Two NP-complete Problems in Coding Theory with an Application in Code Based Cryptography

Christian Wieschebrink
Federal Office for Information Security (BSI)
Godeberger Allee 185-189
53175 Bonn, Germany
Email: christian.wieschebrink@bsi.bund.de

***Abstract*— In this paper it is shown that the reconstruction of a punctured code from a given code is an NP-complete problem. Based on this observation a modification of code-based cryptosystems such as the Niederreiter scheme is suggested. In particular the modification is resistant to the Sidelnikov-Shestakov attack. Some other possible attacks are reviewed, and secure key parameters are estimated.**

I. Introduction

There are several public key cryptosystems whose security is based on hard computational problems from coding theory, such as the McEliece [1] and the Niederreiter [2] encryption scheme. More precisely their security is connected to two intractability assumptions: first it is difficult to solve the decoding problem for arbitrary linear codes, second it is difficult to recover the structure (and thus a decoding algorithm) of a certain code from an arbitrary generator matrix. Indeed, the general syndrome decoding problem for binary codes was shown in [3] to be NP-complete. Moreover there is practical evidence, that it is also hard for random instances. All known algorithms tackling this problem (such as [4], [2]) have exponential expected running time.

The hardness of the second problem crucially depends on the class of codes being used in the above schemes. Among others Goppa codes, generalized Reed-Solomon (GRS) codes and Gabidulin codes [5] have been suggested. The use of GRS codes proved to be insecure due to an attack by Sidelnikov and Shestakov [6], which reconstructs the parameters of the code in polynomial time. In contrast the McEliece scheme using Goppa codes (which is one of the oldest public key schemes) remains unbroken, although Goppa codes can be considered as subfield subcodes of certain GRS codes. The security of systems based on Gabidulin codes seems questionable in the light of Overbeck's attack [7].

In the present paper we suggest a general modification to increase the structural security of code based encryption schemes. It is based on a reconstruction problem for punctured codes which we will introduce in Section II. In Section III the modification of the McEliece scheme is presented in detail. Finally, in Section IV we apply this modification to the Niederreiter scheme with GRS-codes, thwarding the attack from [6]. Additionally we analyze some other possible attacks there.

II. Two Problems Concerning Punctured Codes

When saying "code" we always mean a linear code. In the following let F denote a finite field. For $n \in \mathbb{N}$ let I_n be the $n \times n$ identity matrix (over F).

Definition 1: Let C be a code of length n and $S \subset \{1, \ldots, n\} =: N$. By C_S we denote the code which is obtained by deleting all the coordinates of C in $N \backslash S$. C_S is called *punctured code of C in $N \backslash S$*.

We can also say that C_S is the projection of C onto the coordinates given by S.

Definition 2: Let M be a $k \times n-$Matrix over F with columns $m_1, \ldots, m_n$ and $\tau : \{1, \ldots, q\} \to \{1, \ldots, n\}$ an injection ($q \leq n$). The $k \times q-$matrix consisting of the columns $m_{\tau(1)}, \ldots, m_{\tau(q)}$ (in this order) is denoted by M_τ.

Definition 3: Let C, D be codes over the same field and of same length n. C and D are called *equivalent*, if there is a permutation $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that

$$(c_1, \ldots, c_n) \in C \Leftrightarrow (c_{\pi(1)}, \ldots, c_{\pi(n)}) \in D \ .$$

Now consider the following problem: let C be a code of length n and D a code of length m, $m \leq n$. Does there exist a subset $S \subset \{1, \ldots, n\}$ such that C_S and D are equivalent? For our proof of NP-completeness we need a more precise formulation which makes use of the generator matrices of the respective codes:

EQUIVALENT PUNCTURED CODE (EPC)

Given a $k \times n-$Matrix M over F and a $k \times m-$Matrix H over F with $m \leq n$, does there exist a non-singular $k \times k-$Matrix T and an injective map $\tau : \{1, \ldots, m\} \to \{1, \ldots, n\}$ with $(TM)_\tau = TM_\tau = H$?

