Private-Key Cryptosystems Based on Rank Metric

Maximilien Gadouleau
Department of Electrical and Computer Engineering
Lehigh University
Bethlehem, PA 18015
E-mail: maximilien@lehigh.edu

Zhiyuan Yan
Department of Electrical and Computer Engineering
Lehigh University
Bethlehem, PA 18015
E-mail: yan@lehigh.edu

***Abstract*— An analogue of McEliece's cryptosystem, the Gabidulin-Paramonov-Trejtakov (GPT) public-key cryptosystem is based on rank-metric-based Gabidulin codes instead of Goppa codes. The GPT cryptosystem has attracted steady attention since it is much more robust against the decoding attacks and can therefore achieve the same level of security with much smaller keys. The key sizes, nonetheless, may still be too large for some applications. To reduce the key sizes even further, we propose private-key adaptations of the GPT cryptosystem, and evaluate their robustness against several attacks.**

I. Introduction

McEliece's cryptosystem [8] based on error-correcting codes is an important alternative to the number-theory-based cryptosystems such as RSA. Following the same principle used in McEliece's cryptosystem, Gabidulin, Paramonov, and Tretjakov [4] proposed a public-key cryptosystem which uses Gabidulin codes [3] based on the rank metric, referred to as the GPT cryptosystem henceforth. The two groups of attacks against McEliece's cryptosystem and the GPT cryptosystem respectively can both be divided into two categories: structural and decoding attacks. In comparison to McEliece's cryptosystem, the GPT cryptosystem seems to have two advantages: so far it has been shown to be more robust against the decoding attacks than McEliece's cryptosystem and therefore requires much smaller keys than McEliece's cryptosystem.

Nonetheless, the knowledge of the public key in both systems in a sense helps the attacker. If the public key is also kept private in both systems, smaller key sizes are needed to achieve desirable security levels. Of course, the potential applications for the McEliece's cryptosystem or the GPT cryptosystem are different from the appropriate applications for their private-key adaptations. Two private-key adaptations of McEliece's cryptosystem were proposed by Rao [11] and Rao and Nam [13], respectively. The former adaptation, called PRAC, is a direct adaptation of McEliece's system: the keys remain the same while the encryption key is also kept secret. Unfortunately, this system is vulnerable to chosen-plaintext attacks. Struik and van Tilburg [14] have discovered an efficient attack (referred to as the ST Attack) against the latter adaptation. However, Rao argued that, while efficient in theory, the ST attack may not be applicable in practice [12]. Denny [2] proposed to use Reed-Solomon or Preparata codes in the Rao-Nam scheme to further enhance its security.

In this paper, we introduce two private-key adaptations — one direct and the other with an extra permutation matrix — of the GPT cryptosystem, and investigate their robustness against several attacks. We first consider the chosen-plaintext attacks and show that the chosen-plaintext attacks are futile against our new cryptosystems. We also propose a new chosen-plaintext attack, which is effective only for some special cases. Nonetheless, to defeat this new attack, the parameters for the new private-key cryptosystems have to be chosen so as to avoid these special cases. The Struik and van Tilburg (ST) attack is applicable to our new cryptosystems, and the work factors of the ST attack against our new cryptosystem are evaluated. In comparison to the private-key adaptations of McEliece's cryptosystem, our private-key cryptosystems can achieve the same level of security with smaller key sizes. With respect to the three attacks we consider, the use of permutation matrix in the private-key adaptation of the GPT system does not improve the system's robustness.

The rest of the paper is organized as follows. To make this paper self-contained, in Section II we give a brief review of McEliece's cryptosystem, its private-key adaptations, and the ST attack, and also summarize the key components of the GPT cryptosystem. In Section III, we introduce our private-key cryptosystems based on Gabidulin codes, and analyze the attacks against our cryptosystems. Section IV remarks on the feasibility of the chosen-plaintext attacks, and Section V gives some

concluding remarks.

