TTS: Rank Attacks in Tame-Like Multivariate PKCs

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@chnds.com.tw`

Submitted February 22, 2004; *Third Revision:* September 29, 2004

Abstract

We herein discuss two modes of attack on multivariate public-key cryptosystems. A 2000 Goubin-Courtois article applied these techniques against a special class of multivariate PKC's called "Triangular-Plus-Minus" (TPM), and may explain in part the present dearth of research on "true" multivariates – multivariate PKC's in which the middle map is not really taken in a much larger field. These attacks operate by finding linear combinations of matrices with a given rank. Indeed, we can describe the two attacks very aptly as "high-rank" and "low-rank".

However, TPM was not general enough to cover all pertinent true multivariate PKC's. *Tame-like* PKC's, multivariates with relatively few terms per equation in the central map and an easy inverse, is a superset of TPM that can enjoy both fast private maps and short set-up times.

However, inattention can still let rank attacks succeed in tame-like PKCs. The TTS (Tame Transformation Signatures) family of digital signature schemes lies at this cusp of contention. Previous TTS instances (proposed at ICISC '03) claim good resistance to other known attacks. But we show how careless construction in current TTS instances (TTS/4 and TTS/$2'$) exacerbates the security concern of rank, and show two different cryptanalysis in under 2^{57} AES units.

TTS is not the only tame-like PKC with these liabilities – they are shared by a few other misconstructed schemes. 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. A demonstrative TTS variant with similar dimensions is built for which rank attack takes $> 2^{80}$ AES units, while remaining very fast and as resistant to other attacks. The proposed TTS variants can scale up.

In short: We show that rank attacks apply to the wider class of tame-like PKC's, sometimes even better than previously described. However, this is relativized by the realization that we can build adequately resistant tame-like multivariate PKC's, so the general theme still seem viable compared to more traditional or large-field multivariate alternatives.

1 Introduction

In a sense, this paper describes an episode of the usual balancing act as a cryptologist veers between requirements in speed and security. We will discuss two specialized linear-algebra based attacks against multivariate public-key cryptosystems of a certain type, one that we will term "tame-like".

After we define tame-like PKC's, we will discuss why they are desirable, and how the attacks in question, which we will collectively call "rank attacks", affects the security and design of a tame-like multivariate scheme. Proper criteria for building a good tame-like PKC are given and instances from the TTS family of signature schemes are constructed to be resistant to concerns of rank while retaining good qualities including speed, scalability, flexibility, and minimal resource requirements.

1.1 Multivariate Public-Key Cryptosystems

RSA still "rules" all PKC some 30 years after public-key cryptography was invented ([17]). However, due to current advances in cryptography like number field sieves ([7, 42]), secure RSA applications requires ever-longer keys, which negatively affects the execution speed and cost of deployment.

Multivariate PKCs were introduced as an alternative to cryptosystems with large algebraic structures. A typical multivariate PKC (following notations of [8]) over the base field K has a public map $V = \phi_3 \circ \phi_2 \circ \phi_1 : K^n \to K^m$. 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 affine in K^n and K^m respectively. *The security of the scheme is then based on the NP-hardness* ([22]) *in solving a large system of quadratics 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 the complexity of ϕ_2 — but of this more later.

The currently best-renowned multivariate PKC's, SFLASH^{v2} ([39]) and QUARTZ ([38]), descend from Matsumoto-Imai's $C^\star$ ([28]) and Patarin's HFE ([37]) respectively. Both second-round NESSIE ([33]) digital signature scheme candidates were designed by Patarin-Goubin-Courtois team. The former was eventually recommended for low-cost smart cards. Alas, the security of these candidates is under siege,[1] and their speed and key sizes can still use some improvement.

1.2 Tame-Like Multivariate Public-Key Cryptosystems

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 ([36]). In either family each y_i when written as a quadratic polynomial in the x_j has hundreds of terms, and we cannot invert ϕ_2 without treating all of $\mathbf{y}$ as an element in a larger field, resulting in a time penalty.

Given that multiple variables were introduced for speed originally, it seems natural to investigate alternatives in which each x_i or y_j is *a separate entities, rather than one component of a big field element*, yet ϕ_2^{-1} must remain quickly doable. In this way, the central map does *not* have to be taken over a much larger structure. Such PKC's we term *true multivariates*. We will hereafter concentrate mainly on one subclass of true multivariates, the *Tame-Like Multivariate PKC's.*

A tame-like PKC's is a multivariate whose central map ϕ_2 has relatively few terms in each equation and a speedy inverse image readily available, usually by no more than serial substitution and solving linear systems. As we shall explain, tame-like PKC's are extremely fast and suited for deployment in resource-poor PKI environments. The question is *are they secure enough?*

Early tame-like multivariate PKC's had included Birational Permutation Schemes ([40]) and TTM ([29]). Coppersmith *et al* put paid the former ([10, 11]). 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 ([23]). The techniques they used were not new: One appears to be due to Coppersmith *et al* and the other seems well-known in other circles before introduced to cryptography by Shamir and Kipnis ([4, 10, 11, 26, 43]). But they somewhat expanded the scope and simplified the procedures. They also conveyed the impression that the concept of a faster signature systems than $C^\star$-based ones is beyond redemption. 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 ([8]). As usual, the truth lies somewhere in between.

We will discuss how the rank attacks of Goubin-Courtois function and how well they work in general. We point out liabilities in current TTS instances, in particular, *the non-obvious vulnerability*

[1] Patarin *et al* recently announced that SFLASH^{v2} is not secure enough ([15]), although the cryptanalysis ([13]) is disputed ([49]). 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 ([12, 20]).

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. The result of our suggested repair work seems promising: still lightning-fast, especially suited for embedded implementations but can also excel elsewhere.

