The Public-Key Encryption System $SRTPI\left(\mathbb{F}_q, n_1, n_2, n_3\right)$ And The Digital Signature $TPSig\left(\mathbb{F}_q, n_1, n_2, n_3\right)$ Algorithm Specifications And Supporting Documentation

Yossi (Joseph) Peretz and Nerya Granot
Department of Computer Sciences
Lev Academic Center, Jerusalem College of Technology (JCT)
P.O.B. 16031 Jerusalem, Israel
Phone: 972-2-6751016, Fax: 972-2-6751046
E-mail: yosip@g.jct.ac.il
E-mail: neryagr@gmail.com

Contents

List of Tables

List of Algorithms

1 A complete written specification

Encryption schemes based on Multivariable Quadratic Equations (MQE) over finite fields are among the most promising schemes for Post-Quantum Cryptography (PQC), since they are based on NP-complete problem and on the conjecture that unless $P = NP$, quantum computers cannot solve NP-complete problems efficiently. Many encryption schemes based on MQE were suggested and, unfortunately, many of them were broken (see [31] and [29]). The broken systems were found to have some hidden structure, which on one hand enabled efficient invertibility of the set of equations, but on the other hand was vulnerable to algebraic attacks. Most of the broken MQE based schemes, share the common deficiency that some quadratic forms associated to their central map have a low rank (see [30]) and therefore were vulnerable to the Min-Rank Attack (see [21]). On the other hand, the (experimentally assessed) belief that random quadratic systems are hard to solve on average (see [11], [5] and references therein), is directing towards designing trap-door primitives that are based on randomness, which raises difficulties in designing completely invertible immune primitives. Little was done in this direction regarding asymmetric public-key cryptography (see [11]). This was the rational underlying the following research project.

An overview of Multivariate Public-Key Cryptography (MPKC) is given in [15], where the authors call for a unifying framework for cryptanalysis of MPKC systems, in order to build confidence in their security. The potential of applications of such systems in the realm of limited computing power is also pointed out there, e.g. in Radio Frequency Identification Devices (RFID) and in Wireless Sensing (WS), where other cryptographic systems (e.g. RSA, ELGAMAL, ECC) are irrelevant. The main developments in the cryptanalysis of MQE's schemes are summarized in [13] and in [5].

1.1 Preliminaries

The following contains a minimal mathematical background needed to understand the following suggested systems. Let $\mathbb{F}$ denote any field. Non-symmetric Algebraic Riccati Equation (ARE) over $\mathbb{F}$ is an equation of the form:

$$\mathcal{R}(X) := XCX + XD - AX - B = 0, \tag{1}$$

where A, B, C, D are $m \times m, m \times n, n \times m, n \times n$ matrices and the solution X is a $m \times n$ matrix over $\mathbb{F}$. The operator $Y = \mathcal{R}(X)$ is called the Riccati operator. The complexity of computing X is equivalent to the complexity of the constrained generalized eigenvalue-eigenvector problem defined by:

$$T \begin{bmatrix} X \\ I \end{bmatrix} = \begin{bmatrix} X \\ I \end{bmatrix} L,\ T = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \tag{2}$$

where $\begin{bmatrix} X \\ I \end{bmatrix}$ is the constrained generalized eigenvector and the $n \times n$ matrix $L = CX + D$ can be interpreted as a generalized eigenvalue. The Non-symmetric Simultaneous Algebraic Riccati Equations problem (NSARE) is the following problem: given t quadruples: (A_i, B_i, C_i, D_i), of compatible seizes, where $i = 1, \ldots, t$, find X over $\mathbb{F}$ such that all the equations:

$$XC_iX + XD_i - A_iX - B_i = 0, \tag{3}$$

are satisfied simultaneously for $i = 1, \ldots, t$. Equivalently, the problem can be stated as the simultaneous constrained generalized eigenvalue-eigenvector problem of finding X such that:

$$T_i \begin{bmatrix} X \\ I \end{bmatrix} = \begin{bmatrix} X \\ I \end{bmatrix} L_i,\ T_i = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}, \tag{4}$$

where $L_i = C_iX + D_i$ for $i = 1, \ldots, t$.

In [24], two asymmetric public-key encryption schemes, based on the NSARE problem, were introduced. The schemes are named Tsafenat-Panneach-I (TP-I) and Tsafenat-Panneach-II (TP-II). In the TP-I scheme, the message is spread into r parts which are encrypted in r parallel independent rounds, while in the TP-II scheme, the message is iteratively encrypted in r independent rounds. The basis for the security proof of the TP-I and the TP-II schemes, is the following theorem (see [24]):

Theorem 1.1 *NSARE is NP-hard over any field $\mathbb{F}$ and is NP-complete over any finite field $\mathbb{F}$.*

It follows that any set of multivariable polynomial equations can be reduced (through a polynomial-time reduction) to the NSARE problem (the converse is obvious) and thus any encryption scheme based on multivariable polynomial set of equations can be crypt-analyzed to vulnerabilities by investigating the related equivalent NSARE problem.

In the sequel we will make use of following theorems and lemmas (see [24]):

Theorem 1.2 *Let A be a $m \times n$ matrix over $\mathbb{F}$ with $r = rank\,(A)$. Let $A = FG$ be a full-rank factorization of A over $\mathbb{F}$, where F and G are $m \times r$ and $r \times n$ full-rank matrices. Assume that $\left(F^T F\right)^{-1}$ and $\left(GG^T\right)^{-1}$ exist over $\mathbb{F}$. Let*

$$A^+ = G^T \left(GG^T\right)^{-1} \left(F^T F\right)^{-1} F^T.$$

Then, A^+ is the unique matrix over $\mathbb{F}$ satisfying:

$$AXA = A,\ XAX = X, (AX)^T = AX,\ \ (XA)^T = XA.$$

The matrix A^+ is called the Moor-Penrose Pseudo-Inverse of A.

In the sequel we will use the following lemmas:

Lemma 1.1 *Let A be a $m \times \ell$ matrix over a field $\mathbb{F}$, for which A^+ exists over $\mathbb{F}$. Then:*

1. Let B be a $m \times n$ matrix over $\mathbb{F}$. A solution to the equation $AX = B$ exists over $\mathbb{F}$ if and only if $AA^+B = B$. In this case the set of all solutions is given by: $X = A^+B + (I_\ell - A^+A)\,Y$, where Y is arbitrary $\ell \times n$ matrix over $\mathbb{F}$.

2. Let B be a $n \times \ell$ matrix over $\mathbb{F}$. A solution to the equation $XA = B$ exists over $\mathbb{F}$ if and only if $BA^+A = B$. In this case the set of all solutions is given by: $X = BA^+ + Y\,(I_m - AA^+)$, where Y is arbitrary $n \times m$ matrix over $\mathbb{F}$.

Theorem 1.3 *Let A, B, C, D be a given matrices over $\mathbb{F}$, with sizes $m \times m, m \times n, n \times m, n \times n$, resp. Let $Y = \mathcal{R}\,(X) := XCX + XD - AX - B$ denote the related Riccati operator $\mathcal{R} : \mathbb{F}^{m \times n} \to \mathbb{F}^{m \times n}$. Let $Y \in \mathbb{F}^{m \times n}$ be given. Then, there exist $X \in \mathbb{F}^{m \times n}$ such that $Y = \mathcal{R}\,(X)$ (equivalently, the set $\mathcal{R}^{-1}\,(Y)$ is* *nonempty) if and only if there exist invertible matrix* $V = \begin{bmatrix} V_{1,1} & V_{1,2} \\ V_{2,1} & V_{2,2} \end{bmatrix}$ *with invertible* $n \times n$ *block* $V_{2,2}$ *such that:*

$$V^{-1}TV = \begin{bmatrix} \widehat{A} & 0 \\ \widehat{C} & \widehat{D} \end{bmatrix}, \tag{5}$$

where $T = \begin{bmatrix} A & B+Y \\ C & D \end{bmatrix}$*.*

Proof:
If there exist $X \in \mathbb{F}^{m\times n}$ such that $Y = \mathcal{R}(X)$, let $V = \begin{bmatrix} I_m & X \\ 0 & I_n \end{bmatrix}$. Then, V is invertible with $V_{2,2} = I_n$ invertible. Moreover, $V^{-1} = \begin{bmatrix} I_m & -X \\ 0 & I_n \end{bmatrix}$ and

$$\begin{aligned} V^{-1}TV &= \begin{bmatrix} I_m & -X \\ 0 & I_n \end{bmatrix} \begin{bmatrix} A & B+Y \\ C & D \end{bmatrix} \begin{bmatrix} I_m & X \\ 0 & I_n \end{bmatrix} \\ &= \begin{bmatrix} A - XC & -XCX - XD + AX + B + Y \\ C & D + CX \end{bmatrix} \\ &= \begin{bmatrix} A - XC & Y - \mathcal{R}(X) \\ C & D + CX \end{bmatrix} \\ &= \begin{bmatrix} \widehat{A} & 0 \\ \widehat{C} & \widehat{D} \end{bmatrix}, \end{aligned}$$

where $\widehat{A} = A - XC, \widehat{C} = C, \widehat{D} = D + CX$.
Conversely, if there exist invertible matrix V with invertible $n \times n$ block $V_{2,2}$ such that (5) is satisfied then, comparing the $2,2$ blocks of

$$TV = V \begin{bmatrix} \widehat{A} & 0 \\ \widehat{C} & \widehat{D} \end{bmatrix}, \tag{6}$$

we have: $CV_{1,2} + DV_{2,2} = V_{2,2}\widehat{D}$, from which we conclude that

$$\widehat{D} = V_{2,2}^{-1}CV_{1,2} + V_{2,2}^{-1}DV_{2,2}. \tag{7}$$

Comparing the $1,2$ blocks of (6), we get $V_{1,2}\widehat{D} = AV_{1,2} + (B+Y)V_{2,2}$, from which we conclude that:

$$V_{1,2}\widehat{D}V_{2,2}^{-1} = AV_{1,2}V_{2,2}^{-1} + B + Y. \tag{8}$$

Finally, substitution of (7) into (8) yields:

$$V_{1,2}V_{2,2}^{-1}CV_{1,2}V_{2,2}^{-1} + V_{1,2}V_{2,2}^{-1}D = AV_{1,2}V_{2,2}^{-1} + B + Y,$$

which is just $Y = \mathcal{R}(X)$ with $X = V_{1,2}V_{2,2}^{-1}$. ■

We conclude that solving Riccati equation $Y = \mathcal{R}(X)$ is equivalent to finding block-lower-triangular form of T, with the restriction that $V_{2,2}$ has to be invertible.

Theorem 1.4 *Let A, B, C, D be a given matrices over $\mathbb{F}$, with sizes $m \times m, m \times n, n \times m, n \times n$, resp. Let $Y = \mathcal{R}(X) := XCX + XD - AX - B$ denote the related Riccati operator $\mathcal{R} : \mathbb{F}^{m \times n} \to \mathbb{F}^{m \times n}$. Let $Y \in \mathbb{F}^{m \times n}$ be given and let $T = \left[\begin{array}{cc} A & B+Y \\ C & D \end{array}\right]$. Then, there exist $X \in \mathbb{F}^{m \times n}$ such that $Y = \mathcal{R}(X)$ if and only if there exist n-dimensional T-invariant subspace $\mathcal{M} \subseteq \mathbb{F}^{(m+n)\times 1}$, such that if $v_1, \ldots, v_n$ is a basis for $\mathcal{M}$ and each $v_j, j = 1, \ldots, n$ is partitioned as $v_j = \left[\begin{array}{c} z_j \\ w_j \end{array}\right]$, where w_j has size $n \times 1$, then $w_1, \ldots, w_n$ are independent. Moreover, if the above-mentioned condition is satisfied, let $Z = \left[\begin{array}{ccc} z_1 & \cdots & z_n \end{array}\right]$ and let $W = \left[\begin{array}{ccc} w_1 & \cdots & w_n \end{array}\right]$. Then, $X = ZW^{-1}$ is a solution for $Y = \mathcal{R}(X)$.*

Proof:
If X is a solution for $Y = \mathcal{R}(X)$, let $\mathcal{M}$ be the subspace defined by all the vectors of the form $\left[\begin{array}{c} X \\ I_n \end{array}\right] \cdot \xi$, where $\xi \in \mathbb{F}^{n \times 1}$. Then, obviously $\mathcal{M}$ is n-dimensional subspace of $\mathbb{F}^{(m+n)\times 1}$ and since:

$$\begin{aligned} T \left[\begin{array}{c} X \\ I_n \end{array}\right] \cdot \xi &= \left[\begin{array}{c} AX + B + Y \\ CX + D \end{array}\right] \cdot \xi \\ &= \left[\begin{array}{c} X \\ I_n \end{array}\right] \cdot (CX + D) \cdot \xi \in \mathcal{M}, \end{aligned}$$

it follows that $\mathcal{M}$ is T-invariant subspace.
Conversely, if there exist n-dimensional T-invariant subspace $\mathcal{M} \subseteq \mathbb{F}^{(m+n)\times 1}$, such that if $v_1, \ldots, v_n$ is a basis for $\mathcal{M}$ and each $v_j, j = 1, \ldots, n$ is partitioned as $v_j = \left[\begin{array}{c} z_j \\ w_j \end{array}\right]$, where w_j has size $n \times 1$, then $w_1, \ldots, w_n$ are independent then, there exist $n \times n$ matrix L such that $T \left[\begin{array}{c} Z \\ W \end{array}\right] = \left[\begin{array}{c} Z \\ W \end{array}\right] \cdot L$. Therefore, $T \left[\begin{array}{c} ZW^{-1} \\ I_n \end{array}\right] = \left[\begin{array}{c} ZW^{-1} \\ I_n \end{array}\right] WLW^{-1}$. Let $X = ZW^{-1}$. Then, $T \left[\begin{array}{c} X \\ I_n \end{array}\right] = \left[\begin{array}{c} X \\ I_n \end{array}\right] WLW^{-1}$ and comparing the blocks we get:

$$\begin{cases} AX + B + Y = XWLW^{-1} \\ CX + D = WLW^{-1}. \end{cases} \tag{9}$$

Substitution of $WLW^{-1} = CX + D$ into the first equation of (9) yields $AX + B + Y = X(CX + D)$ which is just $Y = \mathcal{R}(X)$. ■

The $\otimes$ denotes the Kronecker product which is defined as follows: if A and B are $m \times n$ and $s \times t$ matrices resp., then the matrix $A \otimes B$ is a $ms \times nt$ matrix, with (i, j)'th block defined by $a_{i,j}B$, for $1 \leq i \leq m$ and $1 \leq j \leq n$. We have the following lemma:

Lemma 1.2 *Let A be a $n \times n$ matrix over $\mathbb{F}_q$ with $rank\,(A) = r$. Then, the rank of the matrix $I \otimes A + A^T \otimes I$ does not exceed $2nr$.*

Proof:
It is a known fact that $rank\,(A \otimes B) = rank\,(A)\,rank\,(B)$. Therefore,
$rank\,(I \otimes A) = nr = rank\,\left(A^T \otimes I\right)$, from which we conclude that
$rank\,\left(I \otimes A + A^T \otimes I\right) \leq 2nr$. ■

Lemma 1.3 *Let G, H be square matrices with sizes $m \times m$ and $n \times n$, respectively. Then, the matrix $G \otimes H$ is invertible if and only if G, H are invertible with inverse given by $G^{-1} \otimes H^{-1}$.*

The suggested encryption scheme and digital signature are based on the following theory regarding a characterization of solutions to special algebraic Riccati equations and the invertibility of the Riccati operator.

The key theorem:

Theorem 1.5 *Let C be a given $n \times m$ matrix for which C^+ exists over $\mathbb{F}$. Let*

$$B = C^+LCC^+L = C^+\left(CC^+L\right)^2$$

where L is $n \times n$ matrix. Then, $XCX = B$ if, and only if $X = X_{M,U} := C^+M + U$, where M, U satisfy:

$$CC^+M = M, M^2 = \left(CC^+L\right)^2, UM = 0, CU = 0. \tag{10}$$

Moreover, for any X as above, there corresponds unique pair M, U satisfying (10).

Note that assuming the existence of M^+, U satisfy $UM = 0$ and $CU = 0$ if and only if:

$$U = \left(I - C^+C\right)V\left(I - MM^+\right),$$

where V is arbitrary. Particulary,

$$U = \left(I - C^+C\right)V\left(I - CC^+\right),$$

satisfy $UM = 0$ and $CU = 0$ for arbitrary V, since $M = CC^+M$.

The key lemma:

Lemma 1.4 *Let* $L = \begin{bmatrix} I_{n_1} & L_{1,2} & L_{1,3} \\ 0 & -I_{n_2} & L_{2,3} \\ 0 & 0 & L_{3,3} \end{bmatrix}$*, where* $L_{3,3}$ *is* $n_3 \times n_3$ *and let* C *be such that* C^+ *exists and* $CC^+ = \begin{bmatrix} I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & 0 & 0_{n_3} \end{bmatrix}$*. Let* $M = \begin{bmatrix} I_{n_1} & M_{1,2} & M_{1,3} \\ 0 & -I_{n_2} & L_{2,3} \\ 0 & 0 & 0_{n_3} \end{bmatrix}$ *where* $M_{1,2}$ *is arbitrary and* $M_{1,3} = L_{1,3} + L_{1,2}L_{2,3} - M_{1,2}L_{2,3}$*. Then,* $CC^+M = M$ *and* $M^2 = (CC^+L)^2$*.*

Note that the $M_{1,2}$ block is free and will be served to hide information. In view of Theorem 1.5 and Lemma 1.4 we conclude that even simple ARE's such as $XCX = B$ can have an exponential number of solutions.

The key observation:
Let X_0 with size $m \times n$ and A, B, C, D with compatible sizes be given. Let L, U_0 be $n \times n, m \times n$ matrices, and let $Z_0 = C^+ (CC^+L)^2$. Let $\mathcal{X}$ denote the manifold of all matrices $X = X_0 + V$ such that $VCV = Z_0$. Let $\mathcal{H}$ be the manifold of all $X = X_0 + V$ such that

$$V = C^+M + \left(I - C^+C\right) U_0 \left(I - CC^+\right),$$

where M is arbitrary matrix satisfying $CC^+M = M, M^2 = (CC^+L_0)^2$. Theorem 1.5 implies that $VCV = Z_0$ and therefore $\mathcal{H} \subseteq \mathcal{X}$. Now,

$$\begin{aligned} Y = \mathcal{R}(X) &= (X_0 + V) C (X_0 + V) + (X_0 + V) D - A (X_0 + V) - B \\ &= \mathcal{R}(X_0) + VCV + V (D + CX_0) - (A - X_0C) V \\ &= \mathcal{R}(X_0) + Z_0 + \Delta (X_0, V), \end{aligned} \tag{11}$$

where:

$$\Delta (X_0, V) := V (D + CX_0) - (A - X_0C) V.$$

For encryption purposes, we will demand that Δ would be invertible, to whom X_0 is known. In this case, the invertibility of Δ is equivalent to the invertibility of the matrix:

$$\Gamma := (D + CX_0)^T \otimes I - I \otimes (A - X_0C).$$

Under this condition, the operator $\mathcal{R}$ is invertible (as an operator restricted to operate on the manifold $\mathcal{H}$), with inverse given by:

$$X = X_0 + mat\left(\Gamma^{-1} vec\left(Y - \mathcal{R}(X_0) - Z_0\right)\right).$$

Remark 1.1 *Generally,* n *will denote the number of variables in the multivariable quadratic equations and* m *will denote the number of such equations arising from the algebraic Riccati equation* $Y = \mathcal{R}(X)$*. Occasionally,* m *or* n *will denote other elements, depending on the context.*

1.2 The Public-Key Encryption System $SRTPI\left(\mathbb{F}_q, n_1, n_2, n_3\right)$

In the following we describe the system $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$. For the 128-bit, 196-bit and 256-bit security we take $t = 12, 14$ and $t = 16$, respectively. This is according to the attack appearing in [1] (see Table 12).

