, from TTS/2′ (Sec. 2.2):

$$\left[\begin{array}{cccccccc|c}
0 & 0 & 0 & 0 & 0 & 0 & 0 & a_8 & \\
0 & 0 & 0 & 0 & 0 & 0 & b_8 & 0 & \\
0 & 0 & 0 & 0 & 0 & c_8 & 0 & 0 & \\
0 & 0 & 0 & 0 & d_8 & 0 & 0 & 0 & \\
0 & 0 & 0 & d_8 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & c_8 & 0 & 0 & 0 & 0 & 0 & \\
0 & b_8 & 0 & 0 & 0 & 0 & 0 & 0 & \\
a_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\
\hline
\multicolumn{8}{c|}{0} & 0
\end{array}\right]$$

No matter how we write this quadratic part of y_8 as $(\mathbf{x}^T Q \mathbf{x})$, $(Q + Q^T)$ will be as shown to the right, and its kernel is $x_0 = x_1 = \cdots = x_7 = 0$. Indeed, if a quadratic has the form $C_{ab} x_a x_b + C_{cd} x_c x_d + \cdots$ with all the indices $a,\ b,\ c,\ d, \ldots$ distinct from each other, $\{\mathbf{x} :\ 0 = x_a = x_b = x_c = x_d = \cdots\}$ will be the kernel of the corresponding symmetric matrix, hence for TTS/2′ or TTS/4, $\{\mathbf{x} :\ x_{k-8} = \cdots = x_{k-1} = 0\}$ can be said to be the kernel of y_k in $\mathbf{x}$-space. For ease of reference we will use the shorthand $\ker y_i$, or $\ker_{\mathbf{x}} y_i$ if there may be confusion.

We see that the rank of (the symmetric matrix H_8 corresponding to) y_8 in $\mathbf{x}$-space is 8. This rank is unchanged in $\mathbf{w}$-space, because if we write the quadratic part of y_i as both $\mathbf{x}^T \hat{Q} \mathbf{x}$ and $\mathbf{w}^T Q \mathbf{w}$, then clearly $Q = (\mathsf{M}_1)^T \hat{Q}\, (\mathsf{M}_1)$, and $(Q + Q^T) = (\mathsf{M}_1)^T (\hat{Q} + \hat{Q}^T)(\mathsf{M}_1)$. Indeed, if the kernel of y_8 in $\mathbf{x}$-space is S then the kernel in $\mathbf{w}$-space is $(\mathsf{M}_1)^{-1} S$.

We note here that for ℓ cross-terms with distinct indices, the rank of the matrix is 2ℓ.

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 attempt to solve for λ_i via Gaussian elimination. The equations will be almost uniquely solvable when P is 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).

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

Proof. For each k-tuple $(\mathbf{w}_1, \ldots, \mathbf{w}_k)$ and each pair (i, j) we must evaluate $H_i \mathbf{w}_j$ (n^2 multiplications each) and then do Gaussian elimination on nk equations and m variables. The requisite number of multiplications can be found in numerical analysis texts (e.g. [3]). □

According to [22] the kernels corresponding to each y_i form an increasing chain by containment, so once the largest kernel has been found, the scheme should unravel in its entirety. After that one could find M_3, and then M_1 by searching in each kernel space for the next smaller kernel. We note that square terms in the central map are eliminated during symmetrization[5] and does not affect a rank attack. One expects a rank attack to do its worst against a signature scheme, since $k = 1$. *However, the TPM schemes being attacked is not the actual schemes.* Hence, we need to evaluate how well they actually apply to the point where we can make a real decomposition or forgery. We will try to compute the actual effort in attacking a TTS instance on rank in Sec. 3.3.

3.2 Other Concerns in a Rank Attack

Clearly attacking on low rank is devastating when the conditions are met. But it is no panacea and needs some corrections and proper care in implementation. In particular, these can all go wrong:

1. In [22], the target scheme has $r = 2$. It can be a lot higher. For example, $r = 8$ in TTS/4 and TTS/2′; furthermore, we can increase this parameter with relative ease. According to [28], the

[5] In a sense, square terms are fundamentally linear.

dimensions of a TTM instance can be such that $k = \lceil \frac{n}{m} \rceil = 3$. Suppose every central equation has at least two cross-terms, then $r = 4$ and we are talking about $q^{kr} = 2^{96}$ already.

2. Normally, in a PKC everything except the secret key is known. But when trying to break a multivariate PKC, the attacker may not know in advance what scheme a public key represents, only the base field K, the dimensions (n, m), and a set of public-map polynomials. E.g. a TTS central map can spawn (in addition to parameters) adjustible indices or even optional terms.

3. In a TPM scheme of [22], the kernels of the central equations form a decreasing sequence: $\ker y_{i+1} \subset \ker y_i$. In a well-designed scheme, the kernels of the central equations may not form such a sequence, and there may be no *domino effect*. If an attacker need to find every y_i then a lot more effort is necessary (see below). This is intimately connected to the next point.

4. While we assume that y_1 has 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.

   This is a very double-edged sword. In TTS/2$'$, for non-zero α, the rank 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. If we can not make use of the relationship between the combinations, just keeping track of everything is a major chore; if we can, then the cryptanalysis may become substantially easier. It is on this point that we will show how TTS/2$'$ and TTS/4 can be cryptanalyzed by rank attacks with a lower security than previously known.