Proposition 1: EPC is NP-complete for arbitrary F.

Proof: The approach is similar to [3], where NP-completeness of the decoding problem was shown. Namely, the following problem is reduced to EPC:

3-DIMENSIONAL MATCHING (3DM)

Let $N = \{1, \ldots, n\}$ and $K \subset N \times N \times N$. Does K contain a matching, i.e. a subset $K' \subset K$ with $|K'| = n$ such that no two elements of K' agree in any coordinate?

See [8] for a proof of NP-completeness of 3DM.

Clearly, EPC is a problem in NP. Suppose an instance (n, K) of 3DM is given. We can assume $|K| \geq n + 1$ otherwise the problem can be solved trivially in polynomial time. First we construct a $3n \times |K|-$Matrix M_1 as follows.

1-4244-0504-1/06/$20.00 ©2006 IEEE

Authorized licensed use limited to: Universidad Nacional Autonoma De Mexico (UNAM). Downloaded on September 12,2024 at 05:23:14 UTC from IEEE Xplore. Restrictions apply.

For $k := (k_1, k_2, k_3) \in K$ let $v_k \in F^{3n}$ be the column vector with $1 \in F$ in the coordinates $k_1, n + k_2, 2n + k_3$ and $0 \in F$ everywhere else. Now M_1 is the matrix consisting of all v_k. Of course M_1 depends on the order of the k. We assume that some order has been fixed. Note that the columns of M_1 are pairwise distinct. Let M be the matrix obtained from M_1 by appending the $3n \times 3n$ identity matrix twice to the right: $M := (\ M_1 \mid I_{3n} \mid I_{3n}\ )$. Put

$$P := \begin{pmatrix} \frac{I_n}{I_n} \\ \hline I_n \end{pmatrix} . \tag{1}$$

The matrix H is given by $H := (\ P \mid I_{3n} \mid I_{3n}\ )$. M is a $3n \times (|K| + 6n)-$matrix, H is a $3n \times 7n-$matrix; both have full rank. Obviously, M and H can be constructed from (n, K) in polynomial time.

Let us assume that K contains a matching K'. Let M_1' be the matrix consisting of the corresponding columns v_k with $k \in K'$. Then every row of M_1' contains exactly one 1 (and of course each column contains exactly three 1s). So by permuting the rows of M_1' we can obtain matrix P. Thus there exists a permutation matrix T with $TM_1' = P$. Of course T is non-singular. Now we have

$$TM = (\ TM_1 \mid T \mid T\ ) . \tag{2}$$

Every row and every column of T contains exactly one 1. So by changing the order of columns in $(T|T)$ appropriately we obtain the matrix $(I_{3n}|I_{3n})$. This means we can find an injection $\tau : \{1, \ldots, 7n\} \to \{1, \ldots, |K| + 6n\}$ with

$$(TM)_\tau = H . \tag{3}$$

Conversely, assume T and τ with (3) are given. As T is non-singular the columns of TM_1 are pairwise distinct. This means for all $i = 1, \ldots, 3n$: $\tau(n+i) > |K|$ or $\tau(4n+i) > |K|$, because in H columns $n+i$ and $4n+i$ are equal. This implies that every column of I_{3n} occurs in $(T|T)$ (see (2)), thus in T, which means T is a permutation matrix. Thus, all columns of TM_1 have Hamming-weight three, whereas all other columns in TM have weight one. It follows that $\tau(n+i) > |K|$ for all $i = 1, \ldots, 6n$ and $\tau(j) \leq |K|$ for all $j = 1, \ldots, n$. So every column of P occurs in TM_1 or, equivalently, every column of $I' := T^{-1}P$ occurs in M_1. Every row of I' contains exactly one 1 so the columns of I' define a matching in K. ∎

There are ways to restrict EPC while keeping its hardness. For example we allowed the code C_S to be equivalent to D. Does the problem remain hard if we insist on D being *equal* to C_S? As we will see immediately the answer is "yes". First let us rephrase this problem in a more precise way again:

PUNCTURED CODE (PC)

Let M be a $k \times n-$matrix, H a $k \times m-$matrix, $m \leq n$, both over F. Does there exist a non-singular matrix T and a subset $S \subset \{1, \ldots, n\}$ with $|S| = m$ such that $(TM)_S = TM_S = H$?