(m, n)	PubKey	SecKey	Dual Rank	Rank	XL	RSA bits	ECC bits	Sign @ μs	Setup @ ms	Verify @ ms
16, 22	4400 B	879 B	2^{62}	2^{87}	2^{68}	512	112	34	6.4	0.05
20, 26	7540 B	1254 B	2^{71}	2^{88}	2^{84}	768	128	45	11.5	0.09
20, 28	8680 B	1399 B	2^{80}	2^{120}	2^{84}	1024	144	51	15.1	0.11
24, 32	13440 B	1864 B	2^{88}	2^{121}	2^{95}	1536	160	67	25.4	0.18
24, 34	15096 B	2039 B	2^{96}	2^{153}	2^{95}	2048	176	76	32.8	0.20
28, 38	21812 B	2594 B	2^{105}	2^{154}	2^{110}	2560	192	91	48.4	0.26
28, 40	24080 B	2799 B	2^{113}	2^{186}	2^{110}	3072	208	104	58.6	0.28
32, 44	33088 B	3444 B	2^{121}	2^{186}	2^{125}	4096	224	121	90.0	0.44
32, 46	36064 B	3679 B	2^{129}	2^{218}	2^{125}	5120	240	138	105.0	0.48
36, 50	47700 B	4414 B	2^{138}	2^{219}	2^{136}	6144	256	157	143.0	0.60

Table 1: Security and Performance of Enhanced TTS, (m, n) = hash and signature sizes

As seen in Table 1 (speed tests on a 500 MHz Pentium III PC with gcc3), compared to RSA, the patched TTS variant has good security[2] levels against known attacks, and it signs 3 orders of magnitude faster (cf.Tab. 4). We did basic simulations to make sure that no estimate is out of line. We hope to have somewhat spurred renewed interest in multivariates.

2 TTS As Tame-Like Multivariate PKC's

One (the) obvious idea is to have the x_i *computable in mostly sequential order* when given $\mathbf{y}$. We will in Sec. 3 show this previously attempted approach ([21, 40]) to be not entirely sound.

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

One candidate for a suitable ϕ_2 for a tame-like PKC suggests itself naturally. In algebraic geometry there is a type of map called a *Tame Transformation*. With dimensions $m \geq n$, this 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:

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

K is the *base field*. If bijective, it is a *tame automorphism* over K, in which case obviously $m = n$.

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 ([32]). 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 ([29]). Chen and Yang adapted the underlying concept of TTM for digital signatures and for security concerns extended it

[2]Security Estimates for RSA and ECC taken from NESSIE ([34])
[3]Note that in a finite field just about any function can be represented as a polynomial.

slightly ([8]) 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 [8], we will hence term such maps *tame*. For example, the map below

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

is a tame map, because a preimage can be componentwise computed, straightforwardly and quickly, after assigning any $x_1, \ldots, x_7$ and $x_0 \neq -d_{27}^{-1}$. We see that Shamir's Birational Permutations uses a tame-like middle map. Tame maps are also the centerpiece of TTS ([8]):

> *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".*

ϕ_2 was sometimes also called the *kernel* map, 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[4] (cf. Sec. 1) whenever each *central equation*, i.e. an equation giving a y_i in $\mathbf{x}$ from the cental map, has relatively few terms involving the x_j's compared to the dimensions n and m.

2.2 Current Variants of TTS

In the notation of [8], the public or verification map V of TTS has the canonical decomposition 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 $\mathrm{GF}(2^8)$. The current[5] form of TTS is "TTS/4" ([8]), using 20-byte hashes and 28-byte signatures. Its central map $\phi_2 : \mathbf{x} = (x_0, x_1, \ldots, x_{27}) \mapsto \mathbf{y} = (y_8, y_9, \ldots, y_{27})$ 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 $x_0, \ldots, x_7$, subject to the restrictions $x_i \neq -d_{20+i}^{-1}$ for $i = 4 \cdots 7$, then solving sequentially for $x_8, \ldots, x_{27}$. 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 (cf. [8]):

To Setup Keys: Generate random full-rank 28×28 matrix M_1 and 20×20 matrix M_3 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, 1312 bytes long.

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 $V(\mathbf{w})$.

[4] Compared to a straight tame transformation, a ϕ_2 for TTS seems to be missing a few equations. This is because the public map of a signature scheme need not be injective, so some information can be compressed or projected out.

[5] Boldface indices are irregularities in the pattern of indices made in TTS/4 for security improvements ([8]).

TTS/4 and TTS/2′ claim very fast execution times, short signatures, manageable key lengths, and reasonable security. Previous analysis ([8]) seems to show known attacks to be ineffective. The best attacks previously came from the XL family ([14]). [8] claims good XL-resistance for the TTS family schemes because it can be structured to have high-dimensional solution spaces at infinity ([30]). 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 or 2^{88} finite field multiplications. 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 is design misjudgment in these TTS instances that leads to fast cryptanalysis and how to patch them effectively and generically.

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

The presentation of rank attacks [23] was very broad, one might even say ambitious. The authors postulated a type of PKC called TPM (triangular plus-minus), which is essentially just a multivariate PKC using for its central map (ϕ_2) a tame transformation with some equations lopped off at the beginning and some extraneous equations added. TPM was pronounced to be completely useless due to very general attacks. The implication was left hanging in the air that for serious purposes, no tame-like (or non-C^*-descended, non-HFE-derived) PKC's need not even apply, ever.

In a nutshell, *[23] showed that the private keys of TPM's and some similar tame-like PKC's can be distilled from the public key through seeking linear combinations of certain matrices at given ranks.* 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 [23] describe as a TPM signature scheme. T. Moh ([31]) also maintains that the description does not match TTM.