3.3 The "Crawling" Rank Attack vs. TTS/4

Take any given rank 8 central equation, then when $n = 28$ and $m = 20$, according to Prop. 1, we should need $256^8 \cdot \left[20^2 \cdot (28/2 - 20/6) + 20 \cdot 28^2\right] \approx 2^{78}$ field multiplications. NESSIE ([32]) requirements are not counted in field multiplications however, but in AES blocks. Using data from the NESSIE performance report ([34]), and comparing with actual operations, we obtain the exchange rate of one AES block to $\approx 2^8$ multiplications when using table lookups for multiplication. This goes down to $\approx 2^7$ (a factor of two) when using tables of logarithms and exponentials. All told, we can expect a time complexity of $\approx 2^{71}$ if we want to find a vector in any given rank-8 kernel. However, *there are many kernels to choose from, and any single one works*, as follows.

For simplicity in illustration, let $a_8 = a_9 = a_{10} = b_8 = \cdots = d_{10} = 1$, then we have

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

If the 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. These kernels show the way to cryptanalysis. We proceed along these steps:

1. Run the algorithm of Sec. 3.1 to find a vector $\mathbf{u}$, the associated quadratic $z = \sum_i \lambda_i z_i$ of rank 8. Then verify $U = \ker z$ to be of codimension 8, and find a basis for U. The expected number of multiplications needed is roughly 2^{78} divided by the number of rank-8 forms, or $\sim 2^{65}$.

   We note that kernels of these 10,000+ rank-8 forms are largely distinct. Since there are only 20 rank-8 forms y_i, but about 5000 rank-8 forms $y_i + \alpha y_{i+1}$ and almost as many forms $y_i + \alpha y_{i+2}$, so it is with good probability that the first vector yielding a codimension-8 kernel will come from a mixed form rather than from one of the y_i's, and we need to isolate y_i's thence.

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 $\mathrm{span}(y_i, y_{i\pm1})$.

   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 $\ker y_8 \cap \ker y_9 = \{\mathbf{x} : x_0 = x_1 = \cdots = x_8 = 0\}$.

   The same applies 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 find a vector $\mathbf{v}$ in more than one kernel is 2^{-16}, and what we find is $(\ker y_i) \cap (\ker y_{i+1}) \cap (\ker y_{i+2})$.

   The expected number of field multiplications needed for this step is very small, equivalent to trying 2^{16} random vectors $\mathbf{w}$ in Sec. 3.1, or about 2^{30} multiplications here.

3. Of all the linear combinations of quadratic forms f_i that are not multiples of each other and 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 until we find the kernels that corresponds to the y_i's. *Suppose we check 2^{12} vectors from each of the $\sim 2^{16}$ kernels U_i to see if any of them is a y_i, that would take no more than 2^{42} multiplications.*

4. Say we have found the form for 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 eight others as $x_1, \ldots, x_8$, so in short, finding any y_i should yield in very short order all y_j and x_j where $j < i$. Even if we can't do the decomposition, the same incremental search going up and down the indices will locate all the forms y_i and x_i, i.e. the matrices M_1 and M_3, for us.

With the above *crawl* process aiding our attack, the chance of finding a kernel vector is essentially multiplied by about 2^{14} as compared to the attack in [22]. The upshot is that a solution can be located in between 2^{64} to 2^{65} multiplications (or 2^{56} to 2^{57} AES blocks).

We experimented with 2- and 3-term analogues to TTS/2′ schemes. We were unable to complete the whole run with 2^{48} field multiplications for a three-term ($r = 6$, $n = 22$, $m = 16$) TTS/2′ analogue, but the incremental search technique works, and on (Pentium and Athlon) PC's what the above description of the cryptanalysis predicted was in reasonable accord with what happened during our testing. With 2-term TTS/2′ type sample scheme ($r = 4$, $n = 16$, $r = 12$) identifying the initial vector actually takes less time ($\sim 2^{32}$ multiplies) than the search for each new y_i ($\sim 2^{40}$ multiplies).

3.4 The Dual Rank Attack: Vulnerability on the High-Rank Side

It seems natural that the converse of the Rank Attack — finding a large kernel shared by a small subset of the space spanned by the matrices H_i — is to find a small kernel shared by a large subset of the linear combinations of the H_i. In TTS/2 (the original TTS, [8]), the variable x_{27} does not appear in any cross-term, and therefore, $\cap_{i=8}^{27} \ker_{\mathbf{w}} z_i = \{x_0 = x_1 = \cdots = x_{26} = 0\}$. In Birational Permutation Schemes, the last central variables x_n shows in cross-terms of only one equation. This critical weakness ([12]) makes it easy to find linear combinations $\sum_i \alpha_i z_i$ whose kernels share a non-empty intersection.

Coppersmith *et al* ([11, 12]) showed an intricate way to find an ascending chain of kernels in the matrix algebra over a ring, *using algebraic method and without needing to search*. That neatly broke

Birational Permutations. In [22] the same Dual Rank Attack[6] was carried out by searching, while erroneously comparing the CSV idea to those in Sec. 3.1. Of the equivalent formulations, we can distill the essence of the simpler version of the Dual Rank Attack in [22] as follows:

Without loss of generality, let the last variable x_{n-1} appear u times in the cross-terms of the central equations, which is the fewest appearances of all variables. In TTS/4, 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. 3.1.) If we denote by m_{ij} the (i, j)-entry of M_3, then almost every pair of H_i and H_j will have a linear combination with a kernel that 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 the common subspace $U = \{\mathbf{x} : \ x_0 = \cdots = x_{n-2} = 0\}$. We try to find U.