We now describe the system of Bob. Let $n = 4t$ and $m = 3t$. Bob chooses $n \times n$ random matrices A, B, C, D such that: $C = \left[\begin{array}{c} C_1 \\ 0 \end{array}\right]$, where C_1 is $m \times n$ matrix such that C_1^+ exists and $CC^+ = \left[\begin{array}{cc} C_1C_1^+ & 0 \\ 0 & 0 \end{array}\right] = \left[\begin{array}{lll} I_t & 0 & 0 \\ 0 & I_{2t} & 0 \\ 0 & 0 & 0_t \end{array}\right]$. Note that C_1^+ exists and $C_1C_1^+ = I_m$ if and only if $m = rank\left(C_1\right) = rank\left(C_1C_1^T\right) = rank\left(C_1^TC_1\right)$. Next, Bob chooses an $n \times n$ matrix X_0 such that:

$$\Gamma = \left(D + CX_0\right)^T \otimes I_n - I_n \otimes \left(A - X_0C\right),$$

is invertible. Bob chooses random matrices $L_{1,2}, L_{1,3}, L_{2,3}$ with sizes $t \times 2t, t \times t, 2t \times t$ and constructs:

$$L = \left[\begin{array}{lll} I_t & L_{1,2} & L_{1,3} \\ 0 & -I_{2t} & L_{2,3} \\ 0 & 0 & 0_t \end{array}\right].$$

Let $m_1 \cdots m_{t^2}$ denote the t^2-bit parameters for the message to be encrypted by Alice. Let $r_1 \cdots r_{t^2}$ denote parameters for t^2 random bits, to be chosen by Alice. Let $M_{1,2}$ denote a $t \times 2t$ matrix containing the parameters $m_j, j = 1, \ldots, t^2$ and the parameters $r_j, j = 1, \ldots, t^2$, where these $2t^2$ parameters are arranged in a random manner, which will be part of the secrecy of Bob's system. This can be done as follows: Let $1, \ldots, 2t^2$ denote the order of the entries of $M_{1,2}$ from left-to-right and from top-to-bottom. Let $\pi : \left\{1, \ldots, 2t^2\right\} \rightarrow \left\{1, \ldots, 2t^2\right\}$ be a randomly chosen permutation. Then, $m_j, j = 1, \ldots, t^2$ will be placed in the places $\pi\left(1\right), \ldots, \pi\left(t^2\right)$ of $M_{1,2}$ and $r_j, j = 1, \ldots, t^2$ will be placed in the places $\pi\left(t^2 + 1\right), \ldots, \pi\left(2t^2\right)$ of $M_{1,2}$. Let

$$M = \left[\begin{array}{lll} I_t & M_{1,2} & M_{1,3} \\ 0 & -I_{2t} & L_{2,3} \\ 0 & 0 & 0_t \end{array}\right], \tag{12}$$

where:

$$M_{1,3} = -M_{1,2}L_{2,3} + L_{1,2}L_{2,3} + L_{1,3}.$$

Note that $M^2 = \left(CC^+L\right)^2 = L^2$ and $CC^+M = M$ for any choice of $m_1 \cdots m_{t^2}$ and $r_1 \cdots r_{t^2}$. Let $Z_0 = C^+\left(CC^+L\right)^2$ and let U_0 a $n \times n$ randomly chosen matrix. Let:

$$V = C^+M + \left(I_n - C^+C\right)U_0\left(I_n - CC^+\right),$$

and note that $VCV = Z_0$ for any choice of $m_1 \cdots m_{t^2}$ and $r_1 \cdots r_{t^2}$. Let $X = X_0 + V$. Now, Bob chooses random invertible $n \times n$ matrices P_L, P_R. Bob computes:

$$\widetilde{A} = P_L A P_L^{-1}, \widetilde{B} = P_L B P_R^{-1}, \widetilde{C} = P_R C P_L^{-1}, \widetilde{D} = P_R D P_R^{-1}, \widetilde{X} = P_L X P_R^{-1}.$$

The parametric matrix $\widetilde{X}$ (which depends on $m_1 \cdots m_{t^2}$ and $r_1 \cdots r_{t^2}$) will be called the ”black box”.

Since:

$$\widetilde{X} = P_L X P_R^{-1} = P_L \left(X_0 + C^+ M + \widehat{U_0} \right) P_R^{-1},$$

where $\widehat{U_0} := (I_n - C^+ C) \, U_0 \, (I_n - CC^+)$, it follows that:

$$vec\left(\widetilde{X}\right) = \left(P_R^{-T} \otimes \; P_L\right) \cdot \; vec\left(X_0 + C^+ M + \widehat{U_0}\right) = \widetilde{Q} \begin{bmatrix} m_1 \\ \vdots \\ m_{t^2} \\ r_1 \\ \vdots \\ r_{t^2} \\ 1 \end{bmatrix},$$

where $\widetilde{Q}$ is a $n^2 \times \left(2t^2 + 1\right)$ matrix. The matrix $\widetilde{Q} = \left[\begin{array}{ccc} \widetilde{q_1} & \cdots & \widetilde{q_{2t^2+1}} \end{array} \right]$ is calculated as follows: For $1 \leq j \leq t^2$ let M_j denote the matrix M where $m_j = 1$ and all other free parameters are set to 0. For $t^2 + 1 \leq j \leq 2t^2$ let M_j denote the matrix M where $r_{j-t^2} = 1$ and all other free parameters are set to 0. Finally, for $j = 2t^2 + 1$ let M_j denote the matrix M where all the free parameters are set to 0. Thus,

$$\widetilde{q_{2t^2+1}} = \left(P_R^{-T} \otimes \; P_L\right) \cdot \; vec\left(X_0 + C^+ M_{2t^2+1} + \widehat{U_0}\right)$$

and

$$\begin{aligned} \widetilde{q_j} &= \left(P_R^{-T} \otimes \; P_L\right) \cdot \; vec\left(X_0 + C^+ M_j + \widehat{U_0}\right) - \widetilde{q_{2t^2+1}} \\ &= \left(P_R^{-T} \otimes \; P_L\right) \cdot \; vec\left(X_0 + C^+ \left(M_j - M_{2t^2+1}\right) + \widehat{U_0}\right) \end{aligned}$$

for $j = 1, \ldots, 2t^2$.

The Actual Public-Key Of Bob:

$$\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}, \widetilde{Q}.$$

The Secrete-Key of Bob:

$$A, B, C, D, L, \mathcal{R}\left(X_0\right), Z_0, P_L^{-1}, P_R, \Gamma^{-1}, \pi.$$

Alice Sends an encrypted message to Bob:
When Alice wats to send a message $m_1 \cdots m_{t^2}$ to Bob, she chooses random bits $r_1 \cdots r_{t^2}$, computes:

$$\begin{cases} \widetilde{X} = mat\left(\widetilde{Q}\begin{bmatrix} m_1 \\ \vdots \\ m_{t^2} \\ r_1 \\ \vdots \\ r_{t^2} \\ 1 \end{bmatrix}\right) \\ \widetilde{Y} = \widetilde{R}\left(\widetilde{X}\right) := \widetilde{X}\widetilde{C}\widetilde{X} + \widetilde{X}\widetilde{D} - \widetilde{A}\widetilde{X} - \widetilde{B}, \end{cases}$$

and sends $\widetilde{Y}$ to Bob.

Bob decrypts the message:
Bob computes:

$$Y = P_L^{-1}\widetilde{Y}P_R.$$

Note that $Y = R(X) := XCX + XD - AX - B$. Next, Bob computes:

$$W := Y - R(X_0) - Z_0,$$

and note that $W = Y - R(X_0) - VCV = V(D + CX_0) - (A - X_0C)V$. Bob recovers V as:

$$V = mat\left(\Gamma^{-1}vec(W)\right), \tag{13}$$

from which the specific matrix M is recovered as:

$$M = CV,$$

since $CC^{+}M = M$. Finally, from:

$$M = \begin{bmatrix} I_t & M_{1,2} & M_{1,3} \\ 0 & -I_{2t} & L_{2,3} \\ 0 & 0 & 0_t \end{bmatrix},$$

Bob recovers the block $M_{1,2}$, from which he reads the message $m_1 \cdots m_{t^2}$ from the places $\pi(1), \ldots, \pi\left(t^2\right)$.

The related algorithms for the secrete-key generation, public-key generation, encryption and decryption are given in Algorithm 1-Algorithm 5.

Remark 1.2 *Note that the the use of the random bits $r_1, \ldots, r_{t^2}$ makes the encryption algorithm a relation and not a function because the same message $m_1, \ldots, m_{t^2}$ will be encrypted (with high probability) to different values of $\widetilde{Y}$, due to the use of the random bits. It is our belief that in this way the system is more secure to known plain-text attacks. Note that the decryption of $m_1, \ldots, m_{t^2}$ from $\widetilde{Y}$ is deterministic and unique.*

The rational of choosing $n_1 = n_3 = t, n_2 = 2t$ is the following: If $n_3 = 0$ then, $CC^+ = I$ and $L = \begin{bmatrix} I_{n_1} & L_{1,2} \\ 0 & -I_{n_2} \end{bmatrix}$ would imply that $L^2 = I_n$. We therefore would have $Z_0 = C^+ \left(CC^+L\right)^2 = C^+ = C^{-1}$ and thus $\widetilde{Z_0} = \widetilde{C}^{-1}$, where $\widetilde{C}$ is part of the public-key. Therefore, n_3 has to be non-zero. Moreover, from the security point of view, we need that n_3 would be proportional to n_1, say $n_1 = n_3 = t$. Let $n_2 = \alpha t$, with $\alpha \geq 1$. Now, from the efficiency point of view, we need to maximize the portion of $M_{1,2}$ in M. Therefore, we need to maximize $\frac{\alpha t^2}{(2+\alpha)^2 t^2} = \frac{\alpha}{(2+\alpha)^2}$ over the range $\alpha \geq 1$. The last is maximized when $\alpha = 2$.

In a more general view, by fixing $\beta \geq 1$ and letting $n_1 = \beta t, n_2 = \alpha t, n_3 = t$, the portion $\frac{\alpha\beta}{(\alpha+\beta+1)^2}$ is maximized over the range $\alpha \geq 1$ by $\alpha = \beta + 1$. Letting $\beta \to +\infty$ would result with $\frac{1}{4}$. We therefore cannot have more than 25% usage of the matrix space. On the other hand, for the 2^{256}-bit security, we need 487 variables (see Remark 5.4 below). Now, the closest integer to 487 in the form $n_1 n_2 = \beta\left(\beta+1\right) t^2$ is given by $\beta = 4, t = 5$, resulting with a portion of $\frac{21}{88} \approx 0.2386$, with $\beta\left(\beta+1\right) t^2 = 500$ and 13 spare variables, with matrices of sizes 50×50. We choose $\beta = 1, t = 16$, resulting with a portion of $\frac{1}{8} = 0.125$, $\beta\left(\beta+1\right) t^2 = 512$ with 25 spare variables and matrices of sizes 64×64, which is not so far from the optimal value but is more convenient with respect to calculations over $\mathbb{F}_2$ that can be made with bytes. We therefore made a trade-off between memory-space and run-time. This explains the above-mentioned choice of the parameters. Note that possibly, more can be done in the matrix usage, if one uses a non-square matrices and maybe other forms of the matrices M and L (i.e. other than those given in Lemma 1.4).

Computing The Secret Key Parameters:

The secret key of Bob contains the matrices P_L, P_R, Γ which are invertible matrices. Constructing the invertible matrices P_L, P_R over $\mathbb{F}_2$ can be done efficiently by choosing lower-triangular matrix H_L with 1's on the main diagonal where the other elements are chosen randomly and by choosing upper-triangular matrix G_L with 1's on the main diagonal where the other elements are chosen randomly and defining $P_L = H_L G_L$. The construction of P_R is similar. Note that there is no loss of generality in this construction since that any invertible matrix over $\mathbb{F}_2$ can be factorized as LVU where L is lower-triangular, U is upper-triangular and V is diagonal - all of which are invertible. Therefore, over $\mathbb{F}_2$, we have $V = I$ and thus there is no loss of generality in the above mentioned construction. The construction of P_L, P_R over any field $\mathbb{F}$ is similar. In this case the main-diagonal elements of H, G are chosen randomly from $\mathbb{F}^* := \mathbb{F} \setminus \{0\}$.

In order to construct:

$$\Gamma = \left(D + CX_0\right)^T \otimes I_n - I_n \otimes \left(A - X_0 C\right)$$

to be invertible (using deterministic method - see Remark 1.4 below), as a first attempt, we could think of the following method: let $\widehat{D} = \left[\begin{array}{cc} \widehat{D_{1,1}} & \widehat{D_{1,2}} \\ 0 & 0 \end{array} \right]$, where $\widehat{D_{1,1}}, \widehat{D_{1,2}}$ are chosen randomly, where $\widehat{D_{1,1}}$ is $m \times m$. Note that $CC^{+}\widehat{D} = \widehat{D}$, since $CC^{+} = \left[\begin{array}{cc} I_m & 0 \\ 0 & 0 \end{array} \right]$. Let $\widetilde{D}$ be randomly chosen invertible matrix and define $D := \widetilde{D} - \widehat{D}$. Then, according to Lemma 1.1, the equation $D + CX_0 = \widetilde{D}$, which is equivalent to $CX_0 = \widehat{D}$ is solvable since $CC^{+}\widehat{D} = \widehat{D}$. Moreover, we can choose $X_0 = C^{+}\widehat{D} + (I_n - C^{+}C)\, Y_0$ where Y_0 is chosen randomly. Next, we define $A := X_0 C$. With these definitions we get $\Gamma = \widetilde{D}^T \otimes I_n$ which is invertible according to Lemma 1.3, with $\Gamma^{-1} = \widetilde{D}^{-T} \otimes I_n$. In this case, the matrix V in (13) is computed as:

$$V = mat\left(\Gamma^{-1} vec\,(W)\right) = W\widetilde{D}^{-1}. \tag{14}$$

Remark 1.3 *Note that this method might be dangerous since that $A = X_0 C$ is equivalent to $\widetilde{A} = \widetilde{X_0}\widetilde{C}$, where $\widetilde{A}, \widetilde{C}$ are part of the public-key. Since $\widetilde{A} = \widetilde{X_0}\widetilde{C}$ has a solution as an equation with variable $\widetilde{X_0}$, Lemma 1.2 implies that $\widetilde{X_0} = \widetilde{A}\widetilde{C}^{+} + \widetilde{W_0}\left(I - \widetilde{C}\widetilde{C}^{+}\right)$, where $\widetilde{W_0}$ is unknown. This might reduce the security of the system, since $\widetilde{W_0}\left(I - \widetilde{C}\widetilde{C}^{+}\right)$ contains only $4t^2$ unknown free variables, where $\widetilde{X_0}$ contains $16t^2$ variables.*

We have the following lemma:

Lemma 1.5 *Let $D = \left[\begin{array}{cc} D_{1,1} & D_{1,2} \\ D_{2,1} & D_{2,2} \end{array} \right]$ be chosen randomly but such that $D_{2,2}$ is strictly-lower-triangular (i.e. lower-triangular with 0 main-diagonal). Let $G = \left[\begin{array}{cc} G_{1,1} & 0 \\ D_{2,1} & D_{2,2} \end{array} \right]$, where $G_{1,1}$ is chosen randomly but such that it is strictly-lower-triangular. Let H be randomly chosen invertible matrix and let $A = H + C^{+}(G - D)\, C$. Let $X_0 = C^{+}(G - D) + (I - C^{+}C)\, Y_0 \,(I - CC^{+})$. Then, Γ is invertible with inverse given by:*

$$\Gamma^{-1} = -\left(I \otimes H^{-1}\right) \cdot \sum_{k=0}^{n-1} \left(G^T \otimes H^{-1}\right)^k = -\sum_{k=0}^{n-1} \left(\left(G^T\right)^k \otimes \left(H^{-1}\right)^{k+1}\right). \tag{15}$$

Proof:
The definitions of D, G and X_0 implies that $CC^{+}(G - D) = G - D$, from which we have $CX_0 = CC^{+}(G - D) = G - D$, implying that $D + CX_0 = G$. We also have $X_0 C = C^{+}(G - D)\, C = A - H$, implying that $A - X_0 C = H$. Now, the definition of the Kronecker product implies that:

$$\begin{aligned} I_n \otimes (A - X_0 C) &= \text{block-diagonal}\,(A - X_0 C, A - X_0 C, \ldots, A - X_0 C) \\ &= \text{block-diagonal}\,(H, H, \ldots, H)\,, \end{aligned}$$

and that:

$$(D + CX_0)^T \otimes I_n = \left[(D + CX_0)_{j,i} \cdot I_n\right] = \left[(G)_{j,i} \cdot I_n\right].$$

Now, Since G is strictly-lower-triangular, G^T is strictly-upper-triangular. Therefore, Γ is block-upper-triangular with the block H on its main-block-diagonal (Γ has $n \times n$ blocks of $n \times n$ matrices). It follows that Γ is invertible.
In order to prove (15), note first that $\Gamma = G^T \otimes I - I \otimes H$ and that $G^T \otimes H^{-1}$ is strictly-block-upper-triangular with $n \times n$ blocks of $n \times n$ matrices. Thus, $\left(G^T \otimes H^{-1}\right)^n = 0$. Let:

$$\Omega := -\left(I \otimes H^{-1}\right) \cdot \sum_{k=0}^{n-1} \left(G^T \otimes H^{-1}\right)^k .$$

Then, using Lemma 1.3, we get:

$$\begin{aligned}
\Gamma \cdot \Omega &= +\left(I \otimes H\right) \cdot \left(I \otimes H^{-1}\right) \cdot \sum_{k=0}^{n-1} \left(G^T \otimes H^{-1}\right)^k + \\
&\quad - \left(G^T \otimes I\right) \cdot \left(I \otimes H^{-1}\right) \cdot \sum_{k=0}^{n-1} \left(G^T \otimes H^{-1}\right)^k \\
&= + \left(I \otimes HH^{-1}\right) \cdot \sum_{k=0}^{n-1} \left(G^T \otimes H^{-1}\right)^k + \\
&\quad - \left(G^T \otimes H^{-1}\right) \cdot \sum_{k=0}^{n-1} \left(G^T \otimes H^{-1}\right)^k \\
&= I_{n^2} \cdot \sum_{k=0}^{n-1} \left(G^T \otimes H^{-1}\right)^k - \sum_{k=1}^{n} \left(G^T \otimes H^{-1}\right)^k \\
&= \left(G^T \otimes H^{-1}\right)^0 - \left(G^T \otimes H^{-1}\right)^n = I_{n^2}.
\end{aligned}$$

■

Remark 1.4 *In the non-deterministic way to construct the matrix Γ (i.e. by choosing A, B, C, D, X_0 randomly, where C is in the needed form), the matrix Γ was invertible in 28% of the cases, which is the percent of invertible matrices over $\mathbb{F}_2$.*

1.3 The Digital Signature $TPSig\left(\mathbb{F}_q, n_1, n_2, n_3\right)$

We now describe Alice's signature system $TPSig\left(\mathbb{F}_q, n_1, n_2, n_3\right)$:
The $M_{1,2}$ block of M is $n_1 \times n_2$ and thus contains $n = n_1 n_2$ free variables. We number the entries of $M_{1,2}$ by $1, \ldots, n$ with respect to the natural order, i.e. from let-to-right and from top-to-bottom. Let $\pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ be a random permutation. Let $m_1, \ldots, \ldots, m_n$ denote parameters for hash values. Then, we arrange $m_1, \ldots, m_n$ in the places $\pi(1), \ldots, \pi(n)$ of $M_{1,2}$. We keep denote the new matrix as M. Since:

$$Y = \mathcal{R}\left(X_0\right) + Z_0 + \Delta\left(X_0, V\right),$$

is an affine function of the parameters $m_1, \ldots, m_n$, to whom X_0, Z_0 (and the other secrete-key parameters) are known. Therefore,

$$Y = mat\left(Q\begin{bmatrix} m_1 \\ \vdots \\ m_n \\ 1 \end{bmatrix}\right), \tag{16}$$

where Q is $\left(n_1 + n_2 + n_3\right)^2 \times (n+1)$ matrix. Alice chooses $\left(n_1 + n_2 + n_3\right) \times \left(n_1 + n_2 + n_3\right)$ random matrices A, B, C, D, X_0 such that:

$$C = \begin{bmatrix} C_1 \\ 0 \end{bmatrix}, \tag{17}$$

where C_1 is $\left(n_1 + n_2\right) \times \left(n_1 + n_2 + n_3\right)$ matrix such that C_1^+ exists and $CC^+ = \begin{bmatrix} C_1 C_1^+ & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} I_{n_1} & 0 & 0 \\ 0 & I_{n_2} & 0 \\ 0 & 0 & 0_{n_3} \end{bmatrix}$. Alice chooses random matrices $L_{1,2}, L_{1,3}, L_{2,3}$ with sizes $n_1 \times n_2, n_1 \times n_3, n_2 \times n_3$ and constructs:

$$L = \begin{bmatrix} I_{n_1} & L_{1,2} & L_{1,3} \\ 0 & -I_{n_2} & L_{2,3} \\ 0 & 0 & 0_{n_3} \end{bmatrix}.$$

Let $m_1 \cdots m_n$ denote the parameters for the hash value of the message to be signed by Alice. Let $\pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ be a permutation chosen randomly by Alice.