  Some mismatches: A tame-like signature scheme may solve linear equations rather than search; tame-like PKC's need not have a sequence of increasing kernels in the central equations. The TTM central map comprise more than one part, altering its rank properties.

- If not, can the attack be extended to other tame-like systems?
  Yes, some objections of [31] seem valid, yet 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 attacks will run on all TPM and some other tame-like systems with time costs similar to what is given in [23]. The attack concludes successfully any time 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 (cf. Sec. 3.3). 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

Algorithms seeking linear combinations of matrices at a given rank have been around (cf. [4, 43]), but was first introduced for cryptanalysis by Shamir and Kipnis in [26], against HFE. As presented by the later [23], the basic ideas of a Rank Attack can be encapsulated as follows:

1. When a matrix $M \in K^{n \times n}$ has rank r, then there is a $|K|^{-r}$ probability that a random vector $\mathbf{w} \in K^n$ lies in $\ker M$, a vector space isomorphic to K^{n-r}.

2. When a quadratic function undergoes an invertible change of variables from $\mathbf{x}$ to $\mathbf{w}$ (where $\mathbf{x} = \mathsf{M}_1\mathbf{w} + \mathbf{c}_1$), its highest-degree part $\mathbf{x}^T\hat{Q}_i\mathbf{x}$ becomes $\mathbf{w}^T(\mathsf{M}_1^T\hat{Q}_i\mathsf{M}_1)\mathbf{w}$, or we can say that the new matrix $Q_i = \mathsf{M}_1^T\hat{Q}_i\mathsf{M}_1$ gives the quadratic part of $\mathbf{y}$ in $\mathbf{w}$, and $\operatorname{rank} Q_i = \operatorname{rank} \hat{Q}_i$.

   Usually we want Q_i or $\hat{Q}_i$ written as symmetric form. When $\operatorname{char} K = 2$ as most often is the case, we cannot do so, but the symmetric matrices $H_i = Q_i + Q_i^T$, $\hat{H}_i = \hat{Q}_i + \hat{Q}_i^T$ are uniquely determined (no matter how we choose $\hat{Q}_i$ or Q_i) and also satisfy $H_i = \mathsf{M}_1^T\hat{H}_i\mathsf{M}_1$.

3. When $m \le n$, we expect the system of equations $\sum_{j=1}^m \alpha_j(H_j\mathbf{w}) = \mathbf{0}$ to have no non-trivial solutions for (α_j) most of the time if $\mathbf{w}$ is randomly chosen, because there are too many equations. In contrast, if $m > n$, we can always find such (α_j) regardless of H_i and $\mathbf{w}$.

That given, 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 possible in a central equation or a linear combination of central equations. *For a TPM, but not for a general tame-like system, this will be the initial central equation.* We borrow an illustration from [8], 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):

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

We can write the quadratic part of y_8 as $(\mathbf{x}^TQ\mathbf{x})$ in any way, and $(Q + Q^T)$ will be as shown to the left, and its kernel is $x_0 = x_1 = \cdots = x_7 = 0$. Indeed, if a quadratic has the form $C_{ab}x_ax_b + C_{cd}x_cx_d + \cdots$ with all indices $a, b, c, d, \ldots$ distinct, then the kernel of the corresponding symmetric matrix will be $\{\mathbf{x} : 0 = x_a = x_b = x_c = x_d = \cdots\}$, 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. Indeed, if the kernel of y_8 in $\mathbf{x}$-space is S then the kernel in $\mathbf{w}$-space is $(\mathsf{M}_1)^{-1}S$. In general for ℓ cross-terms with distinct indices, the rank of the matrix is 2ℓ. [23] then gives an attack to break a TPM in expected time $O(q^{\lceil \frac{m}{n} \rceil r}m^3)$. The steps outlined therein are:

1. Let D_i be the symmetric matrices representing the homogeneous quadratic portions of the public keys. That is, if $z_i = \mathbf{w}^T\tilde{D}_i\mathbf{w}$ plus lower terms, then $D_i = \tilde{D} + \tilde{D}^T$. Take P to be $\sum_{i=1}^m \lambda_i D_i$, an undetermined linear combination of the D_i.

2. Guess at a random k-tuple $(\mathbf{w}_1, \ldots, \mathbf{w}_k)$ of vectors in K^n, where $k = \lceil \frac{m}{n} \rceil$, then set $P\mathbf{w}_1 = \cdots = P\mathbf{w}_k = \mathbf{0}$ and attempt to solve for λ_i via Gaussian elimination.

3. *In a TPM, the first linear combination located usually represents the initial central equation.* For a tame-like system in general we should have found the central equation(s) or linear combination(s) thereof with the rank r that is the smallest possible.

   If we only use one test vector $\mathbf{w}$ in the case of an encryption scheme, a non-trivial solution (λ_i) will always exist for the system $\sum_{i=1}^m \lambda_i D_i\mathbf{w} = 0$, since $m > n$ (more variables than equations). I.e. every $\mathbf{w}$ belongs to the kernel of some combination of the D_i. We need multiple test vectors so that their common membership in a kernel of some matrix is significant.

   Suppose $\bar{P}$ is a linear combination of the D_i with rank $\bar{r}$, then its kernel has dimension $n - \bar{r}$, so the probability that the k-tuple of vectors $(\mathbf{w}_1, \ldots, \mathbf{w}_k)$ will all fall in $\ker\ \bar{P}$ is $q^{-k\bar{r}}$. Since the rank usually goes up two at a time, the odds should be overwhelming[6] that we have found a multiple of y_1, and the coefficients λ_i (essentially, a row of M_3^{-1} up to a factor).

For a TPM, $\ker H_1 \subset \ker(H_2 + k_1 H_1) \subset \ker(H_3 + k_2 H_2 + k_1 H_1) \subset \cdots$ with dimensions usually increasing in twos. So odds are $255 : 1$ in favor of this being a multiple of H_1 and not any linear combination involving H_2. We can argue similarly for other tame-like PKC's.