1. Form an arbitrary linear combination $H = \sum_i \alpha_i H_i$. Find $V = \ker H$ by Gaussian elimination.
2. Because a matrix in $K^{n \times n}$ can have at most n different eigenvalues, less than n/q of the time we would have $\dim V = 1$. Now set $(\sum_j \lambda_j H_j) V = \{\mathbf{0}\}$ and check if the solution set $\hat{V}$ of the (λ_i), also found via a Gaussian Elimination, form a subspace dimension $m - u$.
3. With probability q^{-u} we have $V = U$. The cost of one trial is bound 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 cut down to a little more than $\left(un^2 + \frac{n^3}{6} \right) q^u$ (in field multiplications) if we only consider linear combinations of $(u+1)$ of the matrices H_i, and don't get too unlucky.

From this subspace, we can expand to find bigger kernels. In [12] this was through taking a sequence of derivatives. It is easier for the case of TPM, as it is for TTS/4 and TTS/2'. The next bigger kernel up the chain, which is $U' = \{x_0 = x_1 = \cdots = x_{25} = 0\}$, can be found by looking at subspaces of V, which will get us U' with probability $1/q$. So for TTS/4 and TTS/2$'$, the cryptanalysis is instant.

We can rephrase the above as to *look at a linear combination of the (duals of) w_i that has low rank when expressed as a linear mapping from the w_j to z_k*, so the name "Dual Rank Attack" seems to fit.

3.5 Further Discussion about Rank Attacks

Cryptanalysis of TTS/4 proceeds identically as TTS/2$'$, except that there seems to be no three-term combinations of rank 8. We can use the *crawl* to fish out successive y_i once one combination with a rank-8 kernel has been found. The complexity should obviously be comparable to that of TTS/2$'$.

We can cryptanalyze improperly constructed instances of TTM very easily. Quite a few variants of TTM had been proposed by T. Moh *et al.* Some of them have central equations of the form $y_j = x_i + A_j x_h x_\ell$. That is an equation of rank 2. The presentation in [22] does not make it very clear, but the attack does not necessarily have to work on the initial equations. If there is no other central equation of rank 2 with either x_h or x_ℓ in a cross-term, the kernel attack will easily locate y_j, x_h, x_ℓ and x_i after an expected 256^{2k} attempts at guessing some kernel vectors, where $k = \lceil \frac{m}{n} \rceil$ is 2 or 3, that's about 2^{58} multiplications max. Suppose we have many equations of rank 2, whose sole cross-terms are $x_i x_{j_1}, x_i x_{j_2}, \ldots, x_i x_{j_s}$. By the same arguments as in Sec. 3.3, we will locate a kernel vector of a quadratic form that looks like $x_i(\alpha_1 x_{j_1} + \alpha_2 x_{j_2} + \cdots + \alpha_s x_{j_s})$ after 256^{2k-s+1} attempts. Even with two cross-terms in each equation, if there are s equations of which any linear combination will still be rank 4, the cost is only $2^{8(4k-s+1)}$ attempts (each is some substitutions plus a Gaussian Elimination).

There is a moral we can distill from this episode. People noticed the impact of rank in multivariate cryptography early on. For example, Theobald was impressed enough to issue a warning ([43]) *"varying ranks of quadratic forms"* comprising the non-linear portion of a multivariate PKC is dangerous. However, with great trepidation we venture this humble opinion:

[6] Suggested by someone asked to review an earlier version. It seems more suitable than High or Max Rank Attack.

> The expert cryptographers were warning against *varying ranks*, however, the dangers that they saw may really have been *chains of kernels ordered by containment*, and in particular, *such a chain of kernels with some vulnerability at either end.*

Note that we said either end. When you have a long chain of kernels, the smallest as well as the largest can be the weakness, like we expanded on, as above.

4 Tame-Like Signatures Free from Rank Concerns

What kind of Tame-Like Signature Schemes can we build that are secure to Rank Attacks? Clearly, being non-TPM is not sufficient, since no TTS instance discussed so far is a TPM. Neither is TTM.

The conclusion we can draw from Sec. 3 is: To be safe, the minimum rank r in the matrices representing the central equations and their linear combinations must be high; so must the minimum non-reducible number of equations u where any given variables shows in cross-terms. What else?

4.1 Criteria for a Safe, Fast Multivariate Signature Scheme

Aside from Rank Attacks, the main concern for a tame-like multivariate PKC must surely come from the powerful method of Gröbner Bases and its distant cousin, the linearization or XL based methods.