Let $M_{1,2}$ denote a $n_1 \times n_2$ matrix containing the parameters $m_j, j = 1, \ldots, n$, where these n parameters are placed in the places $\pi(1), \ldots, \pi(n)$ of $M_{1,2}$. Let

$$M = \begin{bmatrix} I_{n_1} & M_{1,2} & M_{1,3} \\ 0 & -I_{n_2} & L_{2,3} \\ 0 & 0 & 0_{n_3} \end{bmatrix}, \tag{18}$$

where:

$$M_{1,3} = -M_{1,2} L_{2,3} + L_{1,2} L_{2,3} + L_{1,3}.$$

Let $Z_0 = C^+ \left(CC^+L\right)^2$ and let U_0 be a $(n_1+n_2+n_3) \times (n_1+n_2+n_3)$ randomly chosen matrix. Let:

$$V = C^+M + \left(I - C^+C\right) U_0 \left(I - CC^+\right) = C^+M + \widehat{U_0},$$

and note that $VCV = Z_0$ for any choice of $m_1 \cdots m_n$. Let $X = X_0 + V$. Let $h : \mathbb{F}_q^{(\mathbb{N})} \to \mathbb{F}_q^n$ denote a secure public hash function (where: $\mathbb{F}_q^{(\mathbb{N})} = \cup_{\ell=0}^{\infty} \mathbb{F}_q^{\ell}$ is the set of all finite sequences of elements from $\mathbb{F}_q$).

The secrete-key of Alice:

$$\left(\pi, L, M, X_0, \widehat{U_0}\right),$$

The public-key of Alice:

$$(A, B, C, D, Q).$$

Signing a message:

When Alice wats to sign a message $m \in \mathbb{F}_q^{(\mathbb{N})}$ she computes $\langle m_1, \ldots, m_n \rangle = h(m)$. Next, Alice computes $X_m = X_0 + V_m$, where $V_m = C^+M_m + \widehat{U_0}$ and sends $\langle m, X_m \rangle$ to Bob - the verifier.

Verifying a signature:

When Bob the verifier wants to verify Alice's signature on m, he computes: $\langle m_1, \ldots, m_n \rangle = h(m)$ and

$$V_1 := Y_m = mat\left(Q \begin{bmatrix} m_1 \\ \vdots \\ m_n \\ 1 \end{bmatrix}\right). \tag{19}$$

Next, Bob computes:

$$V_2 := \mathcal{R}(X_m) = X_mCX_m + X_mD - AX_m - B.$$

Bob accepts the signature as a valid signature of Alice only if $V_2 = V_1$.

The secrete-key generation, the public-key generation, the signing and verifying algorithms are given in Algorithm 6-Algorithm 9.

Remark 1.5 *Note that a possible forger would choose $\hat{m} \in \mathbb{F}_q^{(\mathbb{N})}$ and compute $Y_{\hat{m}}$ as in (19). Next, he will try to find $X_{\hat{m}}$ such that $Y_{\hat{m}} = \mathcal{R}(X_{\hat{m}})$, which is assumed to be hard. In another scenario, the forger can choose any X and compute $Y = \mathcal{R}(X)$. Next, he will compute $m_1, \ldots, m_n$ from the linear equations (16). Finally, he will try to find $m \in \mathbb{F}_q^{(\mathbb{N})}$ such that $h(m) = \langle m_1, \ldots, m_n \rangle$, which is the problem of finding a pre-image of a secure hash function and is assumed to be hard.*

2 A detailed performance analysis

The systems $TPSig\,(\mathbb{F}_2, 12, 24, 12)$ and $TPSig\,(\mathbb{F}_2, 16, 32, 16)$ and $SRTPI\,(\mathbb{F}_2, 16, 32, 16)$ were implemented as a ANSI C code on Intel® Core I-5, 5200U, 2.7GHz processor with operating system Windows®-8. The time performances of $TPSig\,(\mathbb{F}_2, 12, 24, 12)$ and $TPSig\,(\mathbb{F}_2, 16, 32, 16)$ are given in Table 1 and Table 3, resp. The space performances of $TPSig\,(\mathbb{F}_2, 12, 24, 12)$ and $TPSig\,(\mathbb{F}_2, 16, 32, 16)$ are given in Table 2 and Table 4, resp. The time for generating public and secrete key's for $TPSig\,(\mathbb{F}_2, 12, 24, 12)$ is 37 $[ms]$. The time and space performance of the $SRTPI\,(\mathbb{F}_2, 16, 32, 16)$ appear in Table 5 and Table 6, resp.

Scheme	**Sig.** [**ms**]	**Sig. Val.** [**ms**]	**Total** $[ms]$	**Remarks**
$UOV\,(44, 59)$	4.9312	5.2349	10.1661	(2.2)
0/1 $UOV\,(44, 59)$	4.7339	5.0256	9.7595	(2.2)
Rainbow $(36, 21, 22)$	3.0472	3.4134	6.4606	(2.2)
$enTTS\,(15, 60, 88)$	0.7933	11.4034	12.1967	(2.2)
$RGB\,(\mathbb{F}_{\mathbb{F}_{2^8}}, 40, 35, 20)$	0.4898	0.6059	1.0957	(2.1)
$TPSig\,(\mathbb{F}_2, 12, 24, 12)$	0.32	0.514	0.834	(5.4)

Table 1: Time performance comparison for 2^{128} security level on Intel® Core I-5, 5200U, 2.7GHz, sequential implementation

Scheme	**PK size** [**kB**]	**SK size** [**kB**]	**Total** [**kB**]
$UOV\,(44, 59)$	235.664	194.700	430.364
0/1 $UOV\,(44, 59)$	43.560	116.820	160.380
Rainbow $(36, 21, 22)$	135.880	97.675	233.555
$enTTS\,(15, 60, 88)$	234.960	13.051	248.011
$RGB\,(\mathbb{F}_{2^8}, 40, 35, 20)$	162.960	146.855	309.815
$TPSig\,(\mathbb{F}_2, 12, 24, 12)$	84.3	0.95	85.250

Table 2: Space performance comparison for 2^{128} security level

Remark 2.1 *According to the Key Recovery Attack (KRA) suggested in [29], the security level of RGB* $(\mathbb{F}_{2^8}, 40, 35, 20)$ *is not more than* 2^{53}*!*

Remark 2.2 *The results of [12] were scaled to the processor we used.*

Scheme	**PK +SK Gen. [ms]**	**Sig. [ms]**	**Sig. Val. [ms]**	**Sig. +Sig. Val. [ms]**
$NTRU$	26	355	1.5	356.5
$TPSig\,(\mathbb{F}_2, 16, 32, 16)$	112	0.455	0.8	1.255

Table 3: Time performance comparison for 2^{256} security level on Intel® Core I-5, 5200U, 2.7GHz, sequential implementation

Scheme	**PK size [kB]**	**SK size [kB]**	**Sig. size [kB]**	**PK +SK [kB]**
$NTRU$	1.927	0.144	1.814	2.071
$TPSig\,(\mathbb{F}_2, 16, 32, 16)$	260	1.65	0.5	261.65

Table 4: Space performance comparison for 2^{256} security level

Public − Key +Secrete − Key Generation [ms]	**Encryption [ms]**	**Decryption [ms]**
4090	1.5	8.37

Table 5: Time performance of $SRTPI\,(\mathbb{F}_2, 16, 32, 16)$ with 2^{256} security level on Intel® Core I-5, 5200U, 2.7GHz, sequential implementation

PK [kB]	**SK [kB]**	**Plain − Text [B]**	**Random − Bits [B]**	**Encrypted Message [kB]**
262	2052.5	32	32	0.5

Table 6: Space performance of $SRTPI\,(\mathbb{F}_2, 16, 32, 16)$ with 2^{256} security level

3 Known answer test values

3.1 Test values for $SRTPI\,(\mathbb{F}_2, 16, 32, 16)$

The Known Answer Test (KAT) values for $SRTPI\,(\mathbb{F}_2, 16, 32, 16)$ are given in a separate files attached to this submission, because of their length.

3.2 Test values for $TPSig\,(\mathbb{F}_2, 16, 32, 16)$

The Known Answer Test (KAT) values for $TPSig\,(\mathbb{F}_2, 16, 32, 16)$ are given in a separate files attached to this submission, because of their length.

4 A thorough description of the expected security strength

4.1 Sematically Secure Encryption, Adaptive Chosen Ciphetext Attack

Let us consider a particular system in the $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$ scheme. In view of the public-key, we can see that the entries of $\widetilde{X}$ depend on $m_1, \ldots, m_{t^2}, r_1, \ldots, r_{t^2}$, say: $\widetilde{x}_{i,j} = \sum_{k=1}^{t^2} \alpha_{i,j}^{(k)} m_k + \sum_{k=1}^{t^2} \beta_{i,j}^{(k)} r_k + \gamma_{i,j}$, with known coefficients $\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_{i,j}$. Therefore, the set of all possible $4t \times 4t$ matrices $\widetilde{X}$ define a $2t^2$-dimensional affine subspace of $\mathbb{F}_2^{4t\times 4t}$. Let $\widetilde{\mathcal{X}}$ denote this affine subspace.

The entries of $\widetilde{Y} = \widetilde{\mathcal{R}}\left(\widetilde{X}\right)$ are also depend on $m_1, \ldots, m_{t^2}, r_1, \ldots, r_{t^2}$ but in a quadratic dependence, say: $\widetilde{y}_{i,j} = \sum_{1\leq k<\ell\leq t^2} \zeta_{i,j}^{(k,\ell)} m_k m_\ell + \sum_{1\leq k<\ell\leq t^2} \eta_{i,j}^{(k,\ell)} r_k r_\ell + \sum_{1\leq k,\ell\leq t^2} \theta_{i,j}^{(k,\ell)} m_k r_\ell + \vartheta_{i,j}$, with known coefficients $\zeta_{i,j}^{(k,\ell)}, \eta_{i,j}^{(k,\ell)}, \theta_{i,j}^{(k,\ell)}, \vartheta_{i,j}$. Let $\widetilde{\mathcal{Y}}$ denote the set of all possible $\widetilde{Y}$, i.e. the set of all $\widetilde{Y} = \widetilde{\mathcal{R}}\left(\widetilde{X}\right)$ for $\widetilde{X} \in \widetilde{\mathcal{X}}$. The set $\widetilde{\mathcal{Y}}$ is a $2t^2$-dimensional quadratic manifold of $\mathbb{F}_2^{4t\times 4t}$.

Let $\widetilde{R} : \widetilde{\mathcal{X}} \to \widetilde{\mathcal{Y}}$ denote the bijection defined by the system and let $\widetilde{R}^{-1}$ denote its inverse. Let $\mathbf{m}$ and $\mathbf{r}$ denote $m_1, \ldots, m_{t^2}$ and $r_1, \ldots, r_{t^2}$ respectively. Then, $\widetilde{X}$ and $\widetilde{Y}$ are functions of $\mathbf{m}$ and $\mathbf{r}$ and accordingly, we write $\widetilde{X}\left(\mathbf{m}, \mathbf{r}\right)$ and $\widetilde{Y}\left(\mathbf{m}, \mathbf{r}\right)$. We therefore have $\widetilde{R}\left(\widetilde{X}\left(\mathbf{m}, \mathbf{r}\right)\right) = \widetilde{Y}\left(\mathbf{m}, \mathbf{r}\right)$ and $\widetilde{R}^{-1}\left(\widetilde{Y}\left(\mathbf{m}, \mathbf{r}\right)\right) = \widetilde{X}\left(\mathbf{m}, \mathbf{r}\right)$. Note that $\widetilde{X} : \mathbb{F}_2^{2t^2} \to \widetilde{\mathcal{X}}$ is invertible function since $\widetilde{X} = P_L X P_R^{-1}$, where $X = X_0 + V$ and $V = C^+ M + \widehat{U_0}$. Now, $M = CV$ and $m_1, \ldots, m_{t^2}, r_1, \ldots, r_{t^2}$ can be extracted from the $M_{1,2}$ block of M, using the permutation π. Note that we can use $m_1, \ldots, m_{t^2+k}$ for "message bits" and $r_1, \ldots, r_{t^2-k}$ as "random bits", for some $1 \leq k \leq t^2$.

We next introduce the application of the Optimal Asymmetric Encryption Protocol (OAEP) defined by [3], to the $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$ scheme. Let $G : \mathbb{F}_2^{t^2-k} \to \mathbb{F}_2^{t^2+k}$ and $H : \mathbb{F}_2^{t^2+k} \to \mathbb{F}_2^{t^2-k}$ denote two secure hash functions. Let $\mathbf{r} \in \mathbb{F}_2^{t^2-k}$ be chosen randomly. Let $\mathbf{u} = \left(\mathbf{m}\|0^k\right) + G\left(\mathbf{r}\right)$ and let $\mathbf{v} = \mathbf{r} + H\left(\mathbf{u}\right)$. Then, we define the encryption of $\mathbf{u}, \mathbf{v}$ to be $\widetilde{Y}\left(\mathbf{u}, \mathbf{v}\right) = \widetilde{R}\left(\widetilde{X}\left(\mathbf{u}, \mathbf{v}\right)\right)$.

The decryption is made as follows: we compute $\widetilde{X}\left(\mathbf{u}, \mathbf{v}\right) = \widetilde{R}^{-1}\left(\widetilde{Y}\left(\mathbf{u}, \mathbf{v}\right)\right)$ and extract $\mathbf{u}, \mathbf{v}$ from $\widetilde{X}$. Next, we compute $H\left(\mathbf{u}\right)$ from which we extract $\mathbf{r} = \mathbf{v} + H\left(\mathbf{u}\right)$. Finally, we compute $G\left(\mathbf{r}\right)$ from which we extract $\mathbf{M} = \mathbf{u} + G\left(\mathbf{r}\right)$. If $\mathbf{M}$ has the block 0^k in its right-hand then, $\mathbf{m}$ equals the t^2 left-hand bits of $\mathbf{M}$. Otherwise, we reject the message. We call this public-key encryption system $SRTPI - OAEP\left(\mathbb{F}_2, t, 2t, t\right)$.

Here, the underlying encryption function is $\widetilde{Y}(\mathbf{u},\mathbf{v}) = \widetilde{R}\left(\widetilde{X}(\mathbf{u},\mathbf{v})\right)$, and we say that the function is (ℓ,τ,ϵ)-set partial-domain one-way function if any adversary $\mathcal{A}$ outputting a set of ℓ elements $\mathbf{u}_1,\ldots,\mathbf{u}_\ell \in \mathbb{F}_2^{t^2+k}$, in time τ, by viewing $\widetilde{Y}(\mathbf{u},\mathbf{v})$, has success probability upper-bounded by ϵ that $\mathbf{u}_j = \mathbf{u}$ for some $1 \leq j \leq \ell$. We say that the function is a set partial-domain one-way function if in polynomial-time τ in t,k, ϵ is a negligible function of t,k.

Assumption 4.1 *We assume that the function $\widetilde{Y}(\mathbf{u},\mathbf{v}) = \widetilde{R}\left(\widetilde{X}(\mathbf{u},\mathbf{v})\right)$ is a set partial-domain one-way function.*

In [18] it was proved that OAEP is IND-CCA2 secure (i.e. against Indistinguishability Adaptive Chosen Ciphertext Attack), in the random oracle model, under the assumption that the underlying encryption function is a set partial-domain one-way function. In [4] it was proved that IND-CCA2 implies IND-CCA1 and IND-CPA, as well as NM-CCA2, NM-CCA1 and NM-CPA.

We conclude that under the reasonable Assumption 4.1 (see section 5.), the $SRTPI - OAEP(\mathbb{F}_2, t, 2t, t)$ scheme is IND-CCA2 secure.

4.2 Ephemeral-Only Encryption, Ephemeral Key Exchange Protocol, Chosen Plaintext Attack

These are currently under investigation regarding the suggested schemes
$SRTPI(\mathbb{F}_2, t, 2t, t)$ and $TPISig(\mathbb{F}_2, t, 2t, t)$.

4.3 Existentially Unforgeable Digital Signature, Adaptive Chosen Message Attack

Note that for systems in the $TPI(\mathbb{F}_2, t, 2t, t)$ scheme, because we have $16t^2$ quadratic polynomial equations and only $2t^2$ variables, it follows that there exists a single solution X_m to the equation $X_mCX_m + X_mD - AX_m - B = Y_m$ (given Y_m), with high probability (that depends on the choice of the public-key matrices A, B, C, D). Therefore, with high probability, the scheme is Strongly Existentially Unforgeable under adaptive Chosen-Message Attack (i.e. is S-EUF-CMA secure, with high probability).

4.4 Security Strength Categories

In the following tables, we suggest our sets of parameters for the systems $SRTPI$ and $TPSig$, with their security strength categories, under Assumption 5.1 and under its negation, for different optimization criteria: i.e. for systems achieving the needed security level with minimal number of variables while maximizing the usage of the M matrix space and for systems achieving the needed security level with matrices of size which is a multiple of 8 (in order to use operations on bytes) and with a small as possible number of variables (all in view of the BooleanSolver attack - see [1] and Remark 5.4).

Category	t	ℓ	**Expexted Security**	**Matrices Size**	**Plain – Text Vars.**	**Rand. Vars.**
AES 128	12	2	2^{143}	48×48	128	160
SHA3-256	12	3	2^{146}	48×48	256	32
AES 192	15	3	2^{207}	60×60	192	258
SHA3-384	15	3	2^{210}	60×60	384	66
AES 256	17	3	2^{272}	68×68	256	322
SHA3-512	17	4	2^{274}	68×68	512	66

Table 7: Suggested sets of parameters for $SRTPI\,(\mathbb{F}_2, n_1, n_2, n_3)$ and $TPISig\,(\mathbb{F}_2, n_1, n_2, n_3)$ where $n_1 = t, n_2 = 2t, n_3 = t$ under Assumption 5.1 with parameter ℓ

Remark 4.1 *In Table 7, in category AES 128, one can use $SRTPI$ to encrypt simultaneously* 2 *blocks of* 128*-bit with* 32 *random bits for the extra variables, in category AES 192, one can encrypt* 2 *blocks of* 192*-bit and use* 66 *random bits and in category AES 256, one can encrypt* 2 *blocks of* 256*-bit and use* 66 *random bits. For the digital signature $TPISig$, all the variables should be plaintext variables and the hash function involved should have output with the related size.*

Category	**Parameters** n_1, n_2, n_3	ℓ	**Expected Security**	**Matrices Size**	**Plain – Text Vars.**	**Rand. Vars.**
AES 128	$12, 24, 12$	2	2^{143}	48×48	128	160
SHA3-256	$12, 24, 12$	3	2^{146}	48×48	256	32
AES 192	$16, 32, 16$	3	2^{207}	64×64	192	320
SHA3-384	$16, 32, 16$	3	2^{210}	64×64	384	128
AES 256	$21, 34, 17$	3	2^{272}	72×72	256	458
SHA3-512	$21, 34, 17$	3	2^{274}	72×72	512	202

Table 8: Suggested sets of parameters for $SRTPI\,(\mathbb{F}_2, n_1, n_2, n_3)$ and $TPISig\,(\mathbb{F}_2, n_1, n_2, n_3)$ under Assumption 5.1 with parameter ℓ, targeted to operate with bytes

Category	**Parameters** n_1, n_2, n_3	**Expected** **Security**	**Matrices** **Size**	**Plain – Text** **Vars.**	**Rand.** **Vars.**
AES 128	$12, 119, 12$	2^{143}	143×143	11×128	20
SHA3-256	$12, 122, 12$	2^{146}	146×146	6×256	216
AES 192	$15, 177, 15$	2^{207}	207×207	13×192	159
SHA3-384	$15, 180, 15$	2^{210}	210×210	7×384	12
AES 256	$17, 238, 17$	2^{272}	272×272	15×256	206
SHA3-512	$17, 240, 17$	2^{274}	274×274	7×512	496

Table 9: Suggested sets of parameters for $SRTPI\left(\mathbb{F}_2, n_1, n_2, n_3\right)$ and $TPISig\left(\mathbb{F}_2, n_1, n_2, n_3\right)$ for the case where Assumption 5.1 does not hold

Category	**Parameters** n_1, n_2, n_3	**Expected** **Security**	**Matrices** **Size**	**Plain – Text** **Vars.**	**Rand.** **Vars.**
AES 128	$13, 119, 12$	2^{144}	144×144	12×128	11
SHA3-256	$18, 122, 12$	2^{152}	152×152	8×256	148
AES 192	$16, 177, 15$	2^{208}	208×208	14×192	144
SHA3-384	$21, 180, 15$	2^{216}	216×216	9×384	324
AES 256	$17, 238, 17$	2^{272}	272×272	15×256	206
SHA3-512	$23, 240, 17$	2^{280}	280×280	10×512	400

Table 10: Suggested sets of parameters for $SRTPI\left(\mathbb{F}_2, n_1, n_2, n_3\right)$ and $TPISig\left(\mathbb{F}_2, n_1, n_2, n_3\right)$ for the case where Assumption 5.1 does not hold, targeted to operate with bytes

4.5 Perfect Forward Secrecy, Resistance To Side Channel Attacks, Resistance To Multi-Key Attacks, Resistance To Misuse

These are currently under investigation regarding the suggested schemes $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$ and $TPISig\left(\mathbb{F}_2, t, 2t, t\right)$.

Regarding Timing Attacks, in the direct implementation of RSA and ELGAMAL there is a distinction between the base and the power, and the value of the bits of the power representation significantly influences the execution time of each iteration (almost doubling the execution time: square and multiply or just square, and each such operation takes a period of time, such that these can be easily distinguished) - leading to Timing Attacks and thus to more execution time (and space) - in order to flatten the CPU execution time.

In this context, we can say that for the suggested schemes there is almost no distinction between secrete parameters and public parameters that appear in the the decryption process - almost all are just matrices (except for the permutation π), e.g. $\mathcal{R}\left(X_0\right)$ and Z_0 from the secret key, and Y from the public channel are matrices. Therefore, in the calculation of $W = Y - \left(\mathcal{R}\left(X_0\right) + Z_0\right)$

(in this order!), one cannot distinguish the bits of $\mathcal{R}\left(X_0\right)$ and Z_0 from the known bits of Y (by doing it as explained below). Moreover, since the operations are only matrix summations and matrix multiplications (of random looking matrices), the execution time of the basic operations (i.e. sum of two bits and row-column multiplications) underlying these operations, can be easily made uniform (because they are almost always uniform), in order to not expose any bit of any secrete parameter. Note that the sum of two matrices is made in a sequence of $Read, Read, XOR, Write$ bit-operations and the multiplication of two matrices is made in a sequence of $Read, Read, AND, Write$ (in calculating the row-column entry-against-entry multiplications) and in a sequence of $Read, Read, XOR, Write$ afterwards (in order to sum-up the results). Therefore, each iteration looks the same, in terms of bit operations. Moreover, one cannot distinguish the first *Read* from the second *Read* (e.g. $Read\ Y\left[i,j\right]$ from $Read\ \left(\mathcal{R}\left(X_0\right)+Z_0\right)\left[i,j\right]$), if the precedence of the reads is made random in each iteration, i.e. for each i,j. Finally, the execution time of the XOR gate (or of the AND gate) does not vary significantly with respect to the values of the input bits. Therefore, in a careful implementation of the decryption algorithm, side channel attacks can be easily avoided. Moreover, in the parallel implementation (hardware or software implementations - see the next section), side channel attacks are naturally avoided.

4.6 Algorithm And Implementations Characteristics

4.6.1 Parallelization

In this section we provide the theoretical complexity of parallel circuits implementations (i.e. hardware implementations) of the encryption and the decryption algorithms of the system $SRTPI\left(\mathbb{F}_2, n_1, n_2, n_3\right)$, in order to prove its cost-optimality, relative to the sequential implementation, and its time-optimality. Note that $n_1 n_2$ is the number of free variables and that the involved matrices are $n \times n$ where $n = n_1 + n_2 + n_3$. We assume that multiplication of matrices with sizes $p \times q$ and $q \times r$ over $\mathbb{F}_2$ can be made by prq applications of an AND gate and $pr\left(q-1\right)$ applications of a XOR gate. We assume that each logical gate operates in a constant time, which defines the basic time unit. In parallel circuit implementation, we use prq AND gates to compute all the scalar (i.e. field) multiplications in a single time unit, and we use $pr\left(q-1\right)$ XOR gates, partitioned to pr groups of $q-1$ gates where each group is arranged as a binary-tree, in order to sum-up the results of each row-column multiplication, in $\log_2\left(q\right)$ time units. The total cost would be $pr\left(2q-1\right)$ and the total parallel time would be $1 + \log_2\left(q\right)$. Note that this basic circuit is cost-optimal (w.r.t. the direct sequential implementation) and time-optimal (i.e. there is no cost-optimal parallel circuit with AND and XOR gates that can achieve a better time).

The Encryption Circuit:
Computing

$$\widetilde{X} = mat\left(\widetilde{Q}\begin{bmatrix} m_1 \\ \vdots \\ m_{\lceil n_1 n_2/2\rceil} \\ r_1 \\ \vdots \\ r_{\lfloor n_1 n_2/2\rfloor} \\ 1 \end{bmatrix}\right)$$

in parallel costs $n^2\left(2n_1 n_2 + 1\right)$ and takes $1 + \log_2\left(n_1 n_2 + 1\right)$ time units. Sequentially, it takes $n^2\left(2n_1 n_2 + 1\right)$ time units.

In order to compute

$$\widetilde{Y} = \widetilde{X} \cdot \widetilde{C} \cdot \widetilde{X} + \widetilde{X} \cdot \widetilde{D} - \widetilde{A} \cdot \widetilde{X} - \widetilde{B}$$

in parallel, we take one circuit for multiplying $n \times n$ matrices in 4 stages and one circuit for summing $n \times n$ matrices in another 3 stages. The multiplication circuit costs $n^2\left(2n-1\right)$ and takes $1+\log_2\left(n\right)$ time units and the summation circuit costs n^2 and takes 1 time unit. Thus, the total cost is $n^2\left(2n-1\right) + n^2 = 2n^3$ and the total time is $4\left(1 + \log_2\left(n\right)\right) + 3 = 7 + 4\log_2\left(n\right)$. Sequentially it takes $4n^2\left(2n-1\right) + 3n^2 = 8n^3 - n^2$ time units.

Therefore, the total parallel cost for encryption is $C_P(n_1, n_2, n_3) = n^2(2n_1n_2 + 1) + 2n^3$ and the total parallel time is $T_P(n_1, n_2, n_3) = 4\log_2(n) + \log_2(n_1n_2 + 1) + 8$. The total sequential time is $T_S(n_1, n_2, n_3) = n^2(2n_1n_2 + 1) + 8n^3 - n^2 = 8n^3 + 2n^2n_1n_2$.

Since $C_P(n_1, n_2, n_3) = \Theta(T_S(n_1, n_2, n_3))$, it follows that the circuit is cost-optimal. The time-optimality results from a known theorem that states that the function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ defined by $f(x_1, \ldots, x_n) = x_1 + \cdots + x_n$ cannot be computed by any circuit based on NOT, AND, OR gates (or NOT, AND, XOR gates) in less than $\Omega(\log_2(n))$ time units. Since $x_1 + \cdots + x_n = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$, it follows that $n \times n$ matrix multiplication cannot be done in less than $\Omega(\log_2(n))$ time units. Since the circuit defined above is cost-optimal and achieve the lowest time bound, it follows that the circuit is time-optimal.

To conclude, we have the following time and cost for the encryption circuit:

$$\begin{cases} T_P(n_1, n_2, n_3) = 4\log_2(n) + \log_2(n_1n_2 + 1) + 8 \\ C_P(n_1, n_2, n_3) = n^2(2n_1n_2 + 1) + 2n^3. \end{cases}$$

Assuming that each AND and XOR gate operates in 10^{-6} [msec], the expected parallel encryption time of $SRTPI(\mathbb{F}_2, 16, 32, 16)$ would be $41 \cdot 10^{-6}$ [msec].

The Decryption Circuit:
Computing:

$$Y = P_L^{-1} \cdot \widetilde{Y} \cdot P_R$$

in parallel costs $n^2(2n - 1)$ and takes $2(1 + \log_2(n))$ time units. Sequentially, it takes $2n^2(2n - 1)$ time units.

Computing:

$$W = Y - \mathcal{R}(X_0) - Z_0$$

in parallel costs n^2 and takes 2 time units. Sequentially, it takes $2n^2$ time units.

Computing:

$$V = mat\left(\Gamma^{-1} vec(W)\right)$$

in parallel costs $n^2\left(2n^2 - 1\right)$ and takes $1 + \log_2\left(n^2\right) = 1 + 2\log_2(n)$ time units. Sequentially, it takes $n^2\left(2n^2 - 1\right)$ time units. Note that Γ is $n^2 \times n^2$ matrix.

Computing:

$$M = C \cdot V$$

in parallel costs $n^2(2n-1)$ and takes $1+\log_2(n)$ time units. Sequentially, it takes $n^2(2n-1)$ time units.

Finally, extracting $m_1,\ldots,m_{\lceil n_1 n_2/2\rceil}$ from the $M_{1,2}$ block of M, using the permutation π in parallel costs $\lceil n_1 n_2/2\rceil$ and takes 1 time unit. Sequentially, it takes $\lceil n_1 n_2/2\rceil$ time units.

To conclude, we have a total cost $C_P(n_1,n_2,n_3) = n^2(2n-1)+n^2+n^2\left(2n^2-1\right)+n^2(2n-1)+\lceil n_1 n_2/2\rceil = 2n^4+4n^3-2n^2+\lceil n_1 n_2/2\rceil$ and total parallel time $T_P(n_1,n_2,n_3) = 2(1+\log_2(n))+2+1+2\log_2(n)+1+\log_2(n)+1 = 5\log_2(n)+7$. The total sequential time is $T_S(n_1,n_2,n_3) = 2n^2(2n-1)+2n^2+n^2\left(2n^2-1\right)+n^2(2n-1)+\lceil n_1 n_2/2\rceil = 4n^4+6n^3-2n^2+\lceil n_1 n_2/2\rceil$.

Similarly to the encryption circuit, the decryption circuit is cost-optimal and time-optimal.

To conclude, we have the following time and cost for the decryption circuit:

$$\begin{cases} T_P(n_1,n_2,n_3) = 5\log_2(n)+7 \\ C_P(n_1,n_2,n_3) = 2n^4+4n^3-2n^2+\lceil n_1 n_2/2\rceil . \end{cases}$$

Assuming that each AND and XOR gate operates in 10^{-6} [msec], the expected parallel decryption time of $SRTPI(\mathbb{F}_2,16,32,16)$ would be $37\cdot 10^{-6}$ [msec].

Efficient parallelizations of the secrete-key generating algorithm and the public-key generating algorithm are also possible. The details will be given upon request.

4.6.2 Implicitly Authenticated Key Exchange

Assume that Alice and Bob has a $SRTPI(\mathbb{F}_2,t,2t,t)$ system (with different public-key and secret-key but with the same parameters t) and they want to share a secrete key K, say, for some joint symmetric encryption system (e.g. AES-256). Let the public-key of Alice be given by

$$pk_A = \left(\widetilde{A}^{(a)},\widetilde{B}^{(a)},\widetilde{C}^{(a)},\widetilde{D}^{(a)},\widetilde{Q}^{(a)}\right),$$

with secrete-key

$$sk_A = \left(A^{(a)},B^{(a)},C^{(a)},D^{(a)},L^{(a)},\mathcal{R}^{(a)}\left(X_0^{(a)}\right),Z_0^{(a)},P_L^{(a)-1},P_R^{(a)},\Gamma^{(a)-1},\pi^{(a)}\right).$$

Let the public-key of Bob be give by

$$pk_B = \left(\widetilde{A}^{(b)},\widetilde{B}^{(b)},\widetilde{C}^{(b)},\widetilde{D}^{(b)},\widetilde{Q}^{(b)}\right),$$

with secrete-key

$$sk_B = \left(A^{(b)}, B^{(b)}, C^{(b)}, D^{(b)}, L^{(b)}, \mathcal{R}^{(b)}\left(X_0^{(b)}\right), Z_0^{(b)}, P_L^{(b)-1}, P_R^{(b)}, \Gamma^{(b)-1}, \pi^{(b)}\right).$$

Let us denote by $E_{pk_A}(\mathbf{m}) = \widetilde{Y^{(a)}}(\mathbf{m}) = \widetilde{\mathcal{R}^{(a)}}\left(\widetilde{X^{(a)}}(\mathbf{m})\right)$ the encryption of a message $\mathbf{m} \in \mathbb{F}_2^{2t^2}$ by Alice and let us denote by $E_{pk_B}(\mathbf{m}) = \widetilde{Y^{(b)}}(\mathbf{m}) = \widetilde{\mathcal{R}^{(b)}}\left(\widetilde{X^{(b)}}(\mathbf{m})\right)$ the encryption of the message $\mathbf{m}$ by Bob. We denote by $\mathbf{m} = D_{sk_A}\left(\widetilde{Y^{(a)}}(\mathbf{m})\right)$ and by $\mathbf{m} = D_{sk_B}\left(\widetilde{Y^{(b)}}(\mathbf{m})\right)$ the related decryption functions.

In the protocol appearing as Algorithm 10, we follow the construction of the FHMQV-C protocol for implicit key exchange (see [26], [23] and [27]), while keeping its main and principal ingredients. Let $\mathbf{m}^{(a)} \in \mathbb{F}_2^{2t^2}$ and $\mathbf{m}^{(b)} \in \mathbb{F}_2^{2t^2}$ denote secrete keys of Alice and Bob, respectively, for the purpose of the key exchange protocol. The protocol appears in Algorithm 10, where KDF_1 and KDF_2 denote key derivation functions and MAC denote a message authentication code. The x, y denote ephemeral keys and $\mathbf{m}^{(a)}, \mathbf{m}^{(b)}$ denote static keys of Alice and Bob, respectively.

Among the properties of the FHMQV-C protocol are: security against impersonation and man-in-the-middle attacks, key confirmation and perfect forward secrecy (when adding session key expiration - see [9]). The suggested protocol should be investigated, but since the suggested protocol was designed to keep all the ingredients of the original protocol unchanged, we believe that the suggested protocol keeps these properties unchanged.

5 Analysis of the algorithms with respect to known attacks

5.1 Cryptanalysis Of $SRTPI\left(\mathbb{F}_q, n_1, n_2, n_3\right)$

In the following, the cryptanalysis of $SRTPI\left(\mathbb{F}_q, t, 2t, t\right)$ is given, in view of some algebraic known attacks. We also provide some specialized attacks aimed to the use of algebraic or functional Riccati equations, in view of some known facts concerning Riccati equations and in view of the construction of the encryption system. Some of attacks are similar to the attacks on the digital signature $TPSig\left(\mathbb{F}_q, t, 2t, t\right)$, and thus will be discussed here shortly (and more widely in the relevant section on $TPSig\left(\mathbb{F}_q, t, 2t, t\right)$) and will be commented here to emphasize only the differences.

5.1.1 General Attacks On $SRTPI$

The general attacks on $SRTPI\left(\mathbb{F}_q, t, 2t, t\right)$ will use some MQE solvers in order to solve the polynomial quadratic equations raising from the public-key map $\widetilde{Y} = \widetilde{\mathcal{R}}\left(\widetilde{X}\right)$ and

$$\widetilde{X} = mat\left(\widetilde{Q}\begin{bmatrix} m_1 \\ \vdots \\ m_{t^2} \\ r_1 \\ \vdots \\ r_{t^2} \\ 1 \end{bmatrix}\right) \tag{20}$$

From (20) we have that: $\widetilde{x}_{i,j} = \sum_{k=1}^{t^2} \alpha_{i,j}^{(k)} m_k + \sum_{k=1}^{t^2} \beta_{i,j}^{(k)} r_k + \gamma_{i,j}$, with known coefficients $\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_{i,j},\ 1 \leq\ i,j \leq\ 4t, 1 \leq\ k \leq\ t^2$. We therefore have $m = 16t^2$ quadratic equations with $n = 2t^2$ variables and $\alpha = \frac{m}{n} = 8$.

The complexity of solving the related set of equations with the $F_5, HyridF_5$ and $BooleanSolver$ algorithms, for $SRTPI\left(\mathbb{F}_q, 12, 24, 12\right)$ (128-bit) and for $SRTPI\left(\mathbb{F}_q, 16, 32, 16\right)$ (256-bit) is given in Table 11 (for more details see section 5.2.1).

We conclude that the systems $SRTPI\left(\mathbb{F}_q, 12, 24, 12\right)$ and $SRTPI\left(\mathbb{F}_q, 16, 32, 16\right)$ stand in their aimed security levels, regarding the general attacks.

5.1.2 UOV Attack On $SRTPI$

In order to apply the UOV attack, we need to identify an oil and vinegar partition in the set of quadratic equations arising from the algebraic Riccati equation $\widetilde{Y} = \widetilde{\mathcal{R}}\left(\widetilde{X}\right)$. Since the partition of X as $X = \left[\begin{array}{cc} X_1 & X_2 \end{array}\right]$, where X_1 is $4t \times 3t$

MQE Solver	128-bit	256-bit
F_5	$2^{507.8099}$	$2^{576.5974}$
$HybridF_5$	$2^{309.1691}$	$2^{533.8997}$
$BooleanSolver$	$2^{151.5744}$	$2^{269.4656}$

Table 11: Complexity of general attacks on 128-bit and 256-bit $SRTPI$

and X_2 is $4t \times t$, lead to oil and vinegar partition (see section 5.2.2), it is natural to start with this partition (otherwise, we don't know how to partition the variables in $\widetilde{X}$ directly). Let $P_R^{-1} = \begin{bmatrix} Q_R^{(1)} \\ Q_R^{(2)} \end{bmatrix}$ partitioned accordingly. Then, $\widetilde{X} = P_L X P_R^{-1} = P_L X_1 Q_R^{(1)} + P_L X_2 Q_R^{(2)}$. We can see that the oil and vinegar are well mixed in $\widetilde{X}$ because of the use of P_R^{-1} (this emphasizes the need of P_R) and thus, the UOV attack fails. Since the UOV attack anyway has much larger complexity than the brute-force attack, even if an oil and vinegar partition of the variables of $\widetilde{X}$ could be found (see section 5.2.2), we conclude the UOV attack has a total failure regarding $SRTPI\left(\mathbb{F}_q, t, 2t, t\right)$ systems.

5.1.3 Key Recovery Attack On $SRTPI$

In the following attack we try to find an equivalent key to the secrete-key of Bob, in order to be able to invert the Riccati operator similarly to the way Bob inverts it. From the public-key map (20) we have that: $\widetilde{x}_{i,j} = \sum_{k=1}^{t^2} \alpha_{i,j}^{(k)} m_k + \sum_{k=1}^{t^2} \beta_{i,j}^{(k)} r_k + \gamma_{i,j}$, with known coefficients $\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_{i,j}$, where $1 \leq \ i,j \leq \ 4t, 1 \leq \ k \leq \ t^2$. Since $\widetilde{X} = \widetilde{X_0} + \widetilde{V}$ and since $\widetilde{X_0}$ is a constant matrix, it is natural to define $\hat{X}_0 = (\gamma_{i,j})$ and $\hat{V} = \left(\sum_{k=1}^{t^2} \alpha_{i,j}^{(k)} m_k + \sum_{k=1}^{t^2} \beta_{i,j}^{(k)} r_k\right)$. Now, let $\widetilde{Y}$ be captured from the public channel. Then, in order to find $m_1, \ldots, m_{t^2}$, we need to solve:

$$\begin{aligned}\widetilde{Y} = \mathcal{R}\left(\hat{X}_0 + \hat{V}\right) = \\ = \mathcal{R}\left(\hat{X}_0\right) + \hat{V}\widetilde{C}\hat{V} + \hat{V}\left(\widetilde{D} + \widetilde{C}\hat{X}_0\right) - \left(\widetilde{A} - \hat{X}_0\widetilde{C}\right)\hat{V},\end{aligned}$$

for $\hat{V}$. Obviously, we cannot expect that $\hat{V}\widetilde{C}\hat{V}$ would be a constant (see Remark 5.1 below). We therefore have $16t^2$ equations with $2t^2$ variables and the complexity of solving such a system is given in Table 11.

Remark 5.1 *We could try to compute a constant matrix $\hat{Z}_0$ such that $\hat{V}\widetilde{C}\hat{V} = \hat{Z}_0$ for all $m_1, \ldots, m_{t^2}$ and some $r_1, \ldots, r_{t^2}$. To this purpose, let $\hat{V} = \hat{V}_m + \hat{V}_r$ where $\hat{V}_m = \left(\sum_{k=1}^{t^2} \alpha_{i,j}^{(k)} m_k\right)$ and $\hat{V}_r = \left(\sum_{k=1}^{t^2} \beta_{i,j}^{(k)} r_k\right)$. Now, it is reasonable that $\hat{V}\widetilde{C}\hat{V} = \hat{Z}_0$ for all $m_1, \ldots, m_{t^2}$ and some $r_1, \ldots, r_{t^2}$, only if*

$$\hat{V}_m\widetilde{C}\hat{V}_m + \hat{V}_m\widetilde{C}\hat{V}_r + \hat{V}_r\widetilde{C}\hat{V}_m = 0, \tag{21}$$

in which case $\hat{Z}_0 = \hat{V}_r \widetilde{C} \hat{V}_r$ for the particular $r_1, \ldots, r_{t^2}$ mentioned above. Now, (21) is a functional Riccati equation, which is assumed to be harder than algebraic Riccati equations.

We conclude that the key recovery attack on the $SRTPI\left(\mathbb{F}_q, t, 2t, t\right)$ system does not have any advantage over the brute-force attack.

5.1.4 A Specialized Attack On *SRTPI*

The Specialized Attack on *TPSig* had a total failure (see section 5.2.4). The results are the same for *SRTPI*.

5.1.5 A Direct Min-Rank Attack On *SRTPI*

Let $n = 4t$. The Riccati equation $\widetilde{Y} = \widetilde{\mathcal{R}}\left(\widetilde{X}\right)$ is equivalent to

$$\left[\begin{array}{cc} I & -\widetilde{X} \end{array}\right] \widetilde{T} \left[\begin{array}{c} \widetilde{X} \\ I \end{array}\right] = 0,$$

where $\widetilde{T} = \left[\begin{array}{cc} \widetilde{A} & \widetilde{B} + \widetilde{Y} \\ \widetilde{C} & \widetilde{D} \end{array}\right]$. The last is equivalent to

$$\left(\left[\begin{array}{cc} \widetilde{A} & \widetilde{B} + \widetilde{Y} \end{array}\right] - \widetilde{X}\left[\begin{array}{cc} \widetilde{C} & \widetilde{D} \end{array}\right]\right)\left[\begin{array}{c} \widetilde{X} \\ I \end{array}\right] = 0. \qquad (22)$$

Let $\widetilde{X} = \sum_{1 \leq i,j \leq n} \widetilde{x}_{i,j} E_{i,j}$, where $E_{i,j}$ is the $n \times n$ matrix with 1 in the (i,j)'th entry and 0 in all other entries. Now, in view of Theorem 1.3, we can write $\widetilde{X} = V_{1,2} V_{2,2}^{-1}$. Therefore, (22) is equivalent to:

$$\left(\left[\begin{array}{cc} \widetilde{A} & \widetilde{B} + \widetilde{Y} \end{array}\right] - \sum_{1 \leq i,j \leq n} \widetilde{x}_{i,j} E_{i,j} \cdot \left[\begin{array}{cc} \widetilde{C} & \widetilde{D} \end{array}\right]\right)\left[\begin{array}{c} V_{1,2} \\ V_{2,2} \end{array}\right] = 0, \qquad (23)$$

with the constraint that $V_{2,2}$ should be invertible. Note that (23) alone is the Min-Rank problem (in the formulation of Kipnis & Shamir - see [21]), with the matrices $\left[\begin{array}{cc} \widetilde{A} & \widetilde{B} + \widetilde{Y} \end{array}\right]$ and $E_{i,j} \cdot \left[\begin{array}{cc} \widetilde{C} & \widetilde{D} \end{array}\right]$ for $1 \leq i,j \leq n$ as an input to the problem, where we look for $\widetilde{x}_{i,j} \in \mathbb{F}_2$ in order to make the left-hand matrix of (23) have a rank that is less than or equal to n. For a proof of NP-completeness of the Min-Rank problem see [8].

The Min-Rank attack has expected complexity $q^{\lceil \frac{m}{n} \rceil r} \cdot m^3$ where here m is the number of input matrices, $n \times n$ the matrices size, r the rank bound and q the field size (see [19]). Here we have $q = 2, m = n^2 + 1$ and $r = n = 4t$. Therefore, the Min-Rank attack on $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$ has expected complexity:

$$2^{16t^2 + 4t} \cdot \left(16t^2 + 1\right)^3.$$

For the $SRTPI\left(\mathbb{F}_2, 12, 24, 12\right)$ we have complexity of $\approx 2^{2385.5116}$, which is high above 2^{128}. For the $SRTPI\left(\mathbb{F}_2, 16, 32, 16\right)$ we have complexity of $\approx 2^{4196.0010}$, which is high above 2^{256}.

Note that solving (23) for $\widetilde{X}$ might have many irrelevant solutions because $V_{2,2}$ might be singular (the invertibility of $V_{2,2}$ is not included in (23)), although the equation $\widetilde{Y} = \widetilde{\mathcal{R}}\left(\widetilde{X}\right)$, with high probability, has unique solution because it involves $16t^2$ equations and only $2t^2$ variables.

We conclude that the direct Min-Rank attack has no advantage over the brute-force attack.

5.1.6 A Specialized Min-Rank Attack On $SRTPI$

Let $\widetilde{Y}$ be captured from the public channel and let $\widetilde{T} = \begin{bmatrix} \widetilde{A} & \widetilde{B} + \widetilde{Y} \\ \widetilde{C} & \widetilde{D} \end{bmatrix}$. In view of (5) we have:

$$\widetilde{T}V = V \begin{bmatrix} \widehat{A} & 0 \\ \widehat{C} & \widehat{D} \end{bmatrix}. \tag{24}$$

Now, taking the last n columns of (24) yields:

$$\widetilde{T} \begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix} = \begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix} \widehat{D}. \tag{25}$$

The last implies that

$$\widetilde{T}^k \begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix} = \begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix} \widehat{D}^k, \tag{26}$$

for any $k \geq 0$, from which we conclude that

$$\chi_{\widehat{D}}\left(\widetilde{T}\right) \begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix} = \begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix} \chi_{\widehat{D}}\left(\widehat{D}\right) = 0, \tag{27}$$

where $\chi_{\widehat{D}}\left(\lambda\right)$ is the characteristic polynomial of $\widehat{D}$ for which $\chi_{\widehat{D}}\left(\widehat{D}\right) = 0$ by the Cayly-Hamilton theorem. We therefore have $dim\left(ker\left(\chi_{\widehat{D}}\left(\widetilde{T}\right)\right)\right) \geq n$, since $V_{2,2}$ has to be invertible. Note that $\widetilde{T}$ depends on $\widetilde{Y}$ and thus $\widehat{D}$ depends on $\widetilde{Y}$. Let $dim\left(ker\left(\chi_{\widehat{D}}\left(\widetilde{T}\right)\right)\right) = n + \ell$. Then, we have the following lemma:

Lemma 5.1 *Let $\mathcal{M} \subseteq \mathbb{F}_q^{(m+n)\times 1}$ be a subspace of dimension $n + \ell$, where $0 \leq \ell \leq m$ and $m, n \geq 1$. Then, the number of n-dimensional subspaces of $\mathcal{M}$ is given by the Gaussian Binomial Coefficient:*

$$\begin{bmatrix} n+\ell \\ n \end{bmatrix}_q = \frac{\left(q^{n+\ell} - 1\right)\left(q^{n+\ell} - q\right)\cdots\left(q^{n+\ell} - q^{n-1}\right)}{\left(q^n - 1\right)\left(q^n - q\right)\cdots\left(q^n - q^{n-1}\right)}, \tag{28}$$

which is bounded below by $q^{n\ell}$*.*

The bound $q^{n\ell}$ is concluded from the fact that $\frac{(q^{n+\ell}-q^j)}{(q^n-q^j)} \geq q^\ell$.

We therefore have the following attack on $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$ (where here $n = m = 4t$), which can be considered as a specialized Min-Rank attack:

1. Let us denote by $\chi\left(\lambda\right) = \lambda^n + a_0 + a_1\lambda + \ldots + a_{n-1}\lambda^{n-1}$ a general degree n monic polynomial. For each one of the q^n possible n-tuples $a_0, \ldots, a_{n-1}$, if $dim\left(ker\left(\chi\left(\widetilde{T}\right)\right)\right) \geq n$ (equivalently, if $rank\left(\chi\left(\widetilde{T}\right)\right) \leq m$) go to step 2.

2. For the given n-tuple $a_0, \ldots, a_{n-1}$, for all the n-dimensional subspaces of $dim\left(ker\left(\chi\left(\widetilde{T}\right)\right)\right)$ and for each such subspace let $\{v_1, v_2, \ldots, v_n\}$ be a basis for it. Let $\begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix} = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}$ and check that $V_{2,2}$ is invertible. If so, then $\widetilde{X} = V_{1,2}V_{2,2}^{-1}$ is a possible solution since $ker\left(\chi\left(\widetilde{T}\right)\right)$ is $\widetilde{T}$-invariant. If $\widetilde{Y} \neq \widetilde{\mathcal{R}}\left(\widetilde{X}\right)$ then $\widetilde{X} = V_{1,2}V_{2,2}^{-1}$ is not a solution and go to step 1., otherwise, go to step 3.

3. In order to extract a possible message $m_1, \ldots, m_{t^2}$ from the possible given $\widetilde{X}$, use the public-key to get $\widetilde{x}_{i,j} = \sum_{k=1}^{t^2} \alpha_{i,j}^{(k)} m_k + \sum_{k=1}^{t^2} \beta_{i,j}^{(k)} r_k + \gamma_{i,j}$, with known coefficients $\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_{i,j}$, where $1 \leq\ i, j \leq\ 4t, 1 \leq\ k \leq\ t^2$, and solve the linear equations in order to extract possible $m_1, \ldots, m_{t^2}$ and $r_1, \ldots, r_{t^2}$. Since $\widetilde{Y} = \widetilde{\mathcal{R}}\left(\widetilde{X}\right)$ contains $16t^2$ equations and $2t^2$ variables and since $16t^2 > 2t^2$, we would have a unique solution with high probability. Therefore, a possible solution would be the unique solution with high probability.

Remark 5.2 *Note that in stage 2., the specific vectors* $v_1, \ldots, v_n$ *which has been chosen or their order has no influence on* $\widetilde{X}$ *since that if* S *is invertible* $n \times n$ *matrix, and we replace* $\begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix}$ *with* $\begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix} S$*, we get* $\left(V_{1,2}S\right)\left(V_{2,2}S\right)^{-1} = V_{1,2}V_{2,2}^{-1} = \widetilde{X}$*. It follows that each* n*-dimensional subspace as above gives rise to a single solution and vice-versa (i.e. we have a one-to-one map between solutions of the algebraic Riccati equation and the* n*-dimensional* $\widetilde{T}$*-invariant subspaces that has the property that the lower* n *rows of* $\begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}$ *form an invertible matrix).*

Assumption 5.1 *We assume that* $dim\left(ker\left(\chi_{\widehat{D}}\left(\widetilde{T}\right)\right)\right) \geq n + \ell$ *whenever* $dim\left(ker\left(\chi_{\widehat{D}}\left(\widetilde{T}\right)\right)\right) \geq n$*, where* $\ell = \lceil \alpha n \rceil$ *for some* $0 < \alpha \leq 1$ *and* *for almost any* $\widetilde{X}$*, where* $\widetilde{T} = \begin{bmatrix} \widetilde{A} & \widetilde{B}+\widetilde{Y} \\ \widetilde{C} & \widetilde{D} \end{bmatrix}$ *and* $\widetilde{Y} = \mathcal{R}\left(\widetilde{X}\right)$*. We assume that this property holds for almost all randomly chosen matrices* $\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}$*.*

Now, in view of Lemma 5.1 (and neglecting the cost of constructing a basis for n-dimensional subspace, constructing $\begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix}$, checking the invertibility of $V_{2,2}$, computing $\widetilde{X} = V_{1,2}V_{2,2}^{-1}$ and solving the linear equations in stage 3.) the complexity of the attack under Assumption 5.1 would roughly be $2^n \cdot 2^{\alpha n^2}$, which for $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$ would be $2^{16\alpha t^2+4t}$. Note that we would have a similar complexity of the attack on $TPSig\left(\mathbb{F}_2, t, 2t, t\right)$, because searching for one possible solution and searching for a specific solution is the same because in high probability we have a unique solution for systems with $16t^2$ equations and $2t^2$ or t^2 variables.

Now, in order to sustain the security of the system, let s be the needed level of security. Then, we need that $16\alpha t^2 + 4t \geq s$ or, equivalently, that $\alpha \geq \frac{s-4t}{16t^2}$. Let $\widetilde{X}$ be the true (and almost surely the unique) solution. Then,

$$\widetilde{T}\begin{bmatrix} \widetilde{X} \\ I \end{bmatrix} = \begin{bmatrix} \widetilde{X} \\ I \end{bmatrix}\left(\widetilde{C}\widetilde{X} + \widetilde{D}\right).$$

Let $\widehat{D} = \widetilde{C}\widetilde{X} + \widetilde{D}$. Let $\chi_{\widehat{D}}\left(\lambda\right)$ be the characteristic polynomial of the matrix $\widehat{D}$. Then, the chosen secrete-key should be such that $dim\left(ker\left(\chi_{\widehat{D}}\left(\widetilde{T}\right)\right)\right) \geq n + \ell = n + \lceil \alpha n \rceil$, where $\alpha = \frac{s-4t}{16t^2}$, for any possible $\widetilde{X}$ (or at least, most of the time). Note that $\widetilde{C}\widetilde{X} + \widetilde{D} = P_R\left(CX + D\right)P_R^{-1} = P_R\left(CX_0 + D + M\right)P_R^{-1}$ and that $\widetilde{T} = \begin{bmatrix} P_L & 0 \\ 0 & P_R \end{bmatrix}\begin{bmatrix} A & B+Y \\ C & D \end{bmatrix}\begin{bmatrix} P_L^{-1} & 0 \\ 0 & P_R^{-1} \end{bmatrix}$. Therefore, $\chi_{\widehat{D}}\left(\lambda\right) = \chi_{CX_0+D+M}\left(\lambda\right)$ and $\chi_{\widehat{D}}\left(\widetilde{T}\right) = \begin{bmatrix} P_L & 0 \\ 0 & P_R \end{bmatrix}\chi_{CX_0+D+M}\left(T\right)\begin{bmatrix} P_L^{-1} & 0 \\ 0 & P_R^{-1} \end{bmatrix}$, implying that $rank\left(\chi_{\widehat{D}}\left(\widetilde{T}\right)\right) = rank\left(\chi_{CX_0+D+M}\left(T\right)\right)$. Therefore, the attack can be viewed as a specialization of the Min-Rank attack to $SRTPI$.

For the $SRTPI\left(\mathbb{F}_2, 12, 24, 12\right)$ system we have $s = 128$ and therefore $\alpha = 0.03472$, i.e. $\ell = 2$ and we need to check that $dim\left(ker\left(\chi_{CX_0+D+M}\left(T\right)\right)\right) \geq 48 + 2 = 50$ for any substitution of $m_1, \ldots, m_{144}$ and $r_1, \ldots, r_{144}$ to the $M_{1,2}$ block of M. Assuming this condition is satisfied, the attack would have complexity of 2^{128}.

For the $SRTPI\left(\mathbb{F}_2, 16, 32, 16\right)$ system we have $s = 256$ and therefore $\alpha = 0.046875$, i.e. $\ell = 3$ and we need to check that $dim\left(ker\left(\chi_{CX_0+D+M}\left(T\right)\right)\right) \geq 64 + 3 = 67$ for any substitution of $m_1, \ldots, m_{256}$ and $r_1, \ldots, r_{256}$ to the $M_{1,2}$ block of M. Assuming this condition is satisfied, the attack would have complexity of 2^{256}.

The property appearing in Assumption 5.1 might be a general property of the $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$ system or even of algebraic Riccati equations that should be proved or experimentally assessed. If it is not a general property then, good secrete-keys that achieve this property should be characterized and related instructions should be given by the authority that generates and supplies the secrete-keys, as part of any protocol for a secure use of the system. If it is not a general property and good secret keys cannot be found efficiently then, the sizes of the matrices should be raised in order that the complexity of step 1. (alone) of the attack (i.e. searching for a polynomial $\chi(\lambda)$ in order to find one for which $dim\left(ker\left(\chi\left(\widetilde{T}\right)\right)\right) \geq n$) would take $2^n \geq 2^s$, i.e. that $n \geq s$ (see Table 9 and Table 10).

In view of this attack and if Assumption 5.1 is not a general property of $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$ system or, cannot be verified efficiently, in this case we suggest to use $SRTPI\left(16, 244, 16\right)$ for the 256-bit security, where the $M_{1,2}$ block is of size 16×244 and contains 3584 variables. Since $3584 = 14 \cdot 256$, we can use the system to encrypt 14 blocks of length 256, simultaneously, or to use one block for 256 random bits and 13 blocks for simultaneous encryption. The total size of the matrices was raised by a factor of $\frac{256}{64} = 4$. Comparing with the $SRTPI\left(16, 32, 16\right)$ system performance, the encryption/decryption-time and the secrete/public-key generating-time, per a block of message, would be raised by a factor of $\frac{4^3}{13} = 4.9230$ and the space would be raised by a factor of $\frac{4^2}{13} = 1.2307$.

Similarly, for the 128-bit security, we suggest to use $SRTPI\left(12, 104, 12\right)$, where the size of $M_{1,2}$ is 12×104 and contains 1248 variables. In this case, we suggest to use $9 \cdot 128 = 1152$ bits as 9 blocks of the message each of length 128 encrypted simultaneously, and 96 random bits. The total size of the matrices was raised by a factor of $\frac{128}{48} = \frac{8}{3}$. Comparing with the $SRTPI\left(12, 24, 12\right)$ system performance, the encryption/decryption-time and the secrete/public-key generating-time, per a block of message, would be raised by a factor of $\frac{(8/3)^3}{9} = 2.1069$ and the space would be raise by a factor of $\frac{(8/3)^2}{9} = 0.7901$ (note that this is because we took the performance per block of message).

In general we would need to replace $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$ with $SRTPI\left(\mathbb{F}_2, t, s-2t, t\right)$ (where t^2 is the block length of a message and we generally assume that $s = t^2$ is the needed security level). The resulting size of the $M_{1,2}$ block would be $t \times (s-2t)$, with $ts - 2t^2$ variables. Let $s = \left\lfloor \frac{s}{t} \right\rfloor t + r$. Then, $ts - 2t^2 = \left(\left\lfloor \frac{s}{t} \right\rfloor - 2\right) t^2 + rt$. We therefore can use the system to simultaneously encrypt $\left\lfloor \frac{s}{t} \right\rfloor - 2$ blocks and use rt bits as random bits or, $\left\lfloor \frac{s}{t} \right\rfloor - 3$ blocks and $t^2 + rt$ random bits. The size of the matrices would be $s \times s$ instead of $4t \times 4t$ and thus, the ratio would be $\frac{s}{4t}$. The encryption/decryption-time and the secrete/public-key generating-time, per a block of message, in the sequential implementation, would be raised by a factor of $\left(\frac{s}{4t}\right)^3 / \left(\left\lfloor\frac{s}{t}\right\rfloor - 3\right) \approx \frac{s^2}{64t^2}$ and the space usage, per block, would be raised by a factor of $\left(\frac{s}{4t}\right)^2 / \left(\left\lfloor\frac{s}{t}\right\rfloor - 3\right) \approx \frac{s}{16t}$.

Remark 5.3 *Note that any set of quadratic equations can be reduced to a set of Riccati equations (see [24]). Therefore, if Assumption 5.1 is false then, any scheme based on MQE's should raise the number of variables in view of the attack, in order to sustain its claimed security level. Note also that if $\chi_T(\lambda)$ is factored as $p_1^{k_1}(\lambda)\, p_2^{k_2}(\lambda) \cdots p_r^{k_r}(\lambda)$ (with $p_i(\lambda)$ reducible, by the Berlekamp's algorithm or by the Cantor-Zassenhaus algorithm) then, any solution to the Integer-Linear-Programming instance:*

$$\begin{cases} j_1 + j_2 + \cdots + j_r = n \\ 0 \le j_i \le k_i, \;\; i = 1, \ldots, r, \end{cases}$$

where $k_i,\; i = 1, \ldots, r$ are given such that $k_1 + k_2 + \cdots + k_r = n + m$, gives rise to the polynomial $\chi_{\widehat{D}}(\lambda)$, since $\chi_{\widetilde{T}}(\lambda) = \chi_{\widehat{A}}(\lambda)\, \chi_{\widehat{D}}(\lambda)$. This observation might help to reduce the worst-case complexity 2^n of step 1. of the attack. Also, one should explore the advantage of the attack in the quantum computational model.