Also, in general we can tell if we hit a non-minimal kernel (one containing other kernels), because this is exactly when the P we find will not be unique up to a constant factor.

Proposition 1 (Time to Find a Vector in any Given Kernel) *Suppose one unique linear combination $H = \sum_{i=1}^{m} \alpha_i D_i$ has the minimum rank r, then the algorithm described above will find $\mathsf{M}_1^T H \mathsf{M}_1$, or rather some vector(s) in the dimension-$(n-r)$ kernel $[\mathsf{M}_1^{-1}(\ker H)]$, with an expected cost of $\approx q^{kr}\left(m^2(nk/2 - m/6) + mn^2 k\right)$ multiplications in the finite field.*

Proof. For each k-tuple $(\mathbf{w}_1, \ldots, \mathbf{w}_k)$ and each pair (i, j) we must evaluate $D_i \mathbf{w}_j$ with n^2 multiplications each, 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 [23] 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[7] 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 do not represent the actual schemes correctly.* 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 [23], the target scheme has $r = 2$. It can be a lot higher. For example, $r = 8$ in TTS/4 and TTS/2′; furthermore, often we can increase this parameter with relative ease. According to [29], the 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 [23], 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.

[6] at least when q^2 is usually fairly large (65536 here)
[7] In a sense, square terms are fundamentally linear.

This is a very double-edged sword. In TTS/2$'$, for non-zero α (and most i), 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 $\alpha^2 a_{i+1}c_{i+1}d_{i+1} = \beta(b_i d_{i+1} b_{i+2} + d_i a_{i+1} d_{i+2})$; in TTS/4, we also have (most of the) $\operatorname{rank}(y_i+\alpha y_{i+1}) = \operatorname{rank}(y_i+\alpha y_{i+2}) = 8$, but there are no three-term combinations with rank 8. We see that in either scheme there are thousands of combinations of the y_i at rank 8, *whose kernels are for the most part disjoint*.

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.

3.3 The "Crawling" Rank Attack vs. TTS/2$'$ and TTS/4

It is on this last point — multiple equal-sized kernels — that we shall show how to extend the venerable technique of rank attack to a cryptanalysis of TTS/2$'$ and TTS/4 with an even lower cost.

Take any given rank 8 central equation, then when $n = 28$ and $m = 20$, according to Prop. 1, we should need $256^8 \cdot [20^2 \cdot (28/2 - 20/6) + 20 \cdot 28^2] \approx 2^{78}$ field multiplications to hit this equation. NESSIE ([33]) requirements are not counted in field multiplications however, but in AES blocks. Using data from the NESSIE performance report ([35]), and comparing with actual operations, we obtain the exchange rate of one AES block to between $\approx 2^5$ and 2^6 finite field multiplications if these are done with tables of logarithms and exponentials. All told, we can expect a time complexity of $\approx 2^{72}$ 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*. For simplicity in illustration, let $a_8 = a_9 = a_{10} = b_8 = \cdots = d_{10} = 1$ in TTS/2$'$, then we have

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

With these coefficients there is no rank-8 combination of y_8, y_9, y_{10}; when such a combination exists, the kernel vectors $\mathbf{x}$ would have $x_2 = x_7 = 0$ and $x_0 : x_4 : x_6 : x_8 = x_1 : x_3 : x_5 : x_9$ in fixed ratios. In the case of TTS/4, we have instead the kernels

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

These kernels show the way to cryptanalysis along these steps:

1. Run the algorithm of Sec. 3.1 to find a vector $\mathbf{u}$ and the 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. 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 as above, except that now we only test random vectors $\mathbf{v} \in U$ until $\mathbf{v}$ lies in more than one kernel; i.e., when we solve $\sum_i \alpha_i D_i \mathbf{v} = 0$, the (λ_i) are not unique up to a constant factor. Find a basis $\{(\alpha_i^{(j)})_{i=1\cdots m}\}_{j=1\cdots s}$ for this solution set in α-space, where $s > 1$, and translate it into $(\hat{y}_j)_{j=1\cdots s}$ in quadratic forms, via $\hat{y}_j = \sum_{i=1}^m (\alpha_i^{(j)} z_i)$. When this does happen, we expect the dimension s to be 2 or 3.

In the unlikely case that we find two distinct sets of results ($\mathbf{v}$ and $(\hat{y}_i)$) in maybe 5000 tests, the initial (λ_i), or rather the quadratic form $\sum_{i=1}^{m} \lambda_i z_i$ that these coefficients correspond to, is likely a multiple of some y_i where $9 \leq i \leq 25$. The two different kernels we find should be $\{\mathbf{x} : x_{i-9} = x_{i-8} = \cdots = x_{i-1} = 0\}$ and $\{\mathbf{x} : x_{i-8} = \cdots = x_{i-1} = x_i = 0\}$. The two solution spaces must then correspond to $\mathrm{span}(y_i, y_{i\pm1})$.