Proposition 2 *In a Tame-Like Digital Signature Scheme needing a complexity estimate of C:*

1. *Each central equation should contain as many cross-terms with no repeated indices as possible.*
2. *Almost all linear combinations of central equations should result in quadratic forms of higher rank, a small number (comparable to m, the number of equations) can have equal rank.*

   If k linear combinations of central equations have the same minimal rank $r = 2\ell$, then we need

   $$q^r \cdot \left(m^2(n/2 - m/6) + mn^2\right)/k \geq C. \tag{1}$$

3. *If the minimum number of appearances is u in central equations for any variable x_i, then*

   $$q^u \left(un^2 + n^3/6\right) \geq C. \tag{2}$$

   These sum up what we were doing in the last section.

4. *We want a set A of h indices $0 \leq i < n$ such that every cross-terms in the central map has at least one index in A. For now we prefer $h > n/2$ ([23, 24]). We need $h < m$ and lower h means higher "dimension of solution at infinity" and higher XL/FXL complexity (see Appendix B).*

Item 4 results from XL/FXL ([1, 13, 14, 15]) and Gröbner-based attacks ([2, 18, 19]). One should refer to [29] for some algebraic geometry on XL-type attacks, and to [5, 6, 26] on Gröbner Bases theory. A brief synopsis of using Gröbner Bases against tame-like PKC conforming to item 4 above is: *you shouldn't be able to.* An equally brief synopsis of using XL/FXL methods against conformant signature schemes is that one likely needs to guess at $\hbar = m - h - 1$ variables above, which leads to a large factor of q^{m-h-1} in the time cost. This is related to the $\dim H_\infty$ parameter of the central equations. For now see Table 1 for estimated security levels for conformant tame-like schemes under XL. An explanation of these estimates can be found in Appendix B.

4.2 Tame-Like Digital Signature Schemes Built To Rank Specifications

NESSIE requires a complexity of 2^{80} AES blocks, or about 2^{86} multiplications. Because of birthday attacks, the hash length needs to be 160-bit, or $m \geq 20$. Obviously $n > m$ in a signature scheme like TTS. We need $r \leq 10$, so there should be at least 5, probably 6 or 7 cross-terms in each equation. But we don't want n too large, because (a) $n - m$ too large can lead to searching-like concerns (see [9]); (b) makes securing them against XL attacks harder (see Appendix B); and (c) obviously means longer keys and times (all $\propto n^3$). In all, we want (n, m) no bigger than $(28, 20)$ or perhaps $(32, 24)$.

Is this possible? Yes, by adopting a segmented design. The initial x_i's ($x_0, \ldots, x_7$) are essentially random (see below). The initial equations (starting with y_8) are solved as a linear system for x_8 and subsequent x_i's, with six plus cross-terms each; then the some "tame" equations yield more x_i's through only serial substitution; then the last block of equations is solved as a linear system for the final x_i's (at least nine, which is also the minimum number of cross-terms in this block). For ease of programming, the two systems to solve should have the same number of equations.

What is the security assessment by Rank Attack? Each equation has rank 12 or more. Even if linear combinations of two consecutive equations in the first segment all have the same rank 12, we have a comfortable cushion since $2^{8\times 12} = 2^{96}$. If the last block has 9 equations, the Dual Rank Attack takes $256^9 \cdot (9 \cdot 28^2 + 28^3/3)$ or around 2^{86} multiplications $\approx 2^{80}$ AES blocks.

Can we ensure a signature for any hash? Yes! *Do not use x_0 until the final segment of equations. Make up the first segment with non-zero constant multiples of x_1 on the main diagonal of the system matrix, no other appearances for x_1. Then set up the final segment so that it has constant multiples of x_0 as the main diagonal of its system matrix and no other appearances of x_0*. This will do.

We exhibit an illustrative TTS instance with central map ϕ_2, with two blocks of nine equations each (and 7 and 10 terms per equation respectively) sandwiching two tame equations.

$$\begin{aligned}
y_i &= x_i + \textstyle\sum_{j=1}^{7} p_{ij} x_j x_{8+(i+j \bmod 9)},\ i = 8 \cdots 16;\\
y_{17} &= x_{17} + p_{17,1}x_1x_6 + p_{17,2}x_2x_5 + p_{17,3}x_3x_4\\
&\qquad + p_{17,4}x_9x_{16} + p_{17,5}x_{10}x_{15} + p_{17,6}x_{11}x_{14} + p_{17,7}x_{12}x_{13};\\
y_{18} &= x_{18} + p_{18,1}x_2x_7 + p_{18,2}x_3x_6 + p_{18,3}x_4x_5\\
&\qquad + p_{18,4}x_{10}x_{17} + p_{18,5}x_{11}x_{16} + p_{18,6}x_{12}x_{15} + p_{18,7}x_{13}x_{14};\\
y_i &= x_i + p_{i,0}x_{i-11}x_{i-9} + \textstyle\sum_{j=19}^{i} p_{i,j-18}\ x_{2(i-j)}\ x_j\\
&\qquad + \textstyle\sum_{j=i+1}^{27} p_{i,j-18}\ x_{i-j+19}\ x_j,\ i = 19 \cdots 27.
\end{aligned}$$

To see ϕ_2 more clearly and that it meets our requirements, we tabulate it differently in Table 2.