II. Preliminaries

A. McEliece's cryptosystem

McEliece's cryptosystem comprise the following components [8]:

1) *Private key:* The triple $(\mathbf{G}, \mathbf{S}, \mathbf{P})$ where $\mathbf{G}$ is a $k \times n$ generator matrix of a Goppa code, $\mathbf{S}$ is an invertible $k \times k$ random matrix over GF(2) and $\mathbf{P}$ an $n \times n$ permutation matrix.
2) *Public key:* The encryption matrix $\mathbf{E} = \mathbf{SGP}$ and the error-correction capability t of the Goppa code.
3) *Encryption scheme:* Let the plaintext $\mathbf{c}$ be a word of length k over GF(2). The corresponding ciphertext $\mathbf{y}$ is given by $\mathbf{y} = \mathbf{cE} + \mathbf{z}$, where $\mathbf{z}$ is a random error vector with length n and Hamming weight t.
4) *Decryption scheme:* The bounded-distance decoding for the Goppa code can be used to find $\mathbf{c}$ from $\mathbf{y}$.

B. Private-key adaptations of McEliece's cryptosystem

Rao [11] first introduced a private-key cryptosystem using error-correcting codes called private-key algebraic-coded cryptosystem (PRAC). PRAC is a direct adaptation of McEliece's system, the only difference between the two is that the encryption matrix $\mathbf{E}$ is also kept secret in PRAC. Unfortunately, this system is vulnerable to chosen-plaintext attacks that are based on majority voting [13].

To remedy this vulnerability, Rao and Nam then proposed an improved adaptation of McEliece's cryptosystem (called the Rao-Nam scheme below) [13]. The improvements are based on two modifications aimed to thwart the chosen-plaintext attacks. One modification is to permute only after the error vector is added to the codeword. That is, $\mathbf{y} = (\mathbf{cSG} + \mathbf{z})\mathbf{P}$. The other modification is that the error vectors are no longer the coset leaders in the standard array of the code. Instead, one error vector is chosen at random from each coset. Since the error vectors have a one-to-one correspondence with the syndromes, syndrome decoding based on standard array must be used in decryption instead of the bounded-distance decoding, which is used in McEliece's cryptosystem and the PRAC. This set of error vectors is also part of the secret key for the Rao-Nam scheme. Since the weight of the errors are no longer bounded, we can choose error vectors with an average Hamming weight $\frac{n}{2}$. Hence, chosen-plaintext attacks based on majority voting are no longer effective.

C. ST attack

The basic idea of any chosen-plaintext attack is to find all the possible encryptions of some well-chosen plaintexts to recover the encryption matrix. More specifically, we first obtain the entire set Y of encryptions using all possible error vectors of a chosen plaintext $\mathbf{c}$:

$$Y \stackrel{\text{def}}{=} \{\mathbf{y} : \mathbf{y} = \text{Encryption}_{\mathbf{z}}(\mathbf{c}), \mathbf{z} \in Z\},$$

where Z denotes the set of all possible error vectors. In their attack, Struik and van Tilburg [14] first construct the graph $\Gamma = (Y, D)$ based on the encryptions for $\mathbf{c}$. The set of vertices in Γ is Y, the set of different encryptions of $\mathbf{c}$, and the set of edges D consists of the differences between two encryptions, i.e., $d(\mathbf{y}_1, \mathbf{y}_2) = \{\mathbf{y}_1 - \mathbf{y}_2 : \mathbf{y}_1, \mathbf{y}_2 \in Y\}$. For each of the plaintexts $\mathbf{c}_i = \mathbf{c} + \mathbf{u}_i$, where $\mathbf{u}_i$ is the vector with a 1 in position i and zeros everywhere else, a similar graph Γ_i can be constructed based all possible encryptions. If we can identify two vertices $\mathbf{y}_i$ and $\mathbf{y}$ in Γ_i and Γ respectively that were obtained using the same error vector, we would have $\mathbf{e}_i = \mathbf{y}_i - \mathbf{y}$ where $\mathbf{e}_i$ is the i-th row of the encryption matrix. The main problem is how to determine if the error vectors are the same. That is, how to find the correspondence between Γ and Γ_i. The number of attempts needed apparently depends on the cardinality of the automorphism group of Γ. The work factor of the ST attack is shown to be $O(knN^2 \log N + kn|Aut(\Gamma)|^k)$ [14] where N is the size of Z.