5.1.7 Differential Attack On *SRTPI*

Let $\widetilde{X_1}, \widetilde{X_2}$ be two known matrices computed from

$$\widetilde{X} = mat\left(\widetilde{Q}\begin{bmatrix} m_1 \\ \vdots \\ m_{t^2} \\ r_1 \\ \vdots \\ r_{t^2} \\ 1 \end{bmatrix}\right)$$

by choosing $m_1, \ldots, m_{t^2}$ and $r_1, \ldots, r_{t^2}$. Note that

$$\widetilde{X_1} - \widetilde{X_2} = \widetilde{V_1} - \widetilde{V_2} = \widetilde{C}^{+}\left(\widetilde{M^{(1)} - M^{(2)}}\right).$$

We have:

$$\widetilde{Y_2} - \widetilde{Y_1} = \left(\widetilde{X_2} - \widetilde{X_1}\right)\widetilde{C}\widetilde{X_0} + \widetilde{X_0}\widetilde{C}\left(\widetilde{X_2} - \widetilde{X_1}\right) + \left(\widetilde{X_2} - \widetilde{X_1}\right)\widetilde{D} - \widetilde{A}\left(\widetilde{X_2} - \widetilde{X_1}\right). \tag{29}$$

Assume that the linear system (29) contains sufficient independent equations and that $\widetilde{X_0}$ can be fully recovered. Then,

$$\left(\widetilde{X_1} - \widetilde{X_0}\right)\widetilde{C}\left(\widetilde{X_1} - \widetilde{X_0}\right) = \widetilde{V_1}\widetilde{C}\widetilde{V_1} = \widetilde{Z_0}.$$

Now, for a given $\widetilde{Y}$ caught from the public channel, we have:

$$\widetilde{Y} - \widetilde{\mathcal{R}}\left(\widetilde{X_0}\right) - \widetilde{Z_0} = \widetilde{V}\left(\widetilde{D} + \widetilde{C}\widetilde{X_0}\right) - \left(\widetilde{A} - \widetilde{X_0}\widetilde{C}\right)\widetilde{V},$$

and because $\widetilde{\Gamma} = \left(\widetilde{D} + \widetilde{C}\widetilde{X_0}\right) \otimes I - I \otimes \left(\widetilde{A} - \widetilde{X_0}\widetilde{C}\right)$ is invertible, $\widetilde{V}$ can be recovered as:

$$\widetilde{V} = mat\left(\widetilde{\Gamma}^{-1}\left(\widetilde{Y} - \widetilde{\mathcal{R}}\left(\widetilde{X_0}\right) - \widetilde{Z_0}\right)\right).$$

Now, $\widetilde{V_j} = \widetilde{C}^{+}\widetilde{M^{(j)}} + \widetilde{\widehat{U_0}},\ j = 1, 2$ and therefore $\widetilde{X_2} - \widetilde{X_1} = \widetilde{C}^{+}\left(\widetilde{M^{(2)}} - \widetilde{M^{(1)}}\right)$. Let us partition C as $\begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,1} & C_{2,2} & C_{2,3} \\ 0 & 0 & 0 \end{bmatrix}$. Then

$$\begin{aligned} &(X_2 - X_1)\,C = C^{+}\left(M^{(2)} - M^{(1)}\right)C \\ &= C^{+}\begin{bmatrix} \left(M^{(2)}_{1,2} - M^{(1)}_{1,2}\right)C_{2,1} & \left(M^{(2)}_{1,2} - M^{(1)}_{1,2}\right)C_{2,2} & \left(M^{(2)}_{1,2} - M^{(1)}_{1,2}\right)C_{2,3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned} \tag{30}$$

Since $\left(M^{(2)}_{1,2} - M^{(1)}_{1,2}\right)$ is $t \times 2t$, it follows that $rank\left((X_2 - X_1)\,C\right) \leq t$. Using Lemma 1.2, we conclude that $rank\left(I_{4t} \otimes (X_2 - X_1)\,C + C^T (X_2 - X_1)^T \otimes I_{4t}\right) \leq 8t^2$. Therefore, (29) can reveal at most $8t^2$ entries of $\widetilde{X_0}$ which contains $16t^2$ entries. Moreover, using another messages $\widetilde{X_3}, \widetilde{X_4}, \ldots$ and their differences, would not add any new information about $\widetilde{X_0}$, because (30) would have the same form for any related difference. Therefore, in order to complete the attack, one would have to solve:

$$\begin{aligned} \widetilde{Y} &= \widetilde{\mathcal{R}}\left(\widetilde{X_0}\right) + \widetilde{Z_0} + V\left(\widetilde{D} + \widetilde{C}\widetilde{X_0}\right) - \left(\widetilde{A} - \widetilde{X_0}\widetilde{C}\right)\widetilde{V} \\ &= \widetilde{\mathcal{R}}\left(\widetilde{X_0}\right) + \left(\widetilde{X_1} - \widetilde{X_0}\right)\widetilde{C}\left(\widetilde{X_1} - \widetilde{X_0}\right) + \widetilde{V}\left(\widetilde{D} + \widetilde{C}\widetilde{X_0}\right) - \left(\widetilde{A} - \widetilde{X_0}\widetilde{C}\right)\widetilde{V} \\ &= \widetilde{X_0}\widetilde{D} - \widetilde{A}\widetilde{X_0} - \widetilde{B} + \widetilde{X_1}\widetilde{C}\widetilde{X_1} - \widetilde{X_1}\widetilde{C}\widetilde{X_0} - \widetilde{X_0}\widetilde{C}\widetilde{X_1} + \\ &\quad + \widetilde{V}\left(\widetilde{D} + \widetilde{C}\widetilde{X_0}\right) - \left(\widetilde{A} - \widetilde{X_0}\widetilde{C}\right)\widetilde{V}, \end{aligned} \tag{31}$$

(we used the fact that $2\widetilde{X_0}\widetilde{C}\widetilde{X_0} = 0$ over $\mathbb{F}_2$) where the equation is quadratic in the unknown missing variable of $\widetilde{X_0}$ and in $\widetilde{V}$ (in the cross-terms between $\widetilde{X_0}$ and $\widetilde{V}$, i.e. in the last two terms of (31)). We therefore have $16t^2$ polynomial quadratic equations with $10t^2$ variables - in the worst case and counting $r_1, \ldots, r_{t^2}$ as variables (see Remark 5.5 below).

Applying *BooleanSolver* (Las Vegas version) to (31) would result in expected complexity of $2^{0.7023 \cdot 16t^2}$. Therefore, the complexity of the differential attack on $SRTPI\left(\mathbb{F}_2, 12, 24, 12\right)$ using *BooleanSolver* is $2^{202.2782}$, which is sufficient for 2^{128} security. The complexity of the differential attack on $SRTPI\left(\mathbb{F}_2, 16, 32, 16\right)$ using *BooleanSolver* is $2^{359.6075}$, which is sufficient for 2^{256} security.

5.2 Cryptanalysis Of $TPSig\left(\mathbb{F}_q, n_1, n_2, n_3\right)$

The cryptanalysis of the $TPSig\left(\mathbb{F}_q, n_1, n_2, n_3\right)$ will be made for $q = 2, n_1 = t, n_2 = 2t, n_3 = t$, for various values of the parameter t. We therefore have $n = n_1 n_2 = 2t^2$ variables and $m = \left(n_1 + n_2 + n_3\right)^2 = 16t^2 = 8n$ equations.

5.2.1 General Attacks On $TPSig$

One of the fastest known algorithm for computing Gröbner bases is the F_5 algorithm (see [16]). For a sequence of m polynomials with n variables, its complexity is given by:

$$m \cdot \left(\begin{array}{c} n + d_{reg} - 1 \\ d_{reg} \end{array} \right)^{\omega}, \tag{32}$$

where d_{reg} is the degree of regularity of the sequence and $2 \leq \omega \leq 3$ is the linear algebra constant. The degree of regularity of semi-regular sequence of m polynomials with n variables is the power of z in the first term with non-positive coefficient in the power series expansion of:

$$\frac{\prod_{j=1}^{m} \left(1 - z^{d_j}\right)}{\left(1 - z\right)^n},$$

where d_j is the degree of the j'th polynomial in the sequence. Random sequences of polynomials are conjectured to be semi-regular (Fröberg's conjecture). The degree of regularity for such systems for $m = 8n, n = 2t^2, d_j = 2, j = 1, \ldots, m$ for various values of the parameter t is given in Table 12. The $\log_2$ of the complexity of the F_5 algorithm, using (32) with $\omega = 2$ is also given in Table 12.

We conclude that attacking $TPSig\left(\mathbb{F}_2, 12, 24, 12\right)$ with F_5 has complexity of $2^{507.8099}$, which is significantly higher than the 2^{128} security level. Attacking $TPSig\left(\mathbb{F}_2, 16, 32, 16\right)$ with F_5 has complexity of $2^{576.5974}$, which is significantly higher than the 2^{256} security level.

Another fast algorithm to solve multivariable quadratic polynomial systems is the *Hybrid*F_5 (see [2]), which has complexity:

$$\min_{0 \leq k \leq n-1} \left(q^k \cdot m \cdot \left(\begin{array}{c} n - k + d_{reg}\left(k\right) - 1 \\ d_{reg}\left(k\right) \end{array} \right)^{\omega} \right), \tag{33}$$

where k is the number of assigned variables and $d_{reg}\left(k\right)$ is the degree of regularity of the resulting system. The algorithm passes over all the q^k possibilities of assigning values to k predefined variables and applies the F_5 algorithm to the resulting system of equations. In Table 12, k_* is the optimal k for (33), $deg_{reg}\left(k_*\right)$ is the degree of regularity of the resulting system and the eighth column is for the $\log_2$ of the $HybridF_5$ optimal complexity.

From Table 12 we conclude that attacking $TPSig\left(\mathbb{F}_2, 12, 24, 12\right)$ with $HybridF_5$ has complexity of $2^{309.1691}$, which is significantly higher than the 2^{128} security level. Attacking $TPSig\left(\mathbb{F}_2, 16, 32, 16\right)$ with $HybridF_5$ has complexity of $2^{533.8997}$, which is significantly higher than the 2^{256} security level.

In [1] a hybrid algorithm called *BooleanSolver* (BS) is represented. The algorithm fixes k variables, and the resulting polynomials together with the polynomials raising from the field equations of the rest of the variables, are checked for inconsistency through the Bezout equation. The last is actually solved by the Macaulay matrix. The algorithm has a deterministic version and a Las-Vegas version and optimal values of $k_* = 0.59n$ and $k_* = 0.45n$, resp. were found. The $\log_2$ of the complexity (expected complexity, for the Las-Vegas version) of the algorithm is given by:

$$1 - \gamma - 2\gamma \log_2 \left(\beta^\beta \left(1-\beta\right)^{1-\beta}\right),$$

where $\beta = M\left(\frac{\alpha}{\gamma}\right), \alpha = \frac{m}{n}, k = \left(1-\gamma\right) n, \alpha = \frac{m}{n}$, where $0 \leq \gamma \leq 1, \alpha \geq 1$ and

$$M\left(x\right) = -x + \frac{1}{2} + \frac{1}{2}\sqrt{2x^2 - 10x - 1 + 2\left(x+2\right)\sqrt{x\left(x+2\right)}}.$$

Remark 5.4 *According to [1] with the Las-Vegas version of the BooleanSolver algorithm, with $k_* = 0.45n$ and $\alpha = \frac{m}{n} = 8$, we have:*

$$1 - \gamma + 2\gamma \log_2 \left(\beta^\beta \left(1-\beta\right)^{1-\beta}\right) = 0.5263.$$

Thus, one needs to take $\lceil 128/0.5263 \rceil = 244$ variables, in order to guaranty a security level of 2^{128}. In $TPSig\left(\mathbb{F}_2, 12, 24, 12\right)$ we took $12 \times 24 = 288$ variables. Thus, the BooleanSolver complexity is $2^{151.5744}$, which is sufficient for the 2^{128} security level. Similarly, in order to guarantee a security level of 2^{256}, one needs to take $\lceil 256/0.5263 \rceil = 487$ variables. In $TPSig\left(\mathbb{F}_2, 16, 32, 16\right)$ we took $16 \times 32 = 512$ variables. The BooleanSolver complexity is $2^{269.4656}$, which is sufficient for the 2^{256} security level. Note that BooleanSolver outperforms F_5 and $HybridF_5$ up to $n \approx 1400$ variables (for $m = 8n$) and that there is a phase transition in k_* at $n = 512$, making a switch between guessing all the variables except for 2 and guessing a small number of variables and solving for almost all variables.

5.2.2 UOV Attack On $TPSig$

Let a system of quadratic polynomials over $\mathbb{F}_q$ be given by:

$$f_k\left(x_1, \ldots, x_n\right) = \sum_{i \in V, j \in V} a_{i,j}^{(k)} x_i x_j + \sum_{i \in V, j \in O} b_{i,j}^{(k)} x_i x_j + \sum_{i=1}^{n} c_i^{(k)} x_i + d^{(k)}, \quad (34)$$

t	$n = 2t^2$ $vars.$ $num.$	$m = 16t^2$ $equs.$ $num.$	d_{reg}	$\log_2$ F_5 $cplxty.$	k_*	$d_{reg}\,(k_*)$	$\log_2$ $HybridF_5$ $opt.\ cplxty.$	$\log_2$ BS $cplxty.$
2	8	64	44	59.6085	6	42	23.0865	4.2104
4	32	256	58	167.0066	30	51	49.6348	16.8416
6	72	576	59	259.9080	70	56	91.0696	37.8936
8	128	1024	61	343.8332	126	54	147.7966	67.3664
10	200	1600	62	415.2994	198	58	220.6431	105.2600
12	288	2304	69	507.8099	286	58	309.1691	151.5744
14	392	3136	61	519.3120	390	55	413.4633	206.3096
16	512	4096	63	576.5974	510	56	533.8997	269.4656
18	648	5184	60	596.5106	7	59	594.6228	341.0424
20	800	6400	61	639.5109	2	59	625.7855	421.0400
22	968	7744	63	688.4092	1	59	656.7262	509.4584
24	1152	9216	62	710.8630	9	59	692.6852	606.2976
26	1352	10816	63	748.0304	11	60	730.5836	711.5576
28	1568	12544	63	774.6287	15	60	759.7012	825.2384
30	1800	14400	61	779.8728	4	60	773.6332	947.3400
32	2048	16384	61	802.4298	5	60	796.7908	1077.8624
34	2312	18496	64	855.0721	11	59	812.6838	1216.8056
36	2592	20736	62	854.5379	7	60	839.3525	1364.1696
38	2888	23104	60	851.4929	0	60	851.4929	1519.9544
40	3200	25600	65	926.1768	4	60	873.0163	1684.1600
42	3528	28224	73	1035.5538	14	59	887.6438	1856.7864
44	3872	30976	62	926.2278	6	59	895.9033	2037.8336
46	4232	33856	62	942.1417	10	60	927.2345	2227.3016
48	4608	36864	62	957.3876	2	60	934.3279	2425.1904
50	5000	40000	61	959.3170	14	59	947.2987	2631.5000

Table 12: The F_5, $HybridF_5$ and $BooleanSolver$ complexities for systems with $n = 2t^2$ vars. and $m = 8n$ equs.

where $1 \leq k \leq m$, $a_{i,j}^{(k)}, b_{i,j}^{(k)}, c_i^{(k)}, d^{(k)} \in \mathbb{F}_q$ and $V \cup O$ is a partition of $\{1, \ldots, n\}$. Then, V is called the vinegar set and O the oil set. In the following, we apply the UOV attack on $TPSig\left(\mathbb{F}_q, n_1, n_2, n_3\right)$ (see [20], [22] and [25]), where $n_1 = t, n_2 = 2t, n_3 = t, n = 2t^2, m = 16t^2$. The attack has expected approximate complexity of

$$q^{n_v - n_o - 1} \cdot n_o^4,$$

where $q = |\mathbb{F}_q|$, $n_v = |V|$, and $n_o = |O|$.

In order to apply the UOV attack, we need to identify an oil and vinegar partition in the set of quadratic equations arising from the algebraic Riccati equation $Y = \mathcal{R}\left(X\right)$. For this purpose, let us partition X as $X = \left[\begin{array}{cc} X_1 & X_2 \end{array}\right]$, where X_1 is $4t \times 3t$ and X_2 is $4t \times t$. Now, since C is chosen as $C = \left[\begin{array}{c} C_1 \\ 0 \end{array}\right]$, where C_1 is $3t \times 4t$, we have:

$$XCX = \left[\begin{array}{cc} X_1 C_1 X_1 & X_1 C_1 X_2 \end{array}\right].$$

Partitioning $D = \left[\begin{array}{cc} D_{1,1} & D_{1,2} \\ D_{2,1} & D_{2,2} \end{array}\right]$, $B = \left[\begin{array}{cc} B_1 & B_2 \end{array}\right]$ and $Y = \left[\begin{array}{cc} Y_1 & Y_2 \end{array}\right]$ accordingly, we get:

$$\left\{\begin{array}{l} Y_1 = X_1 C_1 X_1 + X_1 D_{1,1} + X_2 D_{1,2} - A X_1 - B_1 \\ Y_2 = X_1 C_1 X_2 + X_1 D_{2,1} + X_2 D_{2,2} - A X_2 - B_2. \end{array}\right. \tag{35}$$

Note that (35) is exactly in the form of (34) when zooming-in to the entries, where O is the set of variables appearing in the X_1 block and V is the set of variables appearing in the X_2 block. It follows that $n_v = 4t \cdot 3t = 12t^2$ and $n_o = 4t \cdot t = 4t^2$. Therefore, the expected approximate complexity of the UOV attack on the $TPSig\left(\mathbb{F}_q, n_1, n_2, n_3\right)$ system is given by:

$$256 \cdot q^{8t^2-1} \cdot t^8. \tag{36}$$

Thus, for $q = 2$ and $t = 12$, i.e. for $TPSig\left(\mathbb{F}_2, 12, 24, 12\right)$, we have complexity of $6561 \cdot 2^{1175} \approx 2^{1187.6797}$, which is high above the 2^{128} security level.
For $q = 2$ and $t = 16$, i.e. for $TPSig\left(\mathbb{F}_2, 16, 32, 16\right)$, we have complexity of 2^{2087}, which is high above the 2^{256} security level. Note that in any case, we have $n_v = 3n_o$, which was recommended by [22].