y	8	9	10	11	12	13	14	15	16	y	19	20	21	22	23	24	25	26	27	cross
8	1	2	3	4	5	6	7			19	0	18	17	16	15	14	13	12	11	$8, 10$
9		1	2	3	4	5	6	7		20	2	0	18	17	16	15	14	13	12	$9, 11$
10			1	2	3	4	5	6	7	21	4	2	0	18	17	16	15	14	13	$10, 12$
11	7			1	2	3	4	5	6	22	6	4	2	0	18	17	16	15	14	$11, 13$
12	6	7			1	2	3	4	5	23	8	6	4	2	0	18	17	16	15	$12, 14$
13	5	6	7			1	2	3	4	24	10	8	6	4	2	0	18	17	16	$13, 15$
14	4	5	6	7			1	2	3	25	12	10	8	6	4	2	0	18	17	$14, 16$
15	3	4	5	6	7			1	2	26	14	12	10	8	6	4	2	0	18	$15, 17$
16	2	3	4	5	6	7			1	27	16	14	12	10	8	6	4	2	0	$16, 18$

Table 2: Table Form of a Possible Central Map of an Enhanced TTS

This is how to invert ϕ_2:

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. At the very least the determinant of the first system (for any choice of 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 of nine equations 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.

Otherwise this signature scheme is identical to that of TTS/4 and TTS/2′. We can call this *Enhanced TTS*. The public key is still 8680 bytes, and the private key 1399 bytes (with 167 variable non-zero parameters, 1184 parameters in the matrices, and 48 bytes in the vectors). Here in this ϕ_2 we have $h = 15$ as in Item 4 above (all cross-terms vanish if $x_0 = x_2 = x_4 = x_6 = 0 = x_i,\ i = 8 \cdots 18$), and the "dimension of solution set at infinity" ($\dim H_\infty$) parameter is $\hbar = 4$ after the attacker guesses at 8 variables. In case that our FXL estimate is somewhat off, we can up this to $m = 24$, $n = 32$, with the same Rank Attack estimates, see Table 4.

4.3 Can Our Patched TTS Instances Measure Up and Scale Up?

The answers seems to be: yes and yes, in speed and scalability!

Speed: The central portion ϕ_2 as we described it does about 800 field multiplications (instead of < 200 as in TTS/4). ϕ_1 and ϕ_3 does about 400 and 784 respectively. Taking into account that equation solving is harder with lots of loops, the speed should be at least half of the superceded TTS/4 as listed in [9]. Indeed, preliminary testing shows that this is in fact the case. So we have a signature scheme that still signs 1000 times faster than RSA and two degrees of magnitude faster than any other method. On a smart card, we can likely make do with lower-rated hardware and without crypto co-processors, and still work faster than with RSA or ECC, and still have keys generated on-card, unlike that of SFLASH^{v3}.

Scalability: Let the requisite complexity be $C \gtrsim 2^{16k}$. With rank attacks we want $c_0 q^r$ and $c_1 q^u$ to be $\geq C$, with both constants around 2^{14} or 2^{15} (in multiplicatons) or 2^6 (in AES blocks. So a rough requirement is $u \geq 2k - 1$ and $r \geq 2k$, i.e. probably $k + 1$ cross-terms. To maintain our XL/FXL security (see Appendix B), we needs about $m = 4k$, so we can do like the very minimal format we described above for Table 2, using two system of linear equations with $2k - 1$ variables, one with $2k$ terms per equation, one with $k + 1$. There are two middle equations also with $k + 1$ terms. So $n \gtrsim 5k + 2$, maybe a multiple of 4 marginally larger for ease in programming.

Some sample ϕ_2's for TTS meeting our spec above are given in Appendix A.

We can estimate ϕ_2 does $\approx 6k^2(k + 2)$ multiplications. 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. But C^*-derived schemes will take time cubic in k also, and the coefficient in tame-like schemes is much smaller.

5 Conclusion

We describe herein how to construct a tame-like signature scheme less susceptible to attack on rank. The results look quite promising and we think that it bears another look by cryptographers, notwithstanding the apparent slowdown in the research of multivariate PKC of recent.

References

[1] Anonymous communication, *All in the XL Family: Theory and Practice*, preprint.

[2] M. Bardet, J.-C. Faugére, and B. Salvy, Complexity of Gröbner Basis Computations for Regular Overdetermined Systems, Preprint and private communication.

[3] R. Burden and J. D. Faires, *Numerical Analysis, 7th ed.*, PWS-Kent Publ. Co., 2000.

[4] J. Buss, G. Frandsen, J. Shallit, *The Computational Complexity of Some Problems of Linear Algebra*, report RS-96-33 published by BRICS, Aarhus, Denmark. Available at `http://www.brics.dk/RS/96/33`.

[5] L. Caniglia, A. Galligo, and J. Heintz, *Some New Effectivity Bounds in Computational Geometry*, AAECC-6, 1988, LNCS V. 357, pp. 131–151.

[6] L. Caniglia, A. Galligo, and J. Heintz, *Equations for the Projective Closure and Effective Nullstellensatz*, Discrete Applied Mathematics, 33 (1991), pp. 11-23.

[7] S. Cavallar *et al*, *Factorization of a 512-bit RSA modulus*. EUROCRYPT 2000, LNCS V. 1807, pp. 1-17.

[8] J.-M. Chen and B.-Y. Yang, *Tame Transformation Signatures with Topsy-Turvy Hashes*, proc. IWAP '02, Taipei, Taiwan.

[9] J.-M. Chen and B.-Y. Yang, *A More Secure and Efficacious TTS Signature Scheme*, accepted for publication by ICISC '03, expanded preprint available at `http://eprint.iacr.org/2003/160`.

[10] D. Coppersmith, private communication.

[11] D. Coppersmith, J. Stern, and S. Vaudenay, *Attacks on the Birational Permutation Signature Schemes*, CRYPTO'93, LNCS V. 773, pp. 435–443.

[12] D. Coppersmith, J. Stern, and S. Vaudenay, *The Security of the Birational Permutation Signature Schemes*, Journal of Cryptology, 10(3), 1997, pp. 207–221.

[13] N. Courtois, *Generic Attacks and the Security of Quartz*, PKC 2003, LNCS V. 2567, pp. 351–364.

[14] N. Courtois, *Algebraic Attacks over* $\mathrm{GF}(2^k)$, *Cryptanalysis of HFE Challenge 2 and SFLASH*^{v2}, accepted for PKC 2004.

[15] N. Courtois, A. Klimov, J. Patarin, and A. Shamir, *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations*, EUROCRYPT 2000, LNCS V. 1807, pp. 392–407.

[16] N. Courtois, L. Goubin, and J. Patarin, *SFLASH*^{v3}*, a Fast Asymmetric Signature Scheme*, preprint available at `http://eprint.iacr.org/2003/211`.

[17] W. Diffie and M. Hellman, *New Directions in Cryptography*, IEEE Trans. Info. Theory, vol. IT-22, no. 6, pp. 644-654.

[18] J.-C. Faugére, *A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5)*, Proceedings of ISSAC, ACM Press, 2002.

[19] J.-C. Faugére and A. Joux, *Algebraic Cryptanalysis of Hidden Field Equations (HFE) Cryptosystems Using Gröbner Bases*, CRYPTO 2003, LNCS V. 2729, pp. 44-60.

[20] H. Fell and W. Diffie, *Analysis of a Public Key Approach Based on Polynomial Substitution*, CRYPTO'85, LNCS V. 218, pp. 340–349.

[21] M. Garey and D. Johnson, *Computers and Intractability, A Guide to the Theory of NP-completeness*, 1979, p. 251.

[22] L. Goubin and N. Courtois, *Cryptanalysis of the TTM Cryptosystem*, ASIACRYPT 2000, LNCS V. 1976, pp. 44–57.

[23] A. Kipnis, J. Patarin, and L. Goubin, *Unbalanced Oil and Vinegar Signature Schemes*, CRYPTO'99, LNCS V. 1592, pp. 206–222.

[24] A. Kipnis and A. Shamir, *Cryptanalysis of the Oil and Vinegar Signature Scheme*, CRYPTO'98, LNCS V. 1462, pp. 257–266.

[25] A. Kipnis and A. Shamir, *Cryptanalysis of the HFE Public Key Cryptosystem*, CRYPTO'99, LNCS V. 1666, pp. 19–30.

[26] D. Lazard, *Gröbner Bases, Gaussian Elimination and Resolution of Systems of Algebraic Equations*, EUROCAL '83, LNCS V. 162, pp. 146–156.

[27] T. Matsumoto and H. Imai, *Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption*, EUROCRYPT'88, LNCS V. 330, pp. 419–453.

[28] T. Moh, *A Public Key System with Signature and Master Key Functions*, Communications in Algebra, 27 (1999), pp. 2207–2222.

[29] T. Moh, *On The Method of XL and Its Inefficiency Against TTM*, available at `http://eprint.iacr.org/2001/047`

[30] T. Moh and J.-M. Chen, *On the Goubin-Courtois Attack on TTM*, available at `http://eprint.iacr.org/2001/072`

[31] M. Nagata, *On Automorphism Group of* $K[X, Y]$, Lectures in Mathematics, vol. 5, Kinokuniya, Tokyo, Japan, 1972.

[32] New European Schemes for Signatures, Integrity, and Encryption, project homepage at `www.cryptonessie.org`

[33] *NESSIE Security Report, V2.0*, available at `http://www.cryptonessie.org`

[34] *Performance of Optimized Implementations of the NESSIE Primitives, V2.0*, available at `http://www.cryptonessie.org`

[35] J. Patarin, *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88*, CRYPTO'95, LNCS V. 963, pp. 248–261.

[36] J. Patarin, *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms*, EUROCRYPT'96, LNCS V. 1070, pp. 33–48.

[37] J. Patarin, N. Courtois, and L. Goubin, *QUARTZ, 128-Bit Long Digital Signatures*, CT-RSA 2001, LNCS V. 2020, pp. 282–297. Updated version available at `http://www.cryptonessie.org`

[38] J. Patarin, N. Courtois, and L. Goubin, *FLASH, a Fast Multivariate Signature Algorithm*, CT-RSA 2001, LNCS V. 2020, pp. 298–307. Updated version available at `http://www.cryptonessie.org`

[39] A. Shamir, *Efficient Signature Schemes Based on Birational Permutations*, CRYPTO'93, LNCS V. 773, pp. 1–12.

[40] A. Shamir, private communication.

[41] A. Shamir and E. Tromer, *Factoring Large Numbers with the TWIRL Device*, CRYPTO 2003, LNCS V. 2729, pp. 1-26.

[42] J. Stern, F. Chabaud, *The Cryptographic Security of the Syndrome Decoding Problem for Rank Distance Codes*, ASIACRYPT'96, LNCS V. 1163, pp. 368–381.

[43] T. Theobald, *How to Break Shamir's Asymmetric Basis*, CRYPTO'95, LNCS V. 963, pp. 136–147.

[44] L. Wang, *Tractable Rational Map Cryptosystem*, private communications and colloquium presentation.

A Examples of Enhanced TTS, Scaled-Up

We were unable to find ϕ_2 with $n = 28$, $m = 20$ in two systems of 10 equations that can be easily constructed with regular patterns in its indices, unless we accept repetitive cross-terms (there are no repeats now). However, more irregular instances exist, and here is one example:

y	7	6	5	4	3	2	1	cross	y	18	19	20	21	22	23	24	25	26	27	cross
8	8	9	10	11	12			$1,2$	18	0	4	5	15	14	13	12	11	10	9	$7,8$
9	9	10	11	12			13	$2,3$	19	10	0	3	4	16	15	14	13	12	11	$8,9$
10	10	11	12			13	14	$3,4$	20	12	11	0	2	3	17	16	15	14	13	$9,10$
11	11	12			13	14	15	$4,5$	21	14	13	12	0	5	2	8	17	16	15	$10,11$
12	12			13	14	15	16	$5,6$	22	16	15	14	13	0	4	5	9	8	17	$11,12$
13	13		14	15	16		17	$6,2$	23	9	8	17	16	15	0	3	4	11	10	$12,13$
14	14	15	16	17		8		$1,3$	24	11	10	9	8	17	16	0	2	3	12	$13,14$
15	15	16	17		8		9	$2,4$	25	13	12	11	10	9	8	17	0	5	2	$14,15$
16	16	17		8		9	10	$3,5$	26	2	14	13	12	11	10	9	8	0	4	$15,16$
17	17		8		9	10	11	$4,6$	27	3	5	15	14	13	12	11	10	9	0	$16,17$

Table 3: A Different Central Map for Enhanced TTS

Each row specifies a central equation, for the initial equation of the two blocks are:

$$\begin{aligned} y_8 &= x_8 + a_8 x_7 x_8 + b_8 x_6 x_9 + c_8 x_5 x_{10} + d_8 x_4 x_{11} + e_8 x_3 x_{12} + f_8 x_1 x_2 \\ y_{18} &= x_{18} + a_{18} x_{18} x_0 + b_{18} x_{19} x_4 + c_{18} x_{20} x_5 + d_{18} x_{21} x_{15} + e_{18} x_{22} x_{14} + \\ & \quad f_{18} x_{23} x_{13} + g_{18} x_{24} x_{12} + h_{18} x_{25} x_{11} + i_{18} x_{26} x_{10} + j_{18} x_{27} x_9 + k_{18} x_7 x_8 \end{aligned}$$

We estimate this to have a security estimate of about 2^{88} under Rank and Dual Rank attacks, but still the same XL complexity of around 2^{80}. It is our goal to show that the construction is adaptable.

$$\begin{aligned} y_i &= x_i + \textstyle\sum_{j=1}^{7} p_{ij} x_j x_{8+(i+j+1 \bmod 10)},\ i = 8 \cdots 17; \\ y_i &= x_i + p_{i1} x_{i-17} x_{i-14} + p_{i2} x_{i-16} x_{i-15} + p_{i3} x_{i-10} x_{i-1} + p_{i4} x_{i-9} x_{i-2} \\ & \qquad + p_{i5} x_{i-8} x_{i-3} + p_{i6} x_{i-7} x_{i-4} + p_{i7} x_{i-6} x_{i-5},\ i = 18 \cdots 21; \\ y_i &= x_i + p_{i,0} x_{i-10} x_{i-14} + \textstyle\sum_{j=22}^{i} p_{i,j-21}\ x_{2(i-j)}\ x_j \\ & \qquad + \textstyle\sum_{j=i+1}^{31} p_{i,j-21}\ x_{i-j+21}\ x_j,\ i = 22 \cdots 31. \end{aligned}$$

Just in case that our XL/FXL estimate is slightly off, we can scale up to the larger ϕ_2 above, with $(n,m) = (32,24)$, which should have an FXL complexity about 2^{16} times higher (table form below).

y	8	9	10	11	12	13	14	15	16	17	y	22	23	24	25	26	27	28	29	30	31	cross
8	1	2	3	4	5	6	7				22	0	21	20	19	18	17	16	15	14	13	$8,12$
9		1	2	3	4	5	6	7			23	2	0	21	20	19	18	17	16	15	14	$9,13$
10			1	2	3	4	5	6	7		24	4	2	0	21	20	19	18	17	16	15	$10,14$
11				1	2	3	4	5	6	7	25	6	4	2	0	21	20	19	18	17	16	$11,15$
12	7				1	2	3	4	5	6	26	8	6	4	2	0	21	20	19	18	17	$12,16$
13	6	7				1	2	3	4	5	27	10	8	6	4	2	0	21	20	19	18	$13,17$
14	5	6	7				1	2	3	4	28	12	10	8	6	4	2	0	21	20	19	$14,18$
15	4	5	6	7				1	2	3	29	14	12	10	8	6	4	2	0	21	20	$15,19$
16	3	4	5	6	7				1	2	30	16	14	12	10	8	6	4	2	0	21	$16,20$
17	2	3	4	5	6	7				1	31	18	16	14	12	10	8	6	4	2	0	$17,21$

y	term1	term2	term3	term4	term5	term6	term7
18	$1,4$	$2,3$	$8,17$	$9,16$	$10,15$	$11,14$	$12,13$
19	$2,5$	$3,4$	$9,18$	$10,17$	$11,16$	$12,15$	$13,14$
20	$3,6$	$4,5$	$10,19$	$11,18$	$12,17$	$13,16$	$14,15$
21	$4,7$	$5,6$	$11,20$	$12,19$	$13,18$	$14,17$	$15,16$