D. The GPT cryptosystem

The GPT system is analogous to McEliece's cryptosystem. Instead of using codes which have a decoding algorithm in the Hamming metric, the GPT cryptosystem uses Gabidulin codes which have an efficient decoding algorithm in the rank metric. Unfortunately, these codes are very structured, so it is necessary to break their structure by using a distortion matrix, which is described below. The following are the main components of the system:

1) *Private key*: The triple $(\mathbf{G}, \mathbf{S}, \mathbf{X})$, where $\mathbf{G}$ is a $k \times n$ generator matrix of a Gabidulin code over GF(q^m), $\mathbf{S}$ is an invertible $k \times k$ random matrix over GF(q^m) and $\mathbf{X}$ is a $k \times n$ matrix over GF(q^m) such that $\text{rank}(\mathbf{cX}) \leq t_1, \forall \mathbf{c} = (c_1, c_2, \ldots, c_k) \in GF(q^m)^k$, where $t_1 < t$ is called the *distortion parameter*. $\mathbf{X}$ is called the *distortion matrix* with distortion parameter t_1
2) *Public key*: $\mathbf{E} = \mathbf{SG} + \mathbf{X}$ and the distortion parameter t_1.
3) *Encryption scheme*: Let the plaintext $\mathbf{c}$ be a word of length k over GF(q^m). The corresponding ci-

phertext $\mathbf{y}$ is given by $\mathbf{y} = \mathbf{cE} + \mathbf{z}$, where $\mathbf{z}$ is a random error vector with length n and rank $(t-t_1)$.

4) *Decryption scheme*: We have $\mathbf{y} = \mathbf{cSG} + (\mathbf{cX} + \mathbf{z})$, where $(\mathbf{cX} + \mathbf{z})$ has a rank less than or equal to t. Using the decoding algorithm of Gabidulin codes, the receiver retrieves the word $\mathbf{c}' = \mathbf{cS}$, and computes $\mathbf{c} = \mathbf{c}'\mathbf{S}^{-1}$.

III. New private-key cryptosystems based on rank metric and possible attacks

A. Private-key adaptations

We now propose two private-key adaptations of the GPT cryptosystem. We first consider a direct adaptation of the GPT system. That is, we keep the same encryption and decryption matrices, the same way of selecting the error patterns (all error vectors have a rank of $t - t_1$), but we do not publish the encryption matrix $\mathbf{E}$.

An alternative is to use a permutation matrix in the encryption as in the Rao-Nam scheme. The encryption matrix then becomes $\mathbf{E} = \mathbf{SGP} + \mathbf{XP}$, where $\mathbf{P}$ is an $(n \times n)$ permutation matrix. Note that cryptosystem with the permutation is equivalent to the one without: the product $\mathbf{GP}$ is now a generator matrix of a different Gabidulin code with the same minimum rank distance d [7], and the $\mathbf{XP}$ matrix is still a valid distortion matrix with parameter t_1. We will see below that indeed using a permutation matrix does not increase the level of security of our scheme against the attacks considered herein.

An attack against a private-key cryptosystem often consists of two steps. The goal of the first step is to discover the encryption matrix, and the second step is used to recover the key or the plaintext. Note that with the knowledge of the encryption matrix, the second step is same as an attack against the corresponding public-key cryptosystem, which is the GPT cryptosystem in our case.

There are two types of attacks against the GPT system: structural attacks and decoding attacks. Gibson [6] and Overbeck [10] proposed structural attacks against the GPT scheme, and Chabaud and Stern [1] and Ourivski and Johansson [9] proposed efficient decoding attacks against the GPT cryptosystem. So far, both types of attacks against the GPT system have exponential complexities. In our paper [5], we already gave a way to choose the optimal distortion parameter against both decoding and structural attacks.

In the following, we focus on the security against the first step of the attack. We discuss the effectiveness of two previously proposed attacks, and propose a new one.

B. Chosen-plaintext attack

The first chosen-plaintext attack was introduced against PRAC. It was based on majority voting, taking advantage of the low weight of the error patterns to find good estimates of the rows of the encryption matrix. Here we show that this attack does not work against our new cryptosystems.

Let $\mathbf{c}_1$ and $\mathbf{c}_2$ be two plaintexts differing in only one position, i.e.,

$$\mathbf{c}_1 - \mathbf{c}_2 = \mathbf{u}_i \quad \text{for } 1 \leq i \leq k$$

then the difference between their ciphertexts is given by

$$\mathbf{y}_1 - \mathbf{y}_2 = \mathbf{e}_i + (\mathbf{z}_1 - \mathbf{z}_2)\mathbf{P}$$

where $\mathbf{e}_i$ is the i-th row of the encryption matrix $\mathbf{E}$. The difference $(\mathbf{z}_1 - \mathbf{z}_2)\mathbf{P}$ has a rank inferior or equal to $2(t - t_1)$ but there is no bound for its Hamming weight. Hence, one cannot estimate $\mathbf{e}_i$ by using majority voting. Note that the permutation matrix $\mathbf{P}$ does not make a difference in the discussion above. Thus, the permutation does not enhance the security against the chosen-plaintext attack based on majority voting.