5.2.3 Key Recovery Attack On $TPSig$

One of the most successful attacks on multivariable quadratic polynomials based systems is the Key Recovery Attack (KRA) (see [20], [7] and [29]). In the attack, we try to find keys equivalent to the secrete key, generally by using isomorphism of polynomials. Let us denote $\mathbf{m} = \langle m_1, \ldots, m_n \rangle$ and let

$$Y\left(\mathbf{m}\right) = mat\left(Q\begin{bmatrix} m_1 \\ \vdots \\ m_n \\ 1 \end{bmatrix}\right).$$

In the following, we try to find matrices that can replace the secrete-key parameters in Alice's system. We have two possible scenarios. In the first one, we try to find such a replacement that will fit to any hash values $m_1, \ldots, m_n$. In the other scenario, we will try to find such a replacement that will fit only to specific hash values $m_1, \ldots, m_n$. Let $\hat{M}_{1,2}\left(\mathbf{m}\right)$ denote the matrix that contains $m_1, \ldots, m_n$ in the left-to-right, top-to-bottom order. Let $\hat{M}\left(\mathbf{m}\right) = \begin{bmatrix} I & \hat{M}_{1,2}\left(\mathbf{m}\right) & 0 \\ 0 & -I & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and note that $CC^{+}\hat{M}\left(\mathbf{m}\right) = \hat{M}\left(\mathbf{m}\right)$ and $\hat{M}\left(\mathbf{m}\right)^2 = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{bmatrix}$. Let $\hat{Z}_0 = C^{+}\hat{M}\left(\mathbf{m}\right)^2$ and note that $\hat{Z}_0 = C^{+}$ by the structure of C. Let $\hat{V}\left(\mathbf{m}\right) = C^{+}\hat{M}\left(\mathbf{m}\right)$. Then, $\hat{V}\left(\mathbf{m}\right) C \hat{V}\left(\mathbf{m}\right) = \hat{Z}_0$. Now, we need to find a matrix $\hat{X}_0\left(\mathbf{m}\right)$ such that $Y\left(\mathbf{m}\right) = \mathcal{R}\left(\hat{X}_0\left(\mathbf{m}\right) + \hat{V}\left(\mathbf{m}\right)\right)$. Using (11) we get:

$$Y\left(\mathbf{m}\right) = \mathcal{R}\left(\hat{X}_0\left(\mathbf{m}\right)\right) + \hat{Z}_0 + \hat{V}\left(\mathbf{m}\right)\left(D + C\hat{X}_0\left(\mathbf{m}\right)\right) - \left(A - \hat{X}_0\left(\mathbf{m}\right) C\right)\hat{V}\left(\mathbf{m}\right). \tag{37}$$

Obviously, we cannot expect that $\hat{X}_0\left(\mathbf{m}\right)$ would be a constant matrix, and as we can see, (37) is quadratic in $\hat{X}_0\left(\mathbf{m}\right)$ because the first term contains $\hat{X}_0\left(\mathbf{m}\right) C \hat{X}_0\left(\mathbf{m}\right)$. The function $\hat{V}\left(\mathbf{m}\right)$ is known and we should solve the functional Riccati equation (37) for $\hat{X}_0\left(\mathbf{m}\right)$ as a function. We may assume that this task is harder than solving an algebraic Riccati equation. In the second scenario, we try to solve (37) for some specific $\mathbf{m}$. Now it is an algebraic Riccati equation with nontrivial quadratic term and is assumed to be hard. If we try to switch the role of $\hat{X}_0(\mathbf{m})$ and $\hat{V}(\mathbf{m})$ we get:

$$\begin{aligned} Y(\mathbf{m}) = {} & \mathcal{R}\left(\hat{V}(\mathbf{m})\right) + \\ & + \hat{X}_0(\mathbf{m}) C \hat{X}_0(\mathbf{m}) + \hat{X}_0(\mathbf{m})\left(D + C\hat{V}(\mathbf{m})\right) - \left(A - \hat{V}(\mathbf{m}) C\right)\hat{X}_0(\mathbf{m}) \\ = {} & \hat{Z}_0 + \hat{V}(\mathbf{m}) D - A\hat{V}(\mathbf{m}) - B + \\ & + \hat{X}_0(\mathbf{m}) C \hat{X}_0(\mathbf{m}) + \hat{X}_0(\mathbf{m})\left(D + C\hat{V}(\mathbf{m})\right) - \left(A - \hat{V}(\mathbf{m}) C\right)\hat{X}_0(\mathbf{m}), \end{aligned} \tag{38}$$

in which $\mathcal{R}\left(\hat{V}(\mathbf{m})\right)$ became linear in $\hat{V}(\mathbf{m})$, but the whole equation is still quadratic in $\hat{X}_0(\mathbf{m})$.

In another scenario, we may take $\hat{X}_0(\mathbf{m})$ as a constant randomly chosen matrix. In this case, we wont have $\hat{V}(\mathbf{m})$ in the above-mentioned form, and we would have to compute it from:

$$Y(\mathbf{m}) = \mathcal{R}\left(\hat{X}_0\right) + \hat{V}(\mathbf{m}) C \hat{V}(\mathbf{m}) + \hat{V}(\mathbf{m})\left(D + C\hat{X}_0\right) - \left(A - \hat{X}_0 C\right)\hat{V}(\mathbf{m}),$$

which is a functional Riccati equation in $\hat{V}(\mathbf{m})$.

We conclude that the Key Recovery Attack does not have any advantage over the brute-force attack that tries to solve the Riccati equation $Y(\mathbf{m}) = \mathcal{R}(X(\mathbf{m}))$ for $X(\mathbf{m})$ in the first place.

5.2.4 A Specialized Attack On $TPSig$

In the following we suggest an attack specialized to algebraic Riccati equations. From Theorem 1.3 we conclude that solving Riccati equation $Y = \mathcal{R}(X)$ is equivalent to finding block-lower-triangular form of T, with the restriction that $V_{2,2}$ has to be invertible. Equation (5) implies:

$$TV = V \begin{bmatrix} \widehat{A} & 0 \\ \widehat{C} & \widehat{D} \end{bmatrix}. \tag{39}$$

Now, taking the last n columns of (39) yields:

$$T \begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix} = \begin{bmatrix} V_{1,2} \\ V_{2,2} \end{bmatrix} \widehat{D}, \tag{40}$$

from which we conclude:

$$T \begin{bmatrix} X \\ I_n \end{bmatrix} = \begin{bmatrix} X \\ I_n \end{bmatrix} L, \tag{41}$$

where $L = V_{2,2}\widehat{D}V_{2,2}^{-1}$, which can be interpreted as a constrained generalized eigenvalue-eigenvector problem. Let us write $\begin{bmatrix} x_j \\ e_j \end{bmatrix}$ for the j'th column of $\begin{bmatrix} X \\ I_n \end{bmatrix}$. Then, (41) can be written as:

$$\begin{cases} Ax_j + (B+Y)\,e_j = \sum_{i=1}^{n} x_i \ell_{i,j} \\ Cx_j + De_j = \sum_{i=1}^{n} e_i \ell_{i,j}, \end{cases} \tag{42}$$

for $j = 1, \ldots, n$. We therefore have $(m+n) \cdot n$ equations (where mn are quadratic and n^2 are linear) with $mn + n^2$ variables.

Solving such a system (i.e. with number of equations equal to the number of variables) with the *BooleanSolver* Las-Vegas version (which is the fastest known algorithm for such systems) requires expected number of $2^{0.792n(m+n)}$ field operations (see [1]). For the $TPSig\,(\mathbb{F}_2, 12, 24, 12)$ system we have $m = n = 4 \times 12 = 48$ and $n\,(m+n) = 4608$. Thus, the expected complexity of *BooleanSolver* is $2^{3649.536}$, which is high above 2^{128}. For the $TPSig\,(\mathbb{F}_2, 16, 32, 16)$ system we have $m = n = 4 \times 16 = 64$ and $n\,(m+n) = 8192$. Thus, the expected complexity of *BooleanSolver* is $2^{6488.064}$, which is high above 2^{256}.

5.2.5 A Direct Min-Rank Attack On $TPSig$

The Direct Min-Rank attack on $SRTPI$ (see section 5.1.5), had a total failure. The results are the same for $TPSig$.

5.2.6 A Specialized Min-Rank Attack On $TPSig$

The Specialized Min-Rank attack is similar to that on $SRTPI$ (see section 5.1.6), although here, any solution of the related Riccati equation is relevant. This does not cause any problem because we essentially have a unique solution since we have $16t^2$ equations and $2t^2$ variables.

5.2.7 Differential Attack On $TPSig$

In differential attack, we try to gain knowledge on the secrete key, by using some differential-map of the public-key map (see [17] and [14]). In the following we convey a differential attack on the system $TPSig\,(\mathbb{F}_2, t, 2t, t)$. The attack will serve as a reason to the special form of the matrix C as in (17).

Let X_1, X_2 be two different signatures that Alice has been signed and let $Y_1 = \mathcal{R}\,(X_1)\,, Y_2 = \mathcal{R}\,(X_2)$. Let $V_1 = X_1 - X_0, V_2 = X_2 - X_0$ and note that $V_2 - V_1 = X_2 - X_1$. We have:

$$Y_2 - Y_1 = (X_2 - X_1)\,CX_0 + X_0C\,(X_2 - X_1) + (X_2 - X_1)\,D - A\,(X_2 - X_1)\,. \tag{43}$$

Assume that the linear system (43) contains sufficient independent equations and that X_0 can be fully recovered. Then,

$$\left(X_1 - X_0\right) C \left(X_1 - X_0\right) = V_1 C V_1 = Z_0.$$

Finally, from:

$$Y_m - \mathcal{R}\left(X_0\right) - Z_0 = V_m \left(D + C X_0\right) - \left(A - X_0 C\right) V_m,$$

and assuming that $\Gamma = \left(D + C X_0\right) \otimes I - I \otimes \left(A - X_0 C\right)$ is invertible, V_m can be recovered as:

$$V_m = mat\left(\Gamma^{-1}\left(Y_m - \mathcal{R}\left(X_0\right) - Z_0\right)\right),$$

from which we can get $X_m = X_0 + V_m$, for any forged message m.

Fortunately, because of the choice of C as in (17) and the structure of M as in (18), X_0 cannot be fully recovered from the linear system (43). In order to see this, let $W = \left(X_2 - X_1\right) D - A \left(X_2 - X_1\right)$. Then, (43) is equivalent to:

$$vec\left(Y_2 - Y_1 - W\right) = \left(I \otimes \left(X_2 - X_1\right) C + C^T \left(X_2 - X_1\right)^T \otimes I\right) vec\left(X_0\right). \quad (44)$$

Now, $V_j = C^+ M^{(j)} + \widehat{U_0},\ j = 1, 2$ and therefore $X_2 - X_1 = C^+ \left(M^{(2)} - M^{(1)}\right)$. Let us partition C as $\begin{bmatrix} C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,1} & C_{2,2} & C_{2,3} \\ 0 & 0 & 0 \end{bmatrix}$. Then

$$\begin{aligned} &\left(X_2 - X_1\right) C = C^+ \left(M^{(2)} - M^{(1)}\right) C \\ &= C^+ \begin{bmatrix} \left(M^{(2)}_{1,2} - M^{(1)}_{1,2}\right) C_{2,1} & \left(M^{(2)}_{1,2} - M^{(1)}_{1,2}\right) C_{2,2} & \left(M^{(2)}_{1,2} - M^{(1)}_{1,2}\right) C_{2,3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned} \quad (45)$$

Since $\left(M^{(2)}_{1,2} - M^{(1)}_{1,2}\right)$ is $t \times 2t$, it follows that $rank\left(\left(X_2 - X_1\right) C\right) \leq t$. Using Lemma 1.2, we conclude that $rank\left(I_{4t} \otimes \left(X_2 - X_1\right) C + C^T \left(X_2 - X_1\right)^T \otimes I_{4t}\right) \leq 8t^2$. Therefore, (44) can reveal at most $8t^2$ entries of X_0 which contains $16t^2$ random entries. Moreover, using another signed messages $X_3, X_4, \ldots$ and their differences, would not add any new information about X_0, because (45) would have the same form for any such difference. Therefore, in order to complete the attack, one would have to solve:

$$\begin{aligned} Y_m &= \mathcal{R}\left(X_0\right) + Z_0 + V_m \left(D + C X_0\right) - \left(A - X_0 C\right) V_m \\ &= \mathcal{R}\left(X_0\right) + \left(X_1 - X_0\right) C \left(X_1 - X_0\right) + V_m \left(D + C X_0\right) - \left(A - X_0 C\right) V_m \\ &= X_0 D - A X_0 - B + X_1 C X_1 - X_1 C X_0 - X_0 C X_1 + \\ &\quad + V_m \left(D + C X_0\right) - \left(A - X_0 C\right) V_m, \end{aligned} \quad (46)$$

(we used the fact that $2X_0CX_0 = 0$ over $\mathbb{F}_2$) where the equation is quadratic in the unknown missing variable of X_0 and in V_m (in the cross-terms between X_0 and V_m, i.e. in the last two terms of (46)). We therefore have $16t^2$ polynomial quadratic equations with $10t^2$ variables - in the worst case (see Remark 5.5 below).

Remark 5.5 *Assuming that $8t^2$ entries of X_0 were found and since V_m contains $2t^2$ independent variables (and $14t^2$ unknown constants) - although the exact places of the variables is unknown because of the use of π and moreover, the blocks $L_{1,2}, L_{1,3}, L_{2,3}$ are unknown and thus $M_{1,3}$ is unknown. Finally, we have a "noise" added to V_m by $\widehat{U_0}$ which is unknown.*

Applying *BooleanSolver* (Las Vegas version) would result in expected complexity of $2^{0.7023 \cdot 16t^2}$. Therefore, the complexity of the differential attack on $TPSig\,(\mathbb{F}_2, 12, 24, 12)$ using *BooleanSolver* is $2^{202.2782}$, which is sufficient for 2^{128} security. The complexity of the differential attack on $TPSig\,(\mathbb{F}_2, 16, 32, 16)$ using *BooleanSolver* is $2^{359.6075}$, which is sufficient for 2^{256} security.

6 A statement of advantages and limitations

The $TPSig\left(\mathbb{F}_q, n_1, n_2, n_3\right)$ is an efficient digital signature based on algebraic Riccati equation over a finite field. The scheme is based on provable NP-complete problem and thus, fits to the age of quantum computers. The scheme is one of the fastest undefeated known systems in terms of signing-time and signature-validation-time (see section 2). The key-generating-time is relatively little high, but is needed only once in the whole system's life. There is a trade-off between run-time and space usage and indeed the needed space is little high, but is lower than that of many known systems based on MQE's (see section 2). The scheme seems to be secure against all known algebraic attacks and its most appealing feature is that the coefficients of the quadratic map (i.e. the matrices A, B, C, D) and the secrete-key parameters (i.e. $\pi, L, M, X_0, \widehat{U_0}$) are chosen randomly and thus the resulting quadratic equations does not seem to contain any vulnerable structure. Another pleasant feature of the scheme is that the signer does not need to solve any quadratic or even linear equations in order to generate a valid signature and thus, the "vinegar" part can be as large as we wish (obviously bounded, but according to efficiency of evaluations - not of computing solutions). Note also that the "vinegar" and "oil" parts of the scheme are well mixed (see equation (35)), which makes the scheme less vulnerable to UOV attacks. Also, the scheme does not depend on isomorphism of polynomials and thus does not seem to be vulnerable to KRA attacks. As we saw, the only attack that revealed some portion of knowledge from the secrete-key (i.e. from X_0), was the differential attack (see section 5.2.7). But we also saw there that the revealed knowledge cannot be used to gain any knowledge about the unknown part of the secrete-key, no matter how many pairs of plain-text-cypher-text we have. Moreover, the attack on the remaining unknown variables of X_0 results in complexity that is higher than the complexity of the brute-force attack. Another dangerous attack is the specialized Min-Rank attack, which was discussed thoroughly in section 5.1.6. The attack does not compromise the system security level if Assumption 5.1 is valid. Otherwise, one should raise the size of the matrices (see Table 9 and Table 10), resulting in a little higher encryption/decryption-time and a little higher space used per message block (see the discussion before Remark 5.3). All these properties and limitations apply also to the suggested encryption scheme $SRTPI\left(\mathbb{F}_q, n_1, n_2, n_3\right)$. Moreover, the schemes can be modified to provide additional functionalities that extend beyond the minimum requirements of public-key encryption and digital signature (see section 4.1 and section 4.6.2) and it is straightforward to customize the scheme's parameters to meet a range of security targets and performance goals (see section 4.4). The algorithms can be implemented securely and efficiently on a variety of platforms and constrained environments (at least the encryption/decryption algorithms and the signing/signature-validation algorithms) because they involve a small number of basic matrix operations and relatively needs small space (related to other MQE based schemes). The implementation of the algorithms (at least the encryption/decryption algorithms and the signing/signature-validation algorithms) can be parallelized to achieve higher performance (see section 4.6.1). The schemes can be incorporated into existing protocols and applications (see sections 4.4 and 4.6.2), requiring as few changes as possible. Finally, the schemes are very simple since they involve only small number of basic matrix operations over the smallest field $\mathbb{F}_2$ (and thus, all the operators can be implemented by using AND and XOR gates), at least regarding the encryption/decryption algorithms (see Algorithm 4 and Algorithm 5) and the signing/signature-validation algorithms (see Algorithm 8 and Algorithm 9).

References

[1] M. Bardet, J. C. Faugère, B. Salvy, P. J. Spaenlehauer, *On the complexity of solving quadratic Boolean systems*, Elsevier Journal of Complexity, vol. 29, pp. 53-75 (2013).

[2] L. Battale, J. C. Faugère, L. Perret, *Hybrid approach for solving multivariate systems over finite fields*, J. Math. Cryptol. vol. 3, pp. 177-197, (2009).

[3] M. Bellare, P. Rogaway, *Optimal Asymmetric Encryption - How to Encrypt with RSA*, In Eurocrypt 1994, LNCS, vol. 950, pp. 92-111, Springer-Verlag, Berlin, (1994).

[4] M. Bellare, A. Desai, D. Pointcheval, P. Rogaway, *Relations Among Notions of Security for Public-key Encryption Schemes*, Advances in Cryptology - CRYPTO 19998, LNCS, vol. 1462, Springer-Verlag (1998).

[5] O. Billet, J. Ding, *Overview of Cryptanalysis Techniques in Multivariate Public Key Cryptography*, Inbook: Gröbner Bases, Coding, and Cryptography, Editors: M. Sala, T. Mora, L. Perret, S. Sakata and C. Traverso, Springer-Verlag Berlin Heidelberg, pp.263-283 (2009).

[6] J. P. Buhler, H. W. Lenstra Jr., C. Pomerance, *Factoring integers with the number field sieve*, vol. 1554, Lecture Notes in Math., pp. 50-94, Springer Verlag (1993).

[7] J. Buchmann, J. Ding, editors, Post-Quantum Cryptography, Second International Worshop, PQCrypto2008, Cincinnati, OH, USA, October 17-19, 2008, Proceedings, vol. 5299, LNCS, Springer, (2008).

[8] J. F. Buss, G. S. Frandsen, J. Shallit, *The computational complexity of some problems of linear algebra*, Journal of Computer and System Sciences, vol. 58, issue 3, pp. 572-596, (1999).

[9] R. Canetti, H. Krawczyk, *Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels*, Advances In Cryptology - EUROCRYPT, pp. 453-474, (2001).

[10] N. Courtois, A. Klimov, J. Patarin, A. Shamir, *Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations*, Proc. of EUROCRYPT 2000, LNCS 1807, pp. 392-407 (2000).

[11] N. T. Courtois, *General Principles of Algebraic Attacks and New Design Criteria for Cipher Components*, Advanced Encryption Standard - AES 2005, LNCS 3373, pp. 67-83 (2005).

[12] P. Czypek, S. Heyes and E. Thoame, *Efficient implementation of MQPKS on constrained deviecs*, CHES 2012, LNCS, Springer Heidelberg, vol. 7428, pp. 374-389 (2012).

[13] J. Ding, J. E. Gower, D. S. Schmidt, *Multivariate Public Key Cryptosystems*, Series: Advances in Information Security, Editor: Sushil Jajodia, Springer (2006).

[14] J. Ding, B. Y. Yang, C. H. O. Chen, M. S. Chen, C. M. Cheng, *New Differential-Algebraic Attacks and Reparametrization of Rainbow*, Applied Cryptography and Network Security, pp. 242-257, Springer, Berlin/Heidelberg, (2008).

[15] J. Ding, B. Y. Yang, *Multivariate Public Key Cryptography*, Inbook: Post Quantum Cryptography, Editors: D. J. Bernstein, J. Buchmann and E. Dahmen, Springer-Verlag Berlin Heidelberg, pp.193-234 (2009).

[16] J. C. Faugère, *A new efficient algorithm for computing Gröbner bases without reductions to zero (F5)*, Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation (ISSAC), pages 75-83. Editor: T. Mora, ACM Press, (2002).

[17] P. A. Fouque, L. Granboulan, J. Stern, *Differential Cryptanalysis for Multivariable Schemes*, Eurocrypt, vol. 3494, (2005).

[18] E. Fujisaki, T. Okamoto, D. Pointcheval, J. Stern, *RSA-OAEP Is Secure-under Under The RSA Assumption*, Journal Of Cryptology, vol. 17, num. 2, pp. 81-104, (2004).

[19] L. Goubin, N. T. Courtois, *Cryptanalysis of the TTM cryptosystem*, Advances in Cryptology Ů ASIACRYPT 2000, LNCS, vol. 1976, pp 44-57. Tatsuaki Okamoto, ed., Springer (2000).

[20] A. Kipnis, A. Shamir, *Cryptanalysis of the oil and vinegar signature scheme*, CRYPTO 1998, LNCS vol. 1462, pp. 257-266 (1998).

[21] A. Kipnis, A. Shamir, *Cryptanalysis of the HFE public key cryptosystem by relinearization*, CRYPTO 1999, LNCS vol. 1666, pp. 19-30 (1999).

[22] A. Kipnis, J. Patarin, L. Goubin, *Unbalanced Oil and Vinegar signature schemes*, Advances in Cryptology Ů EUROCRYPT 1999, LNCS vol. 1592, pp. 206-222. Editor: Jacques Stern, Springer (1999).