Table 4: A More Conservative Central Map for Enhanced TTS ($n = 32,\ m = 24$)

We will not tabulate other results, but simply give the central map at higher dimensions. In the given simple ϕ_2 below, we have (for $\ell \geq 6$) the $(m, n) = (4\ell, 6\ell - 4)$, with security parameters $(u, r, \hbar) = (2\ell - 2, 4\ell - 10, \ell - 1)$, where $\hbar$ is equal to $m - h - 1$ as in Sec. 4.1, or the excess dimension of solution at infinity (after guessing at $n - m$ variables).

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

Note that if $x_0 = x_2 = \cdots = x_{2\ell-6} = 0$ and $x_i = 0$ for all $i = 2\ell - 4 \cdots 4\ell - 3$, then all cross-terms vanish. That is a total of 3ℓ variables, so $h = 3\ell$, and $\hbar = m - h - 1 = \ell - 1$.

B Security Concerns and Assessments of Multivariate PKC

There are two classes of attacks against multivariates cryptosystems: general and specific attacks. Specific attacks cannot function if we design our schemes carefully. General schemes should always function but can be slow. For example, Gröbner Bases can always be computed, but in the general case has a woefully high time bound. We list what we know of attacks against multivariates, and aside from Rank considerations, we refer the reader to the summaries given in [9].

General Attacks: of the following general types

Gröbner Bases Methods: See [9] for summary. References at [2, 5, 6, 26]. Generally regarded as not practical when the "dimension of solution set at infinity" (see [29]) is non-zero.

Searching vs. Signature Schemes: See [9]. In general not practical against tame-like systems.

Rank Attacks: As discussed in the text.

Linearization-like Methods: Traceable from [25] and developing to XL attacks. See below.

Specific Attacks: Bilinear Relations ([35], used against C^* and TTM) not functional against TTS; Separation of Oil and Vinegar ([23, 24], probably not functional against TTS, but just in case, we keep h relatively high in Sec. 4.1 — see [9]); Patarin's IP Approach ([36]), not functional against TTS (see [9]); Attacks on 2R schemes, nonfunctional against TTS; Subspace Attack against SFLASH (see [9]), nonfunctional against TTS.

The discussion above should not be limited to TTS but is concordant with all tame-like systems constructed according to the rules given in Sec. 4.2.

B.1 Assessing Tame-Like Signatures for XL-Like Attacks

XL-like attacks refer to techniques in which *the original equations are multiplied by all monomials up to some degree, then all these resultant equations is solved as a linear system of equation considering every monomial to be a different independent variable.* If there are enough independent equations, the result is a solution of the system if possible, and a return value of "impossible" otherwise. To help the method terminate earlier, it is a good idea to guess at some variables (FXL variant).

This approach was first proposed in [15] as a refinement of its precursor, *relinearization* ([25]). Several variants were proposed since, and [14] summarized them as well as claimed general usefulness of the method used in multivariate schemes of all kinds.

We have however some reason and expert opinions to believe ([10, 40]), that the general approach is slightly overhyped. There are two problems with XL-like attacks. One is the so-called “solution set at infinity” issue. The parameter $\hbar = \dim H_\infty$, which is equal to $m - h + 1$ in a TTS set up according to Sec. 4.2, needs to be eliminated, usually by guessing at variables. The other is that there are more dependencies in the system of equations than what the author counted in [14] and earlier papers. For example, suppose we wish to attack Enhanced TTS with $n = 28, m = 20$ even after guessing at *thirteen* variables, (i.e. $n = 15$, $m = 20$) and a maximum degree of 6 (resp. 5) there are 54264 (resp. 15504) monomials and only 52820 (resp. 13280) of them are independent out of 77520 equations found. One must get to a degree of 7, in which case using the formulas in [14] and using some blocking optimizations, it takes $> 2^{85}$ AES block equivalents to do the entire cryptanalysis.

According to our computations ([1, 10]), a rough guide is that XL methods should operate only if

$$[t^D] \left\{(1-t)^{m-n-1} (1+t)^m\right\} = \sum_{j=0}^{m-n-1} (-1)^j \binom{m-n-1}{j} \binom{m}{D-j}$$

goes negative. Using the assumption that $\hbar$ variables must be guessed, the complexity for FXL to operate is computed to increase roughly at the level of 2^{4m+6}. If we assume that we don’t have to worry about the $\dim H_\infty$ situation and consider all sorts of optimizations including sparse matrix techniques, the best time bounds we can get are as listed in Table 1. Thus, the minimal XL-Like Attack time bounds are roughly concordant with that of Dual Rank Attack time bounds.