C. New chosen-plaintext attack against the system

Let us denote the set of error vectors of rank weight $t - t_1$ as Z and the set of encryptions of one message $\mathbf{c}$ as Y. That is, $Z = \{\mathbf{z} : \text{rank}(\mathbf{z}) = t - t_1\}$. We note those ciphertexts as $\mathbf{y}^j = \mathbf{cE} + \mathbf{z}^j\mathbf{P} = \mathbf{y} + \mathbf{z}^j\mathbf{P}$. The condition for our new attack to work is that $N = |Z|$ is not a multiple of the characteristic p of the field. Let us note the remainder of the division of N by p as ν. That is, $\nu \equiv N \pmod{p}$.

The idea behind this attack is that once we recover all the elements in Y, the sum of all the ciphertexts in Y can be used to determine $\mathbf{y}$.

$$\begin{aligned}
\sum_{\mathbf{y}^j \in Y} \mathbf{y}^j &= \sum_{\mathbf{z}^j \in Z} (\mathbf{y} + \mathbf{z}^j\mathbf{P}) \\
&= \nu\mathbf{y} + \sum_{\mathbf{z}^j \in Z} \mathbf{z}^j\mathbf{P} \qquad (1) \\
&= \nu\mathbf{y} + \sum_{\mathbf{z}^j \in Z} \mathbf{z}^j \qquad (2) \\
&= \nu\mathbf{y}. \qquad (3)
\end{aligned}$$

The equality in (2) is due to the fact that a permutation does not change the rank of a vector. The equality in (3) holds because, for any $t - t_1$, the sum of all vectors having a given rank $t - t_1$ is equal to $\mathbf{0}$. If the condition is not satisfied, i.e., if $\nu = 0$, the $\mathbf{y}$ term would disappear and the attack would not be applicable.

Since the set Z is known, ν is also known. We can recover $\mathbf{y}$ using the relation

$$\mathbf{y} = \nu^{-1} \sum_{\mathbf{y}^j \in Y} \mathbf{y}^j.$$

We then repeat the process for any $\mathbf{c}_i = \mathbf{c} + \mathbf{u}_i$. We define as Y_i the set of ciphertexts obtained from $\mathbf{c}_i$. That is, $Y_i = \left\{ \mathbf{y}_i^j : \mathbf{y}_i^j = \mathbf{c}_i\mathbf{E} + \mathbf{z}^j\mathbf{P} = \mathbf{y}_i + \mathbf{z}^j\mathbf{P} \right\}$, where $\mathbf{y}_i = \mathbf{y} + \mathbf{e}_i$. Summing up all the ciphertexts in Y_i, we have

$$\begin{aligned} \sum_{\mathbf{y}_i^j \in Y_i} \mathbf{y}_i^j &= \sum_{\mathbf{z}^j \in Z} (\mathbf{y}_i + \mathbf{z}^j\mathbf{P}) \\ &= \nu\mathbf{y}_i + \sum_{\mathbf{z}^j \in Z} \mathbf{z}^j\mathbf{P} \qquad (4) \\ &= \nu\mathbf{e}_i + \nu\mathbf{y}. \qquad (5) \end{aligned}$$

Thus, $\mathbf{e}_i$ can be obtained by

$$\mathbf{e}_i = \left(\nu^{-1} \sum_{\mathbf{y}_i^j \in Y_i} \mathbf{y}_i^j \right) - \mathbf{y} = \nu^{-1} \left[\sum_{\mathbf{y}_i^j \in Y_i} \mathbf{y}_i^j - \sum_{\mathbf{y}^j \in Y} \mathbf{y}^j \right].$$

We recover the encryption matrix $\mathbf{E}$ by repeating this for $1 \le i \le k$.

Hence, our new attack has the following steps:

1) Encipher a plaintext $\mathbf{c}$ until all the N different ciphertexts $\mathbf{y^1}, \mathbf{y^2}, \ldots, \mathbf{y^N}$ are obtained.
2) For $1 \le i \le k$, for the plaintext $\mathbf{c}_i = \mathbf{c} + \mathbf{u}_i$, obtain all the ciphertexts $\mathbf{y_i^1}, \mathbf{y_i^2}, \ldots, \mathbf{y_i^N}$.
3) For $1 \le i \le k$, recover the i-th row of the encryption matrix by computing

$$\mathbf{e}_i = \nu^{-1} \left[\sum_{\mathbf{y}_i^j \in Y_i} \mathbf{y}_i^j - \sum_{\mathbf{y}^j \in Y} \mathbf{y}^j \right].$$

For our private-key cryptosystem based on Gabidulin codes, the attack is effective only when we use $t - t_1 = 1$. Indeed, the number of vectors of length n and rank r over GF(q^m) is given by [7]

$$R(r) = \left(\prod_{i=0}^{r-1} \frac{(q^m - q^i)}{(q^r - q^i)} \right) \cdot \left(\prod_{i=0}^{r-1} (q^n - q^i) \right). \qquad (6)$$

If $t - t_1 = 1$, we see that there are $R(1) = (q^m - 1)(q^n - 1)$ vectors with rank 1. This number is obviously not a multiple of q, so in this case, $\nu \neq 0$, and hence the condition for our attack is met. On the other hand, when $t - t_1 > 1$, $R(t - t_1)$ becomes a multiple of q, and this new attack can not be performed. From above discussion, we see that when $t - t_1 = 1$, the permutation matrix does not help. Hence, the permutation matrix is inconsequential with respect to our new attack.

In the discussion above, Z is restricted to be the set of error vectors with rank weight $t - t_1$. However, Z can be generalized to be a set of error vectors chosen in other ways. Our new attack is still effective for a new set of error vectors if some conditions are satisfied. Let Z' be a subset of Z and Y' as the set of encryptions of $\mathbf{c}$ using an error pattern from Z'. Suppose that Z' verifies the following conditions:

- $N' = |Z'|$ is not a multiple of p,
- Knowing Z', there is an easy way to identify the ciphertexts of Y' from those of Y.

Then attack detailed above is applicable again. However, the two conditions above can be quite stringent. In particular, permutation in the encryption process will make the second condition almost impossible to satisfy. Thus, in this more general case, permutation improves the security.

D. ST attack

The last attack we consider is the ST attack, which works for any parameter value, and we evaluate the work factors of the ST attack against our private-key adaptations below. The work factor of the ST attack depends on the automorphism group defined in [14], and here we assume that the automorphism group is simply the identity. We are therefore considering the worst-case situation: the work factors obtained under this assumption gives lower bounds for the actual work factors. The authors conjecture that the automorphism group is indeed the identity when $Z = \{\mathbf{z} : \text{rank}(\mathbf{z}) = t - t_1\}$.

It can be shown that the work factor W of the ST attack against our scheme is of the order $O(m^2knN^2 \log N)$. If we choose $n = m$, this becomes $O(m^3kN^2 \log N)$. First, suppose we fix m and t. This leaves us with two choices for k: $k_0 = m - 2t$ or $k_1 = m - 2t - 1$. Obviously, k_0 is greater than k_1, and hence with k_0, the work factor is slightly greater and the data rate is slightly higher. Therefore, we should always use k such that $m - k$ is even. Secondly, N depending on $t - t_1$, we can optimize W by choosing $t_1 = 1$. Note that this does not contradict our results from [5] because in this paper we are interested in the security against chosen-plaintext attacks. When m ranges from 5 to 15, the exponents of W are given in Table I.

Assume that an attack is considered impossible if its work factor is greater than 2^{65} and hence the cryptosystem is secure against this attack, then we see that even small parameters can achieve this security: $m = 8$ and $k = 2$ gives a work factor of $2^{73.6}$. Of course, the data rate in this case is quite low. We can then choose $m = 12$

$m\backslash m-k$	4	6	8
5	30.091	0	0
6	36.243	0	0
7	41.767	63.78	0
8	46.976	73.639	0
9	51.99	82.953	108.19
10	56.867	92.004	121.86
11	61.642	100.89	135.03
12	66.338	109.67	147.97
13	70.971	118.36	160.78
14	75.551	126.99	173.48
15	80.087	135.56	186.11

TABLE I

THE EXPONENT OF THE WORK FACTORS OF THE ST ATTACK

and $k = 8$ for a much higher rate. This shows that we can overcome the ST attack by using small parameters.

IV. NUMBER OF ERROR PATTERNS

Rao [12] argued that if the set of error patterns is large enough, the ST chosen-plaintext attack is infeasible since it does not have access to all ciphertexts corresponding to the chose plaintext. Thus, the number of error patterns gives a rough indication of the robustness against chosen-plaintext attack. In the following, we investigate the numbers of error patterns for our new system.