Proposition 2: PC is NP-complete.

Proof: Again we reduce from 3DM. Let (n, K) be an instance of 3DM and M_1 be defined as in the proof of Proposition 1. Set

$$M := (\ M_1 \mid \mathbf{0} \mid \underbrace{I_{3n} \mid \cdots \mid I_{3n}}_{3n \text{ times}}\ )$$

and $H := (\ P \mid \mathbf{0} \mid I_{3n}\ )$, where P is defined as in (1) and $\mathbf{0}$ denotes the zero column vector. M is a $3n \times (|K| + 1 + 9n^2)-$matrix, and M is a $3n \times (4n+1)-$matrix; both can be computed in polynomial time from (n, K).

Suppose K has a matching. Again, let M_1' be the matrix obtained from M_1 by selecting the columns corresponding to the matching. With the same argument as above there is a permutation matrix T with $TM_1' = P$. We also have

$$TM := (\ TM_1 \mid \mathbf{0} \mid T \mid \cdots \mid T\ ) . \tag{4}$$

Now we have to assemble H by picking the right columns from TM. We have just seen that the first n columns of H can be found in the TM_1 part of TM. There is a zero column in TM, so it is clear where the zero column in H comes from. As T contains exactly one 1 in each row and column we can form I_{3n}, i.e. the last $3n$ columns of H, by choosing the proper columns among the last $9n^2$ columns of TM (one column from each block T). So we see that we can define an appropriate $S \subset \{1, \ldots, n\}$ with

$$(TM)_S = H . \tag{5}$$

Now let T and S with (5) be given. TM contains only one zero column, so $|K|+1 \in S$. All columns in H to the right of $\mathbf{0}$ then must originate from the $(T| \cdots |T)$ submatrix in TM, see (4). This means every column of I_{3n} must occur in T, so T is a permutation matrix. All columns in H to the left of $\mathbf{0}$ can be found in TM_1, so all columns of $T^{-1}P$ occur in M_1 (in the given order). As seen before, this implies the existence of a matching in K. ∎

Let us now take a closer look at PC. We showed the worst-case hardness of this problem, but what can we say about the hardness of average cases? The instances of PC created in the above proof are far from being "average": for example, we only made use of the elements $0, 1 \in F$, and T always is a permutation matrix. So let us consider a random instance of the corresponding search problem. We are given matrices M, H and try to find T, S with the required properties (assuming that T, S exist). How can this be accomplished? The first thing to note is that if either T or S is given, the respective other part of the solution can easily be found in polynomial time. Namely, if S is given we have M_S, and T has to satisfy $TM_S = H$. This yields a system of linear equations over F where the entries of T are unknowns. It can be solved by Gaussian elimination with $O(k^4)$ operations in F.

If T is given, we can compute TM, and by successive comparison of each column of H with the columns of TM S can be found. This can be done with $O(kn)$ operations in F.

Let us get back to the general case. For simplicity first assume that the first k coordinates of the code generated by

Authorized licensed use limited to: Universidad Nacional Autonoma De Mexico (UNAM). Downloaded on September 12,2024 at 05:23:14 UTC from IEEE Xplore. Restrictions apply.

H form an information set, i.e. the matrix H' consisting of the first k columns of H is non-singular. The echelon form of H is given by $J := H'^{-1}H$. We have $(H'^{-1}TM)_S = J$, so the problem is solved, if we can find $T' := H'^{-1}T$. Now apply the following algorithm:

Algorithm 1

for all subsets $V \subset \{1, \ldots, n-(m-k)\}$ with $\#V = k$ **do**
  if M_V is non-singular **then**
    $M' := M_V^{-1}M$
    if all columns of J appear in the given order **then**
      output $T = H'M_V^{-1}$ and the corresponding $S \subset \{1, \ldots, n\}$
      exit
    end if
  end if
end for

In the above algorithm we know that the first k columns of J, which form the identity matrix, must appear in M' so we only have to search for the remaining columns of J. There are $\binom{n-m+k}{k}$ subsets V to check, and in each iteration we need $O(k^3)$ operations to invert M_V and $O(k(n-k))$ operations to check the columns, so we get a complexity of $O(\binom{n-m+k}{k}k(k^2+n-k))$.

If the first k columns of H do not give an information set of the code, we try to find a different set of k columns of H which yield a non-singular matrix. In the above algorithm the subsets V then have to be chosen appropriately. One can show that in this case the complexity stays the same. Presently we are not aware of any faster method to solve PC for random instances.

III. Application to Code Based Cryptosystems

We briefly survey the main features of the McEliece and the Niedrreiter public key cryptosystem (see [5] for a detailed description).

For the private key in both cases Alice chooses a code C over F for which a fast decoding algorithm exists. Let us assume the decoding algorithm is able to correct up to b errors. In the McElice version C is represented by a $k \times n$-generator matrix G' in some standardized form (see Definition 4 below for an example). Alice's public key has the form $G := SG'$, where S is a random non-singular $k \times k-$matrix. If Bob wishes to encrypt a message $m \in F^k$, he randomly chooses a vector $e \in F^k$ of Hamming weight $\leq b$ and computes

$$c := mG + e$$

obtaining the ciphertext c. Alice decrypts c by first applying her secret decoding algorithm getting mS and then multiplying with S^{-1} yielding the cleartext m.

Alice's private key in the Niederreiter scheme is represented by a standardized $(n-k) \times k$-parity check matrix H' for the code C. Let T be a random non-singular $(n-k) \times (n-k)$-matrix. Then the secret key is given by $H := TH'$. This time Bob's plaintext $m \in F^k$ has to be a vector of weight $\leq b$. The ciperthext c is computed by

$$c^t := Hm^t \ . \tag{6}$$

Using the syndrome c as input, Alice can calculate m with the decoding algorithm. It is possible to use the echelon form $[I_{n-k}|J]$ of H in this case (and to modify the decrypting procedure appropriately), so that only J has to be published as public key. In fact, as shown in [9], the McEliece and Niederreiter scheme are equivalent if the same class of codes is used.

Now the structural problem consists in recovering G' (resp. H') from the public key G (H). The computational problems discussed in the last section suggest the following general modification of these cryptosystems to increase the hardness of the structural problem. Again, let G be a $k \times n-$generator matrix of C. Alice now randomly chooses column vectors $c_1, \ldots, c_r \in F^k$ and inserts them into G at random positions. She gets a $k \times (n+r)-$matrix G^*, which she uses as new public key. (The c_i as well as the random positions are kept secret.) Encrypting m to c is done as before:

$$c := mG^* + e \ . \tag{7}$$

The error vector e is of length $n+r$ and weight $\leq b$. Decryption now involves an additional step: suppose, $c_1, \ldots, c_r$ are at positions $p_1, \ldots, p_r$ in G^*. First, Alice deletes the coordinates $p_1, \ldots, p_r$ of c. She gets $c' = mG + e'$ where e' is the corresponding shortened error vector, which of course has weight $\leq b$. Now she uses her decoding algorithm to obtain mG and thereby m.

Because of the equivalence it is possible to turn this cryptosystem into a Niederreiter-like scheme. By linear algebra Alice can compute a $(n+r-k) \times (n+r)-$check matrix H^* for G^*. Bob encrypts as in (6) using H^* instead of H. For decryption Alice computes an arbitrary m' with $c^t = H^*m'^t$ (which can be done easily). Then m' can be written $m' = aG^* + m$ with $a \in F^k$. Applying the decoding procedure above to m' therefore gives the cleartext m.

Let us compare the structural problem of this scheme with the above PC problem. In PC we are given a generator matrix M and a generator matrix H of a shorter code. Our task is to find the set S of coordinates such that M_S and H define the same code. In the above cryptosystem an attacker is given the generator matrix G^* from which he tries to derive the decoder of the hidden code C. The obvious way to do this is to find a coordinate set S such that G^*_S is a generator matrix of C and then derive the decoding algorithm from G^*_S. The attacker does not know a generator matrix of C in advance, so the first step seems to be in a way more difficult than an instance of PC. But even if he knew G the attacker's task would (presumably) remain hard.

It is an important issue how to choose the number r of additional columns. Clearly, if r is too small then using G^* instead of G has no big advantage. However, if r is too large, the cryptosystem becomes susceptible to decoding attacks. The problem is that the code gets longer, but the number of

Authorized licensed use limited to: Universidad Nacional Autonoma De Mexico (UNAM). Downloaded on September 12,2024 at 05:23:14 UTC from IEEE Xplore. Restrictions apply.

errors which can be corrected remains the same. For example, consider the following information set attack. Let c be a cipertext as in (7). Now apply algorithm 2:

Algorithm 2

for all subsets $K \subset \{1, \ldots, n+r\}$ with $\#K = k$ **do**
  if G_K is non-singular **then**
    $L := G_K^{-1}$
    $c_K :=$ projection of c onto the coordinates K
    $e' := c - c_K L$
    if $|e'| \leq b$ **then**
      find m with $mG^* = c - e'$ using Gaussian elimination
      output m and exit
    end if
  end if
end for

In other words: An attacker can try to recover m by guessing k coordinates of c hoping that these k coordinates contain no error. If the $k \times k-$matrix consisting of the corresponding columns of G^* is non-singular, m can be recovered by solving a system of linear equations over F. Now the chance of finding k error-free coodinates is good if the ratio $\frac{b}{n+r}$ is small. So finding the optimum r may require some amount of fine-tuning for each class of codes.

The algorithms given in [10], [4] are refinements of this attack for binary codes (like Goppa codes). However, for codes over larger fields the running time is not improved.

IV. Example: GRS Codes

Let F be a finite field with 2^m elements.

Definition 4: Let $k, n \in \mathbb{N}$, $k \leq n \leq 2^m$, $\alpha = (\alpha_1, \ldots, \alpha_n) \in F^n, c = (c_1, \ldots, c_n) \in (F \backslash \{0\})^n$, where the α_i are pairwise distinct. The *generalized Reed-Solomon code (or GRS code)* $GRS_{n,k}(\alpha, c)$ is a linear code over F given by the generator matrix

$$G = \begin{pmatrix} c_1 & c_2 & \cdots & c_n \\ c_1\alpha_1 & c_2\alpha_2 & \ldots & c_n\alpha_n \\ \vdots & & \ddots & \\ c_1\alpha_1^{k-1} & c_2\alpha_2^{k-1} & \cdots & c_n\alpha_n^{k-1} \end{pmatrix}.$$

Given c and α one can apply a polynomial time algorithm which can decode up to $\lfloor \frac{n-k}{2} \rfloor$ errors. (For details see [5], [11]). As mentioned above GRS codes were considered for use in the Niederreiter scheme, but c and α or parameters yielding the same code can be computed by the Sidelnikov-Shestakov algorithm with $O(k^4 + kn)$ operations in F from an arbitrary generator matrix of $GRS_{n,k}(\alpha, c)$, thereby rendering this class of codes useless for cryptographic purposes. (Later Gabidulin [5] found a version of the Sidelnikov-Shestakov attack with running time $O(n^3)$.) But there may be hope by applying the modification from the last section.

So suppose we insert r random columns $c_1, \ldots, c_r \in F^k$ into G at random positions and multiply the resulting matrix with a non-singular $k \times k-$scramble matrix yielding a $k \times (n+r)-$matrix G^*. We can now review several attacks on the modified cryptosystem:

- *Structural attack 1.* A straightforward strategy is to pick n columns among the $n+r$ columns of G^* and apply the Sidelnikov-Shestakov algorithm to the resulting matrix. If the n selected columns contain none of the added columns c_i, the algorithm will succeed, otherwise this process has to be repeated with a different selection of n columns. After $\binom{n+r}{n}$ iterations in the worst case the parameters of the GRS code are found. Here we can assume that there is only one choice of n columns in G^* which form a generator matrix of a GRS code. This seems reasonable since in general GRS codes form a very small fraction of all linear codes of given length and dimension. So the chance of finding another GRS code in G^* can be neglected.
- *Structural attack 2.* We can improve the above attack by exploiting the fact that a punctured GRS code again is a GRS code. Indeed, if a column in G is removed, the result is a generator matrix of a GRS code of length $n-1$ and dimension k. (Of course removing a column only makes sense as long as $n+1 > k$.) Let the $k \times (n+r)-$matrix G^* with columns $g_1, \ldots, g_{n+r}$ and a small positive integer x be given. (We'll see below, how x has to be chosen.) Apply algorithm 3 below.

Algorithm 3

$N := \{1, \ldots, n+r\}$
for all subsets $S \subset N$ with $\#S = n + x$ **do**
  $K := G_S^*$
  apply the Sidelnikov-Shestakov algorithm to K
  if K generates a GRS code **then**
    $S' := \emptyset$
    for all $i \in N \backslash S$ **do**
      $K := K$ with g_i appended
      apply Sidelnikov-Shestakov to K
      if K generates a GRS code **then**
        $S' := S' \cup \{i\}$
      else $K := K$ with g_i removed
      end if
    end for
    if K has n columns **then**
      output $S \cup S'$ and exit
    end if
  end if
end for

The returned set $S \cup S'$ reveals the positions of columns in G^*, which give the generator matrix of the hidden GRS code. Note that the algorithms also makes use of the fact that a permuted GRS code again is a GRS code. After the algorithms stops the matrices K and G do not necessarily define the same GRS code. In general K and G are only equivalent.

The parameter x has to be chosen such that the probability of a random code of length $k + x$ and dimension

Authorized licensed use limited to: Universidad Nacional Autonoma De Mexico (UNAM). Downloaded on September 12,2024 at 05:23:14 UTC from IEEE Xplore. Restrictions apply.

k being a GRS code is small enough. This rules out the cases $x = 1, 2$. But x shouldn't be too large either, because this worsens the running time as we will see below.

The most time consuming part of algorithm 3 is the search for a suitable S in lines 2–5. The chance of finding a "good" S, which ultimately leads to the right column set $S \cup S'$ is

$$\pi(n, k, r, x) := \frac{\binom{n}{k+x}}{\binom{n+r}{k+x}} x \ .$$

The for-loop in lines 7–14 obviously can be done in polynomial time. The expected running time is much smaller than in the structural attack 1, because we only have to find $k+x$ correct columns instead of n columns.

- *Decoding attack.* Let $f := \lfloor \frac{n-k}{2} \rfloor$ be the number of correctable errors of the GRS code. If we apply the decoding attack from the last section to a ciphertext the chance of finding an information set in a single iteration is

$$\tau(n, k, r) := \frac{\binom{n+r-f}{k}}{\binom{n+r}{k}} \ .$$

Let us try to find parameters such that breaking the above scheme becomes infeasible with the above attacks. Of course structural attack 1 is less important, because structural attack 2 runs much faster. We have to estimate the workfactor for structural attack 2 and the decoding attack. In each iteration of attack 2 we basically have to apply the Sidelnikov-Shestakov algorithm to a $k \times (k + x)-$matrix. Neglecting the running time of lines 7–17 we estimate the minimum number of binary operations for structural attack 2 by

$$w_1 := \frac{(k + x)^3 m}{2 \cdot \pi(n, k, r, x)} \ .$$

(We assume that an arithmetic operation in F requires at least m binary operations.) The decoding attack in algorithm 2 can be implemented in such a way, that the calculation of L requires about $k(n+r-k)$ operations in F, if the L of the iteration before is used. Calculation of e' is done in $k(n + r)$ operations. We ignore the running time of lines 6–8 so that we get an estimated minimum workfactor of