[23] S. Liu, K. Sakuri, J. Weng, F. Zhang, Y. Zhao, *Security Model and Analysis of FHMQV, Revisited*, In Proc. of Information Security and Cryptology - Inscrypt 2013. LNCS, vol. 8567, pp. 255-269, Springer, (2014).

[24] Y. Peretz, *On mulivariable encryption schemes based on simultaneous algebraic Riccati equations over finite fields*, Finite Fields and Their Applications, 39, pp. 1-35 (2016).

[25] A. Petzoldt, *Selecting and Reducing Key Sizes for Multivariable Cryptography*, Ph. D. thesis, (2013).

[26] A. P. Saar, P. Elbaz-Vincent, J. C. Bajard, *A Secure And Efficint Authenticated Diffie-Hellman Protocol*, In Proc. of Public Key Infrastructure, Services and Applications - EuroPKI 2009. LNCS, vol. 6391, pp. 83-98, Springer (2010).

[27] A. P. Saar, P. Elbaz-Vincent, *On the Security of the (F)HMQV Protocol*, Proceedings of AFRICACRYPT, (2016).

[28] W. Shen, S. Tang, *TOT, a Fast Multivariable Public Key Cryptosystem with Basic Secure Trapdoor*, Cryptology ePrint Archive, Report 2013/771, http://eprint.iacr.org/ (2013).

[29] K. A. Shim, C. M. Park and N. Koo, *Cryptanalysis of RGB, a mixed multivariable signature scheme*, Elsevier Journal on Finite Fields and Their Applications, vol. 45, pp. 406-421 (2017).

[30] C. Tao, A. Diene, S. Tang and J. Ding, *Simple Matrix Scheme for Encryption*, PQCrypto 2013, LNCS 7932, pp. 231-242 (2013).

[31] C. Wolf, B. Preneel, *Taxonomy of Public-Key Schemes based on the Problem of Multivariate Quadratic Equations*, Cryptology ePrint Archive, Report 2005/077, http://eprint.iacr.org/ (2005).

Algorithm 1 **Secrete-Key Generation in** $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$

```
Require: functions: Inverse, PseudoInverse, IsInvertible, RandBit, RandMatrix, RandPermutation, ⊗, Basic Linear Algebra Operations over $\mathbb{F}_2$
    Input: $t$
    Output: $A, B, C, D, P_L^{-1}, P_R, R\left(X_0\right), Z_0, \Gamma^{-1}, \pi$
$\pi \leftarrow RandPermutation\left(2t^2\right)$
$A \leftarrow RandMatrix\left(4t, 4t\right)$
$B \leftarrow RandMatrix\left(4t, 4t\right)$
$D \leftarrow RandMatrix\left(4t, 4t\right)$
repeat
    $C_1 \leftarrow RandMatrix\left(3t, 4t\right)$
until $IsInvertible\left(C_1 C_1^T\right) == 1$
$C \leftarrow \begin{bmatrix} C_1 \\ 0_{t\times 4t} \end{bmatrix}$
$C_1^+ \leftarrow PseudoInverse\left(C_1\right)$ //computed as $C_1^+ = C_1^T\left(C_1 C_1^T\right)^{-1}$
$C^+ \leftarrow \begin{bmatrix} C_1^+ & 0_{4t\times t} \end{bmatrix}$
repeat
    $X_0 \leftarrow RandMatrix\left(4t, 4t\right)$
    $\Gamma \leftarrow \left(CX_0 + D\right)^T \otimes I_{4t} - I_{4t} \otimes \left(A - X_0 C\right)$
until $IsInvertible\left(\Gamma\right) == 1$
$\Gamma^{-1} \leftarrow Inverse\left(\Gamma\right)$ //This line can be dropped since that $\Gamma^{-1}$ is computed inside $IsInvertible\left(\Gamma\right)$ and is returned from it
$L_{1,2} \leftarrow RandMatrix\left(t, 2t\right)$
$L_{1,3} \leftarrow RandMatrix\left(t, t\right)$
$L_{2,3} \leftarrow RandMatrix\left(2t, t\right)$
$L \leftarrow \begin{bmatrix} I_t & L_{1,2} & L_{1,3} \\ 0 & -I_{2t} & L_{2,3} \\ 0 & 0 & 0_t \end{bmatrix}$
$Z_0 \leftarrow C^+\left(CC^+L\right)^2$
$U_0 \leftarrow RandMatrix\left(4t, 4t\right)$
$Lower_1 \leftarrow I_{4t}$
$Lower_2 \leftarrow I_{4t}$
$Upper_1 \leftarrow I_{4t}$
$Upper_2 \leftarrow I_{4t}$
for $i = 2$ to $4t$ do
    for $j = 1$ to $i - 1$ do
        $Lower_1\left(i, j\right) \leftarrow RandBit$
        $Lower_2\left(i, j\right) \leftarrow RandBit$
        $Upper_1\left(j, i\right) \leftarrow RandBit$
        $Upper_2\left(j, i\right) \leftarrow RandBit$
    end for
end for
$P_L \leftarrow Lower_1 Upper_1$
$P_R \leftarrow Lower_2 Upper_2$
$P_L^{-1} \leftarrow Inverse\left(P_L\right)$
$P_R^{-1} \leftarrow Inverse\left(P_R\right)$
$R\left(X_0\right) \leftarrow X_0 C X_0 + X_0 D - A X_0 - B$
return $A, B, C, D, P_L, P_L^{-1}, P_R, P_R^{-1}, R\left(X_0\right), Z_0, \Gamma^{-1}, \pi$
```

Algorithm 2 **Public-Key Generation in** $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$

```
Require: $t$ and functions: Basic Linear Algebra Operations over $\mathbb{F}_2$
    Input: $A, B, C, D, P_L^{-1}, P_R, \pi$
    Output: $\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}, \widetilde{Q}$
$\widetilde{A} \leftarrow P_L A P_L^{-1}$
$\widetilde{B} \leftarrow P_L B P_R^{-1}$
$\widetilde{C} \leftarrow P_R C P_L^{-1}$
$\widetilde{D} \leftarrow P_R D P_R^{-1}$
$\widetilde{Q} \leftarrow SetQ\left(P_R, P_L, X_0, C, C^+, U_0, L_{1,2}, L_{1,3}, L_{2,3}, \pi, t\right)$
return $\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}, \widetilde{Q}$
```

Algorithm 3 **The Function** $SetQ\left(P_R^{-1}, P_L, X_0, C, C^+, U_0, L_{1,2}, L_{1,3}, L_{2,3}, \pi, t\right)$

```
Require: functions: vec, InitializeZeroMatrix, $\otimes$, Basic Linear Algebra Operations over $\mathbb{F}_2$
    Input: $P_R, P_L, X_0, C, C^+, U_0, L_{1,2}, L_{1,3}, L_{2,3}, \pi, t$
    Output: $\widetilde{Q}$
$\widehat{U_0} \leftarrow \left(I_{4t} - C^+C\right) U_0 \left(I_{4t} - CC^+\right)$
$PRKPL \leftarrow \left(P_R^{-1}\right)^T \otimes P_L$
$XU \leftarrow X_0 + \widehat{U_0}$
$M_{2t^2+1} \leftarrow \begin{bmatrix} I_t & 0_{t\times 2t} & L_{1,2}L_{2,3} + L_{1,3} \\ 0_{2t\times t} & -I_{2t} & L_{2,3} \\ 0_{t\times t} & 0_{t\times 2t} & 0_{t\times t} \end{bmatrix}$
$\widetilde{q}_{2t^2+1} \leftarrow PRKPL \cdot vec\left(C^+ M_{2t^2+1} + XU\right)$
for $\ell = 1$ to $2t^2$ do
    $s \leftarrow \pi\left(\ell\right)$
    $i \leftarrow \left\lfloor \frac{s-1}{2t} \right\rfloor + 1$
    $j \leftarrow s - 2t\left(i - 1\right)$
    $M_{1,2}^{(\ell)} \leftarrow InitializeZeroMatrix\left(t, 2t\right)$
    $M_{1,2}^{(\ell)}\left(i, j\right) \leftarrow 1$
    $M_\ell \leftarrow \begin{bmatrix} I_t & M_{1,2}^{(\ell)} & L_{1,2}L_{2,3} + L_{1,3} - M_{1,2}^{(\ell)}L_{2,3} \\ 0_{2t\times t} & -I_{2t} & L_{2,3} \\ 0_{t\times t} & 0_{t\times 2t} & 0_{t\times t} \end{bmatrix}$
    $\widetilde{q}_\ell \leftarrow PRKPL \cdot vec\left(C^+ M_\ell + XU\right) - \widetilde{q}_{2t^2+1}$
    $\widetilde{Q} \leftarrow \begin{bmatrix} \widetilde{q}_1 & \widetilde{q}_2 & \cdots & \widetilde{q}_{2t^2+1} \end{bmatrix}$
end for
return $\widetilde{Q}$
```

```
Algorithm 4  Encryption in $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$
Require: $\widetilde{A}, \widetilde{B}, \widetilde{C}, \widetilde{D}, \widetilde{Q}$ and functions: mat, Basic Linear Algebra Operations over $\mathbb{F}_2$
    Input: $m_1, \ldots, m_{t^2}$ and random bits $r_1, \ldots, r_{t^2}$
    Output: $\widetilde{Y} = \widetilde{R}\left(\widetilde{X}\right) := \widetilde{X}\widetilde{C}\widetilde{X} + \widetilde{X}\widetilde{D} - \widetilde{A}\widetilde{X} - \widetilde{B}$
$\widetilde{X} \leftarrow mat\left(\widetilde{Q}\begin{bmatrix} m_1 \\ \vdots \\ m_{t^2} \\ r_1 \\ \vdots \\ r_{t^2} \\ 1 \end{bmatrix}\right)$
$\widetilde{Y} \leftarrow \widetilde{X}\widetilde{C}\widetilde{X} + \widetilde{X}\widetilde{D} - \widetilde{A}\widetilde{X} - \widetilde{B}$
return $\widetilde{Y}$
```

```
Algorithm 5  Decryption in $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$
Require: $C, P_L^{-1}, P_R, R\left(X_0\right), Z_0, \Gamma^{-1}, \pi$ and functions: mat, vec, Basic Linear Algebra Operations over $\mathbb{F}_2$
    Input: $\widetilde{Y}$
    Output: $m_1, \ldots, m_{t^2}$
$Y \leftarrow P_L^{-1}\widetilde{Y}P_R$
$W \leftarrow Y - R\left(X_0\right) - Z_0$
$V \leftarrow mat\left(\Gamma^{-1}vec\left(W\right)\right)$
$M \leftarrow CV$
for $i = 1$ to $t$ do
    for $j = 1$ to $2t$ do
        $M_{1,2}\left(i, j\right) \leftarrow M\left(i, t + j\right)$
    end for
end for
for $\ell = 1$ to $t^2$ do
    $s \leftarrow \pi\left(\ell\right)$
    $i \leftarrow \left\lfloor \frac{s-1}{2t} \right\rfloor + 1$
    $j \leftarrow s - 2t\left(i - 1\right)$
    $m_\ell \leftarrow M_{1,2}\left(i, j\right)$
end for
return $m_1, \ldots, m_{t^2}$
```

Algorithm 6 Secrete-Key Generation in $TPSig\left(\mathbb{F}_2, t, 2t, t\right)$

```
Require: functions: RandMatrix, RandPermutation, Basic Linear Algebra Operations over $\mathbb{F}_2$
    Input: $t$
    Output: $\pi, L, X_0, \widehat{U_0}, C, C^+$
repeat
    $C_1 \leftarrow RandMatrix\left(3t, 4t\right)$
until $IsInvertible\left(C_1 C_1^T\right) == 1$
$C \leftarrow \begin{bmatrix} C_1 \\ 0_{t\times 4t} \end{bmatrix}$
$C_1^+ \leftarrow PseudoInverse\left(C_1\right)$ //computed as $C_1^+ = C_1^T\left(C_1 C_1^T\right)^{-1}$
$C^+ \leftarrow \begin{bmatrix} C_1^+ & 0_{4t\times t} \end{bmatrix}$
$X_0 \leftarrow RandMatrix\left(4t, 4t\right)$
$U_0 \leftarrow RandMatrix\left(4t, 4t\right)$
$\widehat{U_0} \leftarrow \left(I_{4t} - C^+C\right) U_0 \left(I_{4t} - CC^+\right)$
$L_{1,2} \leftarrow RandMatrix\left(t, 2t\right)$
$L_{1,3} \leftarrow RandMatrix\left(t, t\right)$
$L_{2,3} \leftarrow RandMatrix\left(2t, t\right)$
$L \leftarrow \begin{bmatrix} I_t & L_{1,2} & L_{1,3} \\ 0 & -I_{2t} & L_{2,3} \\ 0 & 0 & 0_t \end{bmatrix}$
$\pi \leftarrow RandPermutation\left(2t^2\right)$
return $\pi, L, X_0, \widehat{U_0}, C, C^+$ //The matrices $C, C^+$ are part of the public-key
```

Algorithm 7 Public-Key Generation in $TPSig\left(\mathbb{F}_2, t, 2t, t\right)$

```
Require: $t$, Basic Linear Algebra Operations over $\mathbb{F}_2$
    Input: $\widehat{U_0}, X_0, C, C^+, L, \pi, t$
    Output: $A, B, C, D, Q$
$A \leftarrow RandMatrix\left(4t, 4t\right)$
$B \leftarrow RandMatrix\left(4t, 4t\right)$
$D \leftarrow RandMatrix\left(4t, 4t\right)$
$XU \leftarrow X_0 + \widehat{U_0}$
$M_{2t^2+1} \leftarrow \begin{bmatrix} I_t & 0_{t\times 2t} & L_{1,2}L_{2,3} + L_{1,3} \\ 0_{2t\times t} & -I_{2t} & L_{2,3} \\ 0_{t\times t} & 0_{t\times 2t} & 0_{t\times t} \end{bmatrix}$
$X_{2t^2+1} \leftarrow C^+ M_{2t^2+1} + XU$
$Y_{2t^2+1} \leftarrow X_{2t^2+1} C X_{2t^2+1} + X_{2t^2+1} D - A X_{2t^2+1} - B$
$q_{2t^2+1} \leftarrow vec\left(Y_{2t^2+1}\right)$
for $\ell = 1$ to $2t^2$ do
    $s \leftarrow \pi\left(\ell\right)$
    $i \leftarrow \left\lfloor \frac{s-1}{2t} \right\rfloor + 1$
    $j \leftarrow s - 2t\left(i - 1\right)$
    $M_{1,2}^{(\ell)} \leftarrow InitializeZeroMatrix\left(t, 2t\right)$
    $M_{1,2}^{(\ell)}\left(i, j\right) \leftarrow 1$
    $M_\ell \leftarrow \begin{bmatrix} I_t & M_{1,2}^{(\ell)} & L_{1,2}L_{2,3} + L_{1,3} - M_{1,2}^{(\ell)} L_{2,3} \\ 0_{2t\times t} & -I_{2t} & L_{2,3} \\ 0_{t\times t} & 0_{t\times 2t} & 0_{t\times t} \end{bmatrix}$
    $X_\ell \leftarrow C^+ M_\ell + XU$
    $Y_\ell \leftarrow X_\ell C X_\ell + X_\ell D - A X_\ell - B$
    $q_\ell \leftarrow vec\left(Y_\ell\right) - \widetilde{q}_{2t^2+1}$
end for
$Q \leftarrow \begin{bmatrix} q_1 & q_2 & \cdots & q_{2t^2+1} \end{bmatrix}$
return $A, B, C, D, Q$
```

Algorithm 8 Signing with $TPSig\left(\mathbb{F}_2, t, 2t, t\right)$

```
Require: $C^+, L, X_0, \widehat{U_0}, \pi$ and functions: h, Basic Linear Algebra Operations over $\mathbb{F}_2$
    Input: $m \in \mathbb{F}_2^{(\mathbb{N})}$
    Output: $\langle m, X_m \rangle$
$\langle m_1, \ldots, m_{2t^2} \rangle \leftarrow h\left(m\right)$
for $\ell = 1$ to $2t^2$ do
    $s \leftarrow \pi\left(\ell\right)$
    $i \leftarrow \left\lfloor \frac{s-1}{2t} \right\rfloor + 1$
    $j \leftarrow s - 2t\left(i-1\right)$
    $M_{1,2}\left(i, j\right) \leftarrow m_\ell$
end for
$M_{1,3} \leftarrow -M_{1,2}L_{2,3} + L_{1,2}L_{2,3} + L_{1,3}$
$M \leftarrow \begin{bmatrix} I_t & M_{1,2} & M_{1,3} \\ 0 & -I_{2t} & L_{2,3} \\ 0 & 0 & 0_t \end{bmatrix}$
$X_m \leftarrow X_0 + C^+M + \widehat{U_0}$
return $\langle m, X_m \rangle$
```

Algorithm 9 Signature Verification with $TPSig\left(\mathbb{F}_2, t, 2t, t\right)$

```
Require: $A, B, C, D, Q$ and functions: h, mat, Basic Linear Algebra Operations over $\mathbb{F}_2$
    Input: $\langle m, X_m \rangle$
    Output: $flag$
$\langle m_1, \ldots, m_{2t^2} \rangle \leftarrow h\left(m\right)$
$V_1 \leftarrow mat\left(Q\begin{bmatrix} m_1 \\ \vdots \\ m_{2t^2} \\ 1 \end{bmatrix}\right)$
$V_2 \leftarrow X_mCX_m + X_mD - AX_m - B$
if $V_1 == V_2$ then
    $flag \leftarrow 1$
else
    $flag \leftarrow 0$
end if
return $flag$
```

Algorithm 10 FHMQV-C protocol with $SRTPI\left(\mathbb{F}_2, t, 2t, t\right)$

```
Require: $pk_A, pk_B$ and functions: secure hash function $h$, Basic Linear Algebra Operations over $\mathbb{F}_2$
  I. The initiator $A$ does the following:
    1. Choose $\mathbf{x} \in \mathbb{F}_2^{2t^2}$ randomly and
       compute $\widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right) = E_{pk_B}\left(\mathbf{m^{(a)}} + \mathbf{x}\right)$
    2. Send $\left(A, B, \widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right)\right)$ to $B$
  II. At receipt of $\left(A, B, \widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right)\right)$, $B$ does the following:
    1. Choose $\mathbf{y} \in \mathbb{F}_2^{2t^2}$ randomly and
       compute $\widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right) = E_{pk_A}\left(\mathbf{m^{(b)}} + \mathbf{y}\right)$
    2. Decrypt $\mathbf{m}^{(a)} + x = D_{sk_B}\left(\widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right)\right)$
    3. Set $\mathbf{m}_B = h\left(\mathbf{m}^{(a)} + x, \mathbf{m}^{(b)} + y, \widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right), \widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right)\right)$
    4. Compute $K_B = KDF_1\left(\mathbf{m}_B, A, B, \widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right), \widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right)\right)$
    5. Compute $t_B = MAC_{K_B}\left(B, \widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right)\right)$
    6. Send $\left(B, A, \widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right), t_B\right)$ to $A$
  III. At receipt of $\left(B, A, \widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right), t_B\right)$ $A$ does the following:
    1. Decrypt $\mathbf{m}^{(b)} + y = D_{pk_A}\left(\widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right)\right)$
    2. Set $\mathbf{m}_A = h\left(\mathbf{m}^{(a)} + x, \mathbf{m}^{(b)} + y, \widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right), \widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right)\right)$
    3. Compute $K_A = KDF_1\left(\mathbf{m}_A, A, B, \widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right), \widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right)\right)$
    4. Verify that $t_B == MAC_{K_A}\left(B, \widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right)\right)$
    5. Compute $t_A = MAC_{K_A}\left(A, \widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right)\right)$
    6. Send $t_A$ to $B$
    7. Compute $K = KDF_2\left(\mathbf{m}_A, A, B, \widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right), \widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right)\right)$
  IV. At receipt of $t_A$, $B$ does the following:
    1. Verify that $t_A == MAC_{K_B}\left(A, \widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right)\right)$
    2. Compute $K = KDF_2\left(\mathbf{m}_B, A, B, \widetilde{Y^{(b)}}\left(\mathbf{m}^{(a)} + x\right), \widetilde{Y^{(a)}}\left(\mathbf{m}^{(b)} + y\right)\right)$
  V. The shared session key is $K$
```