The number of vectors of length n over GF(q^m) with rank r is given by Equation (6). If we choose $n = m$, the number of vectors with rank r becomes

$$R(r) = \prod_{i=0}^{r-1} \frac{(q^n - q^i)^2}{(q^r - q^i)}.$$

Let us denote the number of vectors with Hamming weight r as $H(r)$. For any $r < n$, we have $R(r) \geq H(r)$. This means that using the cryptosystems based on rank metric can have a much larger number of error patterns than the cryptosystem based on Hamming metric. Hence, our private-key cryptosystem based on rank metric can achieve the same level of security as those based on Hamming metric such as the Rao-Nam scheme with smaller key sizes. If, for example, we choose $q = 2, n = m = 20, k = 14, t = 3, t_1 = 1$, the number of error patterns available is approximately $N = 2^{77.4}$. Denny [2] finds $N = 2^{40}$ for a $(25, 20)$ Reed-Solomon code over GF(2^{16}), which is much smaller. Indeed, using smaller key sizes than the ones used in Denny's system, our cryptosystem has more error patterns and are more secure against chosen-plaintext attacks.

V. CONCLUSION

We have proposed two private-key cryptosystems based on Gabidulin codes, one being direct adaptation of the GPT cryptosystem and the other with an extra permutation matrix, and evaluated their security against several attacks. These private-key cryptosystems achieve a sufficient level of security against these attacks with much smaller key sizes. They also have very large numbers of error vectors, which helps to thwart the ST attack in practice. The two cryptosystems are virtually the same with respect to the attacks considered herein.

REFERENCES

[1] F. Chabaud and J. Stern, "The cryptographic security of the syndrome decoding problem for rank distance codes," in *Advances in Cryptology: ASIACRYPT '96*, LNCS, vol. 1163, pp. 368–381. Springer-Verlag, 1996.

[2] W. F. Denny, "Encryptions using linear and non-linear codes: Implementation and security considerations," Ph.D. dissertation, The Center for Advanced Computer Studies, University of Southwestern Louisiana, Lafayette, 1988.

[3] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Problems on Information Transmission*, vol. 21, no. 1, pp. 1-12, January 1985.

[4] E. M. Gabidulin, A. V. Paramonov and O.V. Tretjakov, "Ideals over a non-commutative ring and their application in cryptology," *LNCS*, vol. 573, 1991, pp. 482-489.

[5] M. Gadouleau and Z. Yan, "Optimal Distortion Parameter for the GPT Public-Key Cryptosystem," *Proceedings of 2005 IEEE Sarnoff Symposium*, pp. 133-136, April 2005, Priceton, NJ.

[6] J. K. Gibson, "The security of the Gabidulin public-key cryptosystem," in *Proceedings of EUROCRYPT '96*, pp. 212-223. 1996

[7] P. Loidreau, "Étude et Optimisation de Cryptosystèmes Clé Publique Fondés sur la Théorie des Codes Correcteurs," Ph.D. Dissertation, École Polytechnique, May 2001. In French.

[8] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," *DSN Progress report*, Jet Propulsion Lab, 1978.

[9] A. V. Ourivski and T. Johansson, "New Technique for decoding codes in the rank metric and its cryptography applications," *Problems on Information Transmission*, vol. 38, no. 3, pp. 237-246, 2002.

[10] R. Overbeck, "Extending Gibson's attacks on the GPT cryptosystem". *Proceedings of the 2005 International Workshop on Coding and Cryptography*, Bergen, Norway, March, 2005.

[11] T. R. N. Rao, "Cryptosystems Using Algebraic Codes". *International Conference on Computer Science and Signal Processing*, Bangalore, India, December 1984.

[12] T. R. N. Rao, "On Struik-Tilburg cryptanalysis of Rao-Nam scheme," *CRYPTO '87*. New York: Springer-Verlag, 1987, pp. 448-459.

[13] T. R. N. Rao and K. H. Nam, "Private-key algebraic cryptosystems," *CRYPTO '86*, pp. 458-461. New York: Springer-Verlag, 1986.

[14] R. Struik and J. van Tilburg, "The Rao-Nam scheme is insecure against a chosen-plaintext attack," *CRYPTO '87*, pp. 445-457. New York: Springer-Verlag, 1987.