Function Hiding Based on Error Correcting Codes

Sergio Loureiro[1], Refik Molva
Institut Eurecom, Sophia Antipolis, France
{loureiro, molva}@eurecom.fr

Abstract

This paper presents an original approach to the problem of function hiding based on Error Correcting Codes and evaluates the security of this approach. The novelty of the technique consists in using Error Correcting Codes to hide functions instead of encrypting data vectors. This protocol mainly deals with the issue of secure evaluation of functions in potentially hostile environments.

1: Introduction

With the advent of new computing paradigms like mobile code and ubiquitous computing, the privacy and integrity of software programs become a major concern beyond classical data security considerations. Running a program in a potentially hostile environment may lead to various security requirements, as follows:

- a company might need to prevent the disclosure of certain sensitive algorithms implemented in its software products despite extensive code analysis and reverse engineering by potential intruders including its customers;

- a mobile software agent acting on behalf of a person might need to ensure the integrity of a critical operation performed on an untrusted remote host;

- a data collection agent might need to ensure both the privacy and the integrity of the results computed at various competing sites.

In this paper we suggest a cryptographic mechanism for evaluating a function on an untrusted environment while assuring the confidentiality of the function. The aim of function hiding is twofold:

- algorithm confidentiality, i. e., concealing the internal behaviour of a program;

- integrity of execution, i. e., if an attacker cannot derive the algorithm of the program, then he is unable to find the best way of changing it to his benefit.

The paper is organized as follows: section two deals with the already existing approaches and a definition of autonomous protocol is given. In the next section, a small introduction on cryptosystems based on error-correcting codes is given, with a focus on the importance of the codes used. Section four gives a presentation of an original protocol in order to achieve function confidentiality and discuss its security. Section five focuses on future work and conclusion.

2: Related Work

The problem that is dealt with in this section was also mentioned in the seminal paper by Abadi, Feigenbaum and Kilian [2], which focuses on hiding data from an oracle, or in other words, computing with encrypted data. Based on this idea, Abadi and Feigenbaum [1] developed a protocol to secure circuit evaluation, which allows a player to evaluate his data on another player’s boolean circuit, thereby preserving the confidentiality of his data and also hiding the circuit from the data’s owner. Even though it was originally intended for data confidentiality, this protocol also deals with the problem of function hiding. The major drawback of the protocol is the

[1] Supported by FCT grant PRAXIS XXI/BD/13875/97

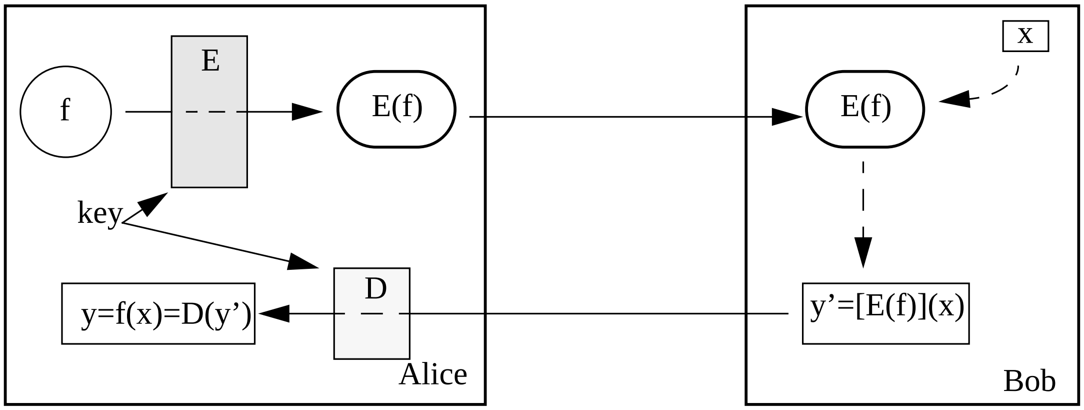

FIGURE 1. Autonomous protocol for function hiding

requirement for a large number of interactions between the two players.

Before, Brassard and Crepeau [6] presented a protocol where Alice could convince Bob of good results achieved by a boolean circuit simulation, without revealing her inputs, but this protocol does not provide circuit hiding.