More likely we locate only a single multi-dimensional solution space for the (α_i) in 10000 tries, that should be $\mathrm{span}(y_i, y_{i+1})$ or $\mathrm{span}(y_i, y_{i+1}, y_{i+2})$ depending on its dimension. For 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 to hit 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 $\hat{y}_i$ we find, modulo constant factors, 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 $\hat{y}_i$ 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 [23]. The upshot is that a solution can be located in between 2^{64} to 2^{65} multiplications (or 2^{58} to 2^{59} 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 or Athlon) PC's, the above projected cryptanalysis process 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

A natural converse[8] 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. Let the fewest number of appearances of all variables in the cross-terms of the central equations be a total u times, and without loss of generality let this be the last variable x_{n-1}. In TTS/4, this is x_{27}, which only appears in y_{27}. In the earlier TTS/2 (cf. [8]), x_{27} does not appear in any cross-term . In Birational Permutation Schemes ([40]), the last central variable appear in only one equation. Here we try to describe concisely why u cannot be too small:

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

1. If x_{n-1} does not appear in any y_j, then every matrix $\hat{H}_j$ will have zeros for the entire last row and column. Thus the intersection of all the $\ker_{\mathbf{x}} z_j$ (and hence the $\ker_{\mathbf{w}} z_j = \ker D_j$) will be non-empty and contain the subspace corresponding to $U_{n-1} = \{\mathbf{x} : x_0 = \cdots = x_{n-2} = 0\}$.

2. Suppose x_{n-1} only appear in a cross-term in one central equation, say y_{n-1}. Then whenever $\alpha_{n-1} = 0$, the matrix $P = \sum_i \alpha_i H_i$ will again contain the subspace U_{n-1}. Indeed, denote by m_{ij} the (i, j)-entry of M_3, and we see that for every pair of indices (i, j) there will be a linear combination, namely $m_{j,n-1} D_i - m_{i,n-1} D_j$ whose kernel contain the same subspace U_{n-1}.

   We can extend the above to: *for all indices $i < j$, if we cannot find a c_{ij} such that $U_{n-1} \subset \ker(D_j + c_{ij} D_i)$, then $m_{i,n-1} = 0$* — so we can quotient x_{n-1} out of the system.

3. In general, with almost any $(u+1)$-subset picked from the D_i, there is a unique linear combination of these matrices with a kernel containing the common subspace U_{n-1}.

Now we must exploit this critical weakness ([11]) by finding linear combinations $\sum_i \alpha_i z_i$ whose kernels share a non-empty intersection, which Coppersmith-Stern-Vaudenay ([10, 11]) did elegantly *without needing to search*, in a way that can be extended to find an ascending chain of kernels in the matrix algebra over a ring. This neatly broke Birational Permutations. The basic C-S-V lemma is:

> When $P = D_j + \lambda D_i$ is the linear combination whose kernel contains U_{n-1}, then P has a characteristic polynomial $f(x) = \det(D_j - \lambda D_i - xI)$ *with double roots*. Hence, if we solve the resultant of $f'(x)$ and $f(x)$ as an equation in λ, that should be our c_{ij}.

[23] carried out the same Dual Rank Attack[9] by searching, while mistakenly comparing the C-S-V idea to those in Sec. 3.1. As an equivalent formulation, the essence of the more plebian G-C version of the Dual Rank Attack (to find U_{n-1}) can be distilled as follows:

1. Form an arbitrary linear combination $P = \sum_i \alpha_i D_i$; find $V = \ker P$ by Gaussian elimination.

2. When $\dim V \geq 1$, set $(\sum_j \lambda_j D_j)V = \{\mathbf{0}\}$ 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 $V = U_{n-1}$. We can then expand to find bigger kernels.

One trial costs an 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 only $(u+1)$ of the matrices D_i, and don't get too unlucky.

The method of Coppersmith *et al* is still applicable in expanding to larger kernels, but for a TPM, or even the non-TPM TTS/2′ and TTS, it is a lot easier. The next bigger kernel up the chain, which is $U_{26} = \{x_0 = x_1 = \cdots = x_{25} = 0\}$, can be found by looking at subspaces of $V = U_{27}$, which will get us U_{26} with probability $1/q$. After we have the entire sequence of kernels, the cryptanalysis is almost complete, so for TTS/4 and TTS/2′, the cryptanalysis can be almost instant.

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

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

$y_j = x_i + A_j x_h x_\ell$. That is an equation of rank 2. The presentation in [23] 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, where the effort in each attempt is some substitutions plus a Gaussian Elimination.

There is a moral to learn from this episode. People noticed the impact of rank in multivariate cryptography early on. For example, Theobald was impressed enough to issue a warning ([44]) *"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:

> 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, main concerns for a tame-like multivariate PKC must surely be 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 equations should have 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, only a relatively small number can have equal rank.*

   If k linear combinations of central equations share a 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. *Let A be the smallest maximal set of v indices $0 \le i < n$ such that every cross-term in the central map has at least one index in A, then we require*

$$q^{2v-n-1}(n-v)^4 \ge C. \tag{3}$$

This is to avoid concerns that plagued Oil-and-Vinegar (OV) and unbalanced OV schemes ([24, 25]). Within this restraint, lower v means higher XL/FXL complexity (see Appendix).
5. *Let $D_0 = D_0(m,k) := \min\{D : [t^D]\left((1-t)^{k-1}\,(1+t)^m\right)$, $T := \binom{m-k+D_0}{D_0}$, $R = \binom{m-k+D_0-2}{D_0-2}$ then*

$$\min_k q^k \cdot m^2 RT(c_0 + c_1 \lg T) < C \tag{4}$$

where $c_0 \approx 4$, $c_1 \approx 1/4$ are constants.

Item 5 results from XL/FXL ([12, 13, 14, 47, 49]) and Gröbner-based attacks ([2, 19, 20]). One should refer to [30] for some algebraic geometry on XL-type attacks, and to [5, 6, 27] on Gröbner Bases theory. 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 the Appendix.

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 $m \ge 20$. Obviously $n > m$ in a signature scheme like TTS. We need $r \le 10$, 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 [8]); (b) makes securing them against XL attacks harder (see Appendix 5.2); 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.

$$\begin{aligned}
y_i &= x_i + \textstyle\sum_{j=1}^{7} p_{ij} x_j x_{8+(i+j \bmod 9)},\ i = 8 \cdots 16;\\
y_{17} &= x_{17} + p_{17,1}x_1x_6 + p_{17,2}x_2x_5 + p_{17,3}x_3x_4\\
&\qquad + p_{17,4}x_9x_{16} + p_{17,5}x_{10}x_{15} + p_{17,6}x_{11}x_{14} + p_{17,7}x_{12}x_{13};\\
y_{18} &= x_{18} + p_{18,1}x_2x_7 + p_{18,2}x_3x_6 + p_{18,3}x_4x_5\\
&\qquad + p_{18,4}x_{10}x_{17} + p_{18,5}x_{11}x_{16} + p_{18,6}x_{12}x_{15} + p_{18,7}x_{13}x_{14};\\
y_i &= x_i + p_{i,0}x_{i-11}x_{i-9} + \textstyle\sum_{j=19}^{i-1} p_{i,j-18}\ x_{2(i-j)-(i \bmod 2)}\ x_j + p_{i,i-18}x_0x_i\\
&\qquad + \textstyle\sum_{j=i+1}^{27} p_{i,j-18}\ x_{i-j+19}\ x_j,\ i = 19 \cdots 27.
\end{aligned}$$

To see ϕ_2 more clearly and that it meets our requirements, we tabulate it differently in Table 2. Of course, we will need to scale up the new variant if our estimate is somewhat off, or to meet future, higher security requirements. We will discuss this next in Sec. 4.3.

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	3	1	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	7	5	3	1	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	11	9	7	5	3	1	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	15	13	11	9	7	5	3	1	0	16, 18

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

The ϕ_2 given above can be inverted 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. 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). 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 8 variables are guessed.

4.3 Scaled-Up Versions of Enhanced TTS

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 an example is given in Appendix A.

However, we can scale the above up to provide for a whole sequence of TTS instances, which we will call the "odd sequence" because the parameter u is odd. We have (for $\ell \geq 4$) the $(m, n) = (4\ell, 6\ell - 2)$, with security parameters $(u, r) = (2\ell - 1, 4\ell - 6)$

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

There is a different sequence of Enhanced TTS instances with the same Rank Attack estimates, accounting for an even more conservative estimate for XL/FXL attacks, as in the equations below

$$
\begin{aligned}
y_i &= x_i + \textstyle\sum_{j=1}^{7} p_{ij} x_j x_{8+(i+j+1 \bmod 10)},\ i = 8 \cdots 17;\\
y_i &= x_i + p_{i1} x_{i-17} x_{i-14} + p_{i2} x_{i-16} x_{i-15} + p_{i3} x_{i-10} x_{i-1} + p_{i4} x_{i-9} x_{i-2}\\
&\qquad + p_{i5} x_{i-8} x_{i-3} + p_{i6} x_{i-7} x_{i-4} + p_{i7} x_{i-6} x_{i-5},\ i = 18 \cdots 21;\\
y_i &= x_i + p_{i,0} x_{i-10} x_{i-14} + \textstyle\sum_{j=22}^{i-1} p_{i,j-21}\ x_{2(i-j)-(i \bmod 2)}\ x_j + p_{i,i-21}\ x_0 x_i\\
&\qquad + \textstyle\sum_{j=i+1}^{31} p_{i,j-21}\ x_{i-j+21}\ x_j,\ i = 22 \cdots 31.
\end{aligned}
$$

This larger ϕ_2 (also cf. Tab. 3) has $(n, m) = (32, 24)$ and an FXL complexity about $2^{16}\times$ higher.

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	1	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	5	3	1	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	9	7	5	3	1	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	13	11	9	7	5	3	1	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	17	15	13	11	9	7	5	3	1	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 3: A More Conservative Central Map for Enhanced TTS ($n = 32,\ m = 24$)

We give the central map at higher dimensions for these Enhanced TTS instances, called the "even sequence" because the parameter u is even. In the given ϕ_2 below, we have (for $\ell \geq 5$) the $(m, n) = (4\ell, 6\ell - 4)$, with security parameters $(u, r, v) = (2\ell - 2, 4\ell - 10, 4\ell - 2)$.

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

5 Discussion and Conclusion

We provide an executive summary of what we see in tame-like PKC's and especially signature schemes, which we will (hopefully) justify in what remains of this section:

Advantages of tame-like PKC's are speed, ease of implementation, and avoiding old attacks.

Fast Signing: In SFLASH^{v2}, the signing action includes 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 non-trivial procedure! — almost a thousand times. This is no problem on a modern PC, but setting up on-card for the SFLASH^{v2} takes a *long* time. In a tame-like PKC, with few terms per equations, we can setup quickly by brute-force.

Avoidance of Previous Liabilities: There are many possible tame-like maps ϕ_2, so we can dodge or alleviate some weaknesses including those that SFLASH^{v2} must design around.

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

5.1 Does Our Patched TTS Instances Measure Up?

The answers seems to be: yes in all categories, as shown in Tab. 4 below:

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 4: TTS and NESSIE round 2 candidates signature schemes on a 500MHz Pentium III

The public (encryption) does not change from TTS/4 to TTS (20,28), but the running time listed here differ from the data in [8]. We may attribute most of the speed up to the improvements in version 3 of gcc, the very popular GNU compiler[10] that we had upgraded to.

TTS verifies fast due to its smaller dimensions. E.g. SFLASH^{v2} requires larger dimensions chiefly to cater to the vagaries of security requirements for C^*-derivatives. In checking (verifying) a signature we run the public map which takes time approximately equal to $mn^2/2$ multiplications, which depends only on the hash and signature size and not on any details of central map.

TTS signs fast due to the structure of its public map. Like other multivariates, ϕ_1^{-1} and ϕ_3^{-1} costs about n^2 and m^2 (784 and 400 for TTS (20,28)) multiplications respectively while ϕ_2^{-1} does about $n\ell+2(\ell^3/3+\ell^2)$, where $\ell \approx m/2$. As the dimensions grow, $n \sim 3m/2$. The entire signing operation will take around $m^2(c_2 + m/12)$ multiplications in the field, where the quadratic term coefficient c_2 is around 4.5, and will be larger than 4. The implication is that for practical ranges, the speed of signatures in TTS goes down in speed, more inversely quadratic in m than the cubic.

Signing in n-bit RSA should cost a sequence of exponentiations and multiplications of length proportional to n, when each multiplication costs $\sim 2n^2$ in multiplications or divisions[11]. So we expect signing time to be $\propto n^3$. Assuming that we multipliies and divides by c-bit chunks, the cost should be greater than $2n^3/c^2$ multiplicative operations, which for $c = 32$ and $n = 1024$ is about 2,000,000. TTS (20,28) does about 2,000. So TTS should be 1,000 times faster. Since the security of each scheme increases in similar fashion, this should hold at all equivalent practical security levels.

We see that the above rough guide is good to the order of magnitude (cf. Tab. 4). Now look at SFLASH^{v2}. Out of several proposed methods, the most expedious inverse map for its central map involves solving a system of linear equations that can be written as

$$\mathbf{x}\mathbf{y}^{q^a} = \mathbf{x}^{q^b}\mathbf{y}, \quad \mathbf{x}, \mathbf{y} \in (\mathrm{GF}(q))^n$$

[10] Our previous tests used gcc-2.96, but gcc3 was used by some other NESSIE entrants for their timings.

[11] Example: an expert, Prof. Brad Lucier of Purdue U., opines that better algorithms (e.g. FFT or Karatsuba) start to multiply very slightly better than n^2 at around 1400 bits and divide better than n^2 only at around 6800 bits.

where a, b are parameters roughly proportional to n. Raising to any q^k-th being linear with a fixed matrix, the system can be set up and solved in αn^3 multiplications, where α has the order of "a few".

Since C^{*--} signature schemes also seems to follow roughly $n \sim 3m/2$, a back-of-the-envelope computation shows that at roughly equal security levels, TTS would be about two orders of magnitude faster than SFLASH, and this is again in accord with Tab. 4. Since SFLASH^{v2} had been the signing speed demon, *TTS (20,28) — while a factor of two slower than the superseded versions in [8] — is indubitably snappy. 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.*

One might say that the Achilles' heel for the extended family of multivariate PKC's has always been the novelty and untested status, but the next weakness to be mentioned will likely the size of the public keys. To a large extent, this is no longer a problem for non-embedded applications, even though 100-kilobyte public keys might conceivably sometimes pose a problem for electronic commerce. The most practical way around this bugbear for smart cards is to avoid having the public key stored on-card. However, since modern security often dictates keys to be *generated* on-card, we prefer greatly that it be possible to generate public keys from a small amount of stored information.

The next question is: Why is the generation of keys fast for tame-like systems (again cf. Tab. 4)?

Let the (i, j)-position matrix element in D_k be R_{ijk} and that for H_k be $\hat{R}_{ijk}$. Then $R_{ijk} = \sum_\ell (\mathsf{M}_3)_{k\ell} \hat{R}_{ij\ell}$, computable by n^2 matrix multiplications at m^2 multiplications per, and

$$\hat{R}_{ijk} = \sum_{p\, x_\alpha x_\beta \text{ in } y_k} p \cdot ((\mathsf{M}_1)_{\alpha i}(\mathsf{M}_1)_{\beta j} + (\mathsf{M}_1)_{\alpha j}(\mathsf{M}_1)_{\beta i}).$$

Where the sum is taken over all the cross-terms in the central equation giving y_k. If the average number of cross-terms (total cross-terms divided by equations) is t, then we can do this in $3tmn^2$ multiplications, for a total of $m(m + 3t)n^2$. For the dimensions involved, t is usually proportional to m, so the growth in time is quartic. According state-of-the art ([46]) key generation for multivariates using other methods (variations of polynomial interpolation) is sixth-order, so we can expect a difference of at least two orders of magnitudes in setup timings for TTS (or another tame-like signature scheme) and SFLASH, which is borne out by Tab. 4.

We wish to point out: the fact that verification and key generation varies respectively as the cubic (mn^2) and the quartic (m^2n^2) in the dimensions can be seen from Tab. 1. The eventually cubic behavior of signing time is not apparently visible at the dimensions that we are using, but it can be seen that the increase is somewhat more than in the quadratic.

5.2 Security for Tame-Like Signature Schemes and Especially TTS

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 [8].

General Attacks: of the following general types

Gröbner Bases Methods: References at [1, 2, 5, 6, 27]. See Appendix B.

Searching vs. Signature Schemes: Dominated by algebraic methods and in general not practical against tame-like systems ([8]).

Rank Attacks: As discussed in the text.

Linearization-Type: Traceable from [26] and developing to XL attacks. Appendix B below.

Specific Attacks: Bilinear Relations ([36], used against C^* and TTM) not functional against TTS; Separation of Oil and Vinegar ([24, 25], guarded against; Patarin's IP Approach ([37]), not functional against TTS (see [8]); Attacks on 2R schemes, nonfunctional against TTS; Subspace Attack against SFLASH (see [8]), 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. What we can say about the security of TTS?

The main condition for XL and current Gröbner Bases methods to operate is given in the Appendix. If we assume an optimistic bound for the equation-solving phase, the XL/FXL family of methods will take the amount of time as given in Tab. 1.

5.3 A Summary

We have described 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.

Remark: We thank Prof. J. Ding of U. Cincinnati for helpful discussion about possible vulnerabilities. This preprint will be suitably updated soon.

References

[1] Ars, Faugére, Sugita, Kawazoe, and Imai, *Comparing XL and Groebner Basis Algorithms over Finite Fields* (title may be changed), to appear, ASIACRYPT 2004.

[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, *A More Secure and Efficacious TTS Scheme*, ICISC '03, LNCS V. 2971, pp. 320-338; expanded preprint available at `http://eprint.iacr.org/2003/160`.

[9] D. Coppersmith, private communication.

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

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

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

[13] N. Courtois, *Algebraic Attacks over* $\mathrm{GF}(2^k)$*, Cryptanalysis of HFE Challenge 2 and SFLASH*^{v2}, PKC '04, LNCS V. 2947, pp. 201-217.

[14] 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.

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

[16] C. Diem, *The XL-algorithm and a conjecture from commutative algebra*, ASIACRYPT'04, to appear.

[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 (F4)*, Journal of Pure and Applied Algebra, 139 (1999), pp. 61–88.

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

[20] 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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[37] 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.

[38] 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`

[39] 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`

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

[41] A. Shamir, private communication.

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

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

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

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

[46] C. Wolf, *Efficient Public Key Generation for Multivariate Cryptosystems*, preprint, available at `http://eprint.iacr.org/2003/089/`.

[47] B.-Y. Yang and J.-M. Chen, *Theoretical Analysis of XL over Small Fields*, ACISP 2004, LNCS V. 3108, pp. 277-288.

[48] B.-Y. Yang, J.-M. Chen, and Y.-H. Chen, *TTS: High-Speed Signatures from Low-End Smartcards,*, CHES '04, LNCS V. 3156, pp. 371-385.

[49] B.-Y. Yang, J.-M. Chen, and N. Courtois, *On Asymptotic Security Estimates in XL and Gröbner Bases-Related Algebraic Cryptanalysis*, to appear at ICICS '04 and LNCS.

A An Instance of a Differently-Formed Enhanced TTS

Each row in Tab. 5 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}.

y	7	6	5	4	3	2	1	cross
8	8	9	10	11	12			$1,2$
9	9	10	11	12			13	$2,3$
10	10	11	12			13	14	$3,4$
11	11	12			13	14	15	$4,5$
12	12			13	14	15	16	$5,6$
13	13		14	15	16		17	$6,2$
14	14	15	16	17		8		$1,3$
15	15	16	17		8		9	$2,4$
16	16	17		8		9	10	$3,5$
17	17		8		9	10	11	$4,6$

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

Table 5: A Different Central Map for Enhanced TTS

B Assessing Tame-Like Signatures for XL-Like Attacks

XL-like attacks are 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 [14] as a refinement of its precursor, *relinearization* ([26]).

XL only operates on determined or over-determined systems, i.e. $\ell_1(\mathbf{x}) = \cdots \ell_m(\mathbf{x}) = 0$, where $n(\leq m)$ is the dimension of $\mathbf{x}$. With more variables than equations, we must guess at enough variables so as to have at least as many equations as variables. XL at degree D then goes:

1. Take all monomials $\mathbf{x}^{\mathbf{b}} = x_1^{b_1} x_2^{b_2} \cdots x_n^{b_n}$ with total degree $|\mathbf{b}| = \sum_i b_i \leq D-2$, and generate all equations $\mathbf{x}^{\mathbf{b}} \ell_i(\mathbf{x}) = 0$. Call these equations $\mathcal{R}^{(D)}$.

2. The set $\mathcal{T} = \mathcal{T}^{(D)}$ of monomials of total degree $\leq D$ has $T = \binom{n+D}{D}$ elements. Run an elimination on the $R = m\binom{n+D-2}{D}$ equations $\mathcal{R}^{(D)}$, treating each $\mathbf{x}^{\mathbf{b}} \in \mathcal{T}^{(D)}$ as a variable. I, The number of independent equations cannot exceed $T-1$ when the system is solvable. We may conclude the elimination with an equation to solve for x_1 (say), if the row echelon form has a final equation containing up to the $D+1$ terms $1, x_1, \ldots, x_1^D$. To achieve this we only need $I \geq T - D$ (instead of $T = I - 1$).

3. *If necessary, solve the univariate equation giving x_1, and repeat as needed.*

Since there are T monomials (including 1) and R equations, the time complexity of XL is whatever time to solve a system of equations that big. For some rave reviews, see [13] and other papers.

We have however some reason and expert opinions to believe ([9, 41]), that the general approach is slightly overhyped. There are two problems with XL-like attacks. One is the so-called $\dim H_\infty$ parameter, which 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 [13] and earlier papers. For example, attacking Enhanced TTS with $n = 28$, $m = 20$ even after guessing at *thirteen* variables, (i.e. $n = 15$, $m = 20$) using 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 [13] and using some blocking optimizations, it takes $> 2^{85}$ AES block equivalents to do the entire cryptanalysis.

According to our computations ([16, 47, 49]) a rough guide is for XL methods to 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. According to the latest reports, the advanced Gröbner Bases methods of $\mathbf{F_4}$ and $\mathbf{F_5}$ are almost equivalent to the XL+XL2 combination. These algorithms (XL+XL2, $\mathbf{F_4}$, and $\mathbf{F_5}$) have a slightly lower operative degree, but cannot use sparse matrix algorithms as well as the original XL. Furthermore, all of these must be combined with guessing at an optimized number of variables. Please see [49] for an analysis. Assuming Strassen-like elimination of order $\lg 7 \approx 2.8$, XL/FXL will beat $\mathbf{F_4}$ and $\mathbf{F_5}$ by a small margin.

If we assume that we don't have to worry about the $\dim H_\infty$ situation and consider all sorts of optimizations including the sparse matrix techniques, the best time bounds we can get (which are computed case-by-case) 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.

Contents