$$w_2 := \frac{(k(n + r) - \frac{1}{2}k^2)m}{\tau(n, k, r)}$$

for the decoding attack. For fixed n we now try to find k, r such that $min(w_1, w_2)$ is maximized. Note that with growing r (and fixed k) w_1 increases, whereas w_2 decreases. As k is bounded, such an optimum pair (k, r) must exist. Table 1 below shows the results for some example lengths n. For reasons of comparability we also present the workfactor of structural attack 1 and of the method described in section II. Note that for this the attacker has to know a generator matrix of the hidden GRS code in advance, so this attack is not applicable in practice. As we can see, cryptographically secure keys of moderate length (using check matrices in standard form as public keys) are within reach.

TABLE I
ESTIMATED WORKFACTORS FOR DIFFERENT PARAMETERS

CODE CHARACTERISTICS				
field	$\mathbb{F}_{128}$	$\mathbb{F}_{256}$	$\mathbb{F}_{512}$	$\mathbb{F}_{512}$
length (n)	128	256	384	512
dimension (k)	79	169	245	335
#random columns (r)	20	39	64	83
$\approx$ #BINARY OPERATIONS				
structural attack 1	2^{100}	2^{184}	2^{285}	2^{367}
structural attack 2 (x=3)	2^{51}	2^{84}	2^{116}	2^{146}
decoding attack	2^{51}	2^{84}	2^{116}	2^{145}
attack w. known GRS code	2^{86}	2^{162}	2^{246}	2^{320}
$\approx$ size of public key (KB)	4.7	20.8	54.6	95.7

V. CONCLUSION

We presented a new generic modification for code based cryptosystems inspired by a NP-complete reconstruction problem for punctured codes. As seen above it may be suitable for fixing previously broken schemes such as the Niederreiter scheme based on GRS codes. Although the modification increases the size of the public key, the new key sizes are still smaller than those needed in cryptosystems using Goppa codes, where a workfactor of 2^{84} requires a public key size of at least 70KB [12] in comparison to 20.8KB in our approach.

REFERENCES

[1] R. McEliece, "A public-key cryptosystem based on algebraic coding theory," *DSN Progress Report, Jet Prop. Lab., California Inst. Tech.*, vol. 42-44, pp. 114–116, 1978.

[2] N. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *Problems of Control and Information Theory*, vol. 15, pp. 159–166, 1986.

[3] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, 1978.

[4] A. Canteaut and F. Chabaud, "A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 367–378, 1988.

[5] E. Gabidulin, "Public-key cryptosystems based on linear codes," 1995. [Online]. Available: http://citeseer.ist.psu.edu/gabidulin95publickey.html

[6] V. Sidelnikov and S. Shestakov, "On insecurity of cryptosystems based on generalized Reed-Solomon codes," *Discrete Math. Appl.*, vol. 2, no. 4, pp. 439–444, 1992.

[7] R. Overbeck, "A new structural attack for GPT and variants," in *Mycrypt 2005*, ser. Lecture Notes in Computer Science, no. 3715. Springer-Verlag, 2005.

[8] M. Garey and D. Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[9] R. Deng, Y. Li, and X. Wang, "On the equivalence of McEliece's and Niederreiter's public-key cryptosystems," *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 271–273, 1994.

[10] E. Brickell and J. Lee, "An observation on the security of McEliece's public-key cryptosystem," in *EUROCRYPT '88*, ser. Lecture Notes in Computer Science, no. 330. Springer-Verlag, 1988, pp. 275–280.

[11] F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. North Holland, 1997.

[12] N. Tomforde, "Eine Sicherheitsanalyse des McEliece Kryptosystems," diploma thesis, University of Greifswald, Federal Office for Information Security (BSI), 2004.

Authorized licensed use limited to: Universidad Nacional Autonoma De Mexico (UNAM). Downloaded on September 12,2024 at 05:23:14 UTC from IEEE Xplore. Restrictions apply.