Recently, Sander and Tschudin [27], [26] defined a function hiding scheme based on an autonomous protocol as depicted in Figure 1. This protocol is autonomous in so far as the interactions between the owner of the function (Alice) and the remote party that evaluates the function (Bob) consist only of the transmission of the function by Alice to Bob and the transmission of the result back to Alice. Unlike the protocol by Abadi and Feigenbaum [1], an autonomous protocol does not involve the exchange of information between the players during function evaluation.

In an autonomous protocol, a function *f* owned by Alice is evaluated by Bob on the input data *x* provided by Bob, while preventing the disclosure of *f* to Bob. The confidentiality of *f* is assured by the transformation *E* that satisfies the following properties:

- it is infeasible to derive *f* from *E(f)* without the knowledge of a secret trapdoor;

- the cleartext result *f(x)* can be derived from the encrypted result *[E(f)](x)* in polynomial time using a secret trapdoor.

Sander and Tschudin [27] illustrated the autonomous protocol concept with a method that allows to encrypt polynomials, based on the Goldwasser-Micali [18] encryption scheme. Therefore, function hiding is achieved, when the functions can be expressed in terms of polynomials.

Another autonomous protocol is described in [4], where a binary decomposition of all possible terms of a polynomial is evaluated, so the cleartext result of the function can be computed by selecting the results corresponding to the components of the function.

In [27], it is also mentioned the possibility of using the so-called composition technique, but no security evaluation is provided. The composition technique consists in multiplying function *f* by a random invertible function.

We suggest an original autonomous protocol based on Error Correcting Codes (ECC) Public Key Cryptosystems (PKC). First of all, a brief overview of ECC cryptosystems is given.

3: Cryptosystems Based on ECC

Cryptosystems based on Error Correcting Codes rely on the difficulty of decoding or finding a minimum weight codeword in a large linear code with no visible structure. These general problems common to coding theory were proven to be NP complete [13] and were used on the public key cryptosystems proposed by McEliece [24], Niederreiter [25] and Gabidulin [14]. These cryptosystems are closely related and one can see the latter two as a derivation of the former.

The McEliece scheme uses a generator matrix and the Niederreiter scheme a parity-check matrix, but they were proven to be equivalent in terms of security for the same parameters [33]. For the same parameters, the Niederreiter cryptosystem reveals some advantages [8], for example, the size of the public key and the number of operations to encrypt.

Generally, the secret key to this kind of public key cryptosystems is the code itself, for which an efficient decoding algorithm is known, and the public key is a transformation of the generator or parity-check matrices. In other words, the efficient decoding algorithm is the trapdoor to the public key transformation.

Gibson [17] demonstrated that there is no advantage of using the Gabidulin cryptosystem over McEliece. Due to an attack developed by Gibson in [17], the size of the code has to be significantly increased with respect to the McEliece scheme in order to achieve equivalent security.

Some identification schemes that take advantage of these problems were also proposed (e. g. [30]).

3.1: Security of Cryptosystems Based on ECC

Despite the general problem of decoding being NP-complete, the best known attacks exploit the properties of linear codes to find a trapdoor, i. e. to recover the structure of the original code or to find an equivalent code. That is usually called a Brickell-like attack [7].

The security of the cryptosystem is highly dependent on the kind of codes used. The initial proposal from Niederreiter used concatenated codes, which were proven to be insecure [28]. Reed-Solomon codes were also proven to be insecure [29].

McEliece proposed Goppa codes that proved to be secure. Nevertheless, Goppa codes generated by a Goppa polynomial which has binary coefficients are also insecure [22].

The properties that a code should have in order to be an eligible candidate for these cryptosystems, which result from the experience gained from successful attacks against this kind of cryptosystems, are the following [8]:

- The type of codes must be large enough to avoid any enumeration;

- An efficient decoding algorithm should exist for this type;

- The generator or parity-check matrix of a transformation of the code must not give any information about its structure.

If the codes obey these rules then the security of the cryptosystems is equivalent to the problem of decoding any linear code.

The rest of the discussion will be focused on McEliece Public Key Cryptosystem, because the majority of existing work was dedicated to this cryptosystem and the use of Goppa codes was adopted because they fulfil these requirements.

3.2: The McEliece Public-Key Cryptosystem

Let C be a q-ary linear code with size n, dimension k and minimum distance d. Let G be a $k \times n$ generator matrix of the code C for which an efficient decoding algorithm exists. The encryption matrix is $E = SGP$, where S is a random $k \times k$ invertible matrix over $GF(q)$ and P is a random $n \times n$ permutation matrix.

Encryption: a plaintext message represented by the vector $x \in GF(q)^k$ is encrypted into the cyphertext y by $y = xE + z$, where z is a randomly chosen error vector that is correctable with the code C ($w(z) \leq \left\lfloor \frac{d-1}{2} \right\rfloor$, where $w(z)$ is the Hamming weight of z).

Decryption: a cyphertext y is decrypted by $yP^{-1} = xSG + zP^{-1}$, and yP^{-1} is decoded with the decoding algorithm of C to retrieve xS (zP^{-1} is correctable since $w(zP^{-1}) = w(z)$). The plaintext x is obtained by $x = (xS)S^{-1}$.

The public key is E and $w(z)$. The secret trapdoor consists of S, P, and the decoding algorithm of C.

3.3: Goppa Codes

The generator matrix is obtained with a polynomial of degree *t*, called a Goppa polynomial, and with a generating vector $\in GF(q)^n$. The decoding of Goppa codewords requires the knowledge of the generating vector and either the weights vector or the Goppa polynomial.

The parameters of a $[n,k,d]$ binary Goppa code are related in the following ways: $n = 2^m$, $d = 2t + 1$ and $k = n - mt$, where *t* is the maximum number of errors the code is able to correct.

This kind of codes fulfils all the properties referenced in Section 3.1. There is a significant number of different Goppa codes, efficient decoding algorithms exist and there is no algorithm to retrieve the characteristic parameters of the code from a permuted generation matrix [9]. For more information on Goppa Codes see [23].

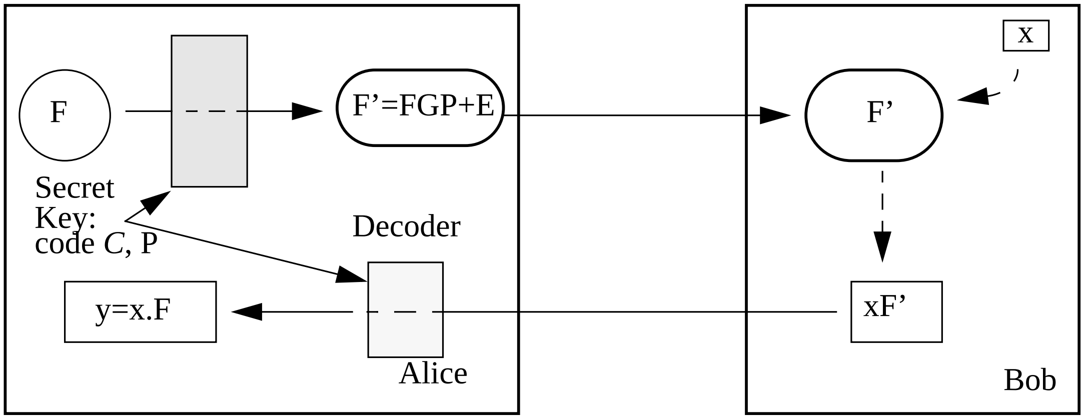

FIGURE 2. Autonomous protocol based on ECC

4: Function Hiding

The original idea presented in this paper consists of hiding a function represented on a matrix format, with a transformation similar to the one used to construct the public key on Error-Correcting Code Public Key Cryptosystems.

Figure 2 depicts the operations performed by the two players of the autonomous protocol using the proposed function hiding scheme as described below.

4.1: Protocol Description

Let G be a generating matrix for an $[n,k,d]$ Goppa code C. Let P be a $n \times n$ random permutation matrix and E a $k \times n$ random matrix, where at least $n - t$ columns consist of the null vector. *G*, *P*, and *E* are kept secret by Alice. Let *F* be a $k \times k$ matrix over Z_2 representing function *f*. Alice computes the encrypted function *F*’ by $F' = FGP + E$ and sends *F*’ to Bob. Bob evaluates *F’* on his data $x \in (Z_2)^k$ by $y' = xF'$ and sends back the result, which is *y’*, to Alice.

Alice decrypts the result $y_1 = y'P^{-1}$, and uses *C*’s secret decoding algorithm to retrieve the cleartext result $y = xF$ from $y = xFG + xEP^{-1}$ (xEP^{-1} is a correctable error vector since $w(xEP^{-1}) \leq t$).

4.2: Cryptoanalysis

The proposed protocol’s security evaluation has been inspired from extensive literature about the cryptoanalysis of the McEliece scheme. Two broad types of attacks have been analysed:

- attacks on the public key, aiming at retrieving the secret key from the public key;

- attacks on the cyphertext, aiming at the disclosure of the plaintext.

The second class of attacks has received much more attention, for example it has been mentioned in [21], [31], [11], [32], [10] and [9] (to cite a few). Furthermore, Berson in [5] proved that it is easy to recover the plaintext if it has been encrypted twice with the same key using the McEliece scheme, and a different error vector.

Nevertheless, in our solution, the attacks on the public key are the sole concern since the function hiding property relies on the difficulty of retrieving the secret function *F* from the public key *F’*. In the sequel of this section we outline the attacks on the public key.

Brute Force attack

The complexity of the brute force attack on the original McEliece public key cryptosystem can be measured by searching exhaustively for all the possible combinations of permutations (*n!*), Goppa codes ($\sim 2^{mt} / t$), and invertible matrices ($\sim 0.29*2^{k^2}$)[23]. In the case of our solution, the complexity of the brute force attack is increased due to the fact that the matrix *F* does not have to be invertible. Using the parameters proposed by McEliece [1024, 524, 50], this attack is obviously not feasible.

Trapdoor attack

The trapdoor attack consists of the analysis of the code structure in order to find an equivalent code.

Heiman [19] was the first to tackle this specific problem and proved that the random matrix *S* used in the original McEliece scheme serves no security purpose concerning the protection of the code, because it does not change the codewords of the original code. The matrix *S* serves the purpose of hiding the systematic structure of the Goppa code matrix *G*, and increasing the number of possible enumerations of the public key.

Adams [3] showed that the likelihood of finding a trapdoor for Goppa codes is small and that there is usually only one trapdoor. The optimum values of the code parameters were also found. Having t=37 and k=654 gives the best result for the same n=1024. This is an important result because it shows that increasing the weight of the error vector could not bring added security.

It was later proved by Gibson [15], that each permutation applied to Goppa codes can be regarded as a possible trapdoor and there are at least *m.n.(n-1)* trapdoors. This results from the fact that no equivalent Goppa polynomials are able to generate equivalent codes. However, this number is still very small when compared with the *n!* possible trapdoors.

The same author in [16] describes an efficient way of obtaining the Goppa polynomial from the public key and from the generating vector, therefore the secret key can only be regarded as the generating vector. The concrete number of trapdoors is still open, but it renders an exhaustive search not feasible, according to [16]. In the author’s opinion the existence of only one trapdoor can, on the other hand, make it easier to find, as in the Gabidulin cryptosystem [17].

In summary, the best known attack on the secret key requires an evaluation of *m.n.(n-1)* trapdoors on an universe of *n!* permutations, which can therefore be considered secure for sufficiently large codes [16].

4.3: Discussion

We reviewed the best known attacks on the McEliece scheme in the previous section, proving that our original utilization of the scheme does not introduce any security breaches and that function hiding relies on the notorious security of the scheme. In short, this conclusion is based on the facts that the matrix representing the function is not relevant for the protection of the code and that the best way to disclose the function is to search for a trapdoor.

As an alternative to the McEliece scheme, the Niederreiter cryptosystem could be used for function hiding by replacing the generator matrix *G* by a parity-check matrix. This would eliminate the error matrix *E* from the function hiding scheme. However, the error matrix *E* used in our scheme enhances the security of the function hiding, in particular against decomposition or brute force attacks. Moreover, the use of matrix *E* as a randomizer is an important security advantage to our scheme over the composition techniques based on the multiplication by random matrices. The other autonomous protocols [27], [4] do not share this advantage either.

One of the disavantages of our protocol is the expansion of the matrix that expresses the function. Nevertheless, this expansion also happens to a higher degree, with the other autonomous protocols previously mentioned. This expansion depends on the size of the code used, which is highly dependent on the number of errors. On the other hand, the encryption and decryption operations are less complex when compared with [27].

The description of our scheme was done for binary codes, like on the McEliece scheme, but can be extended to q-ary linear codes which were proven to be even more secure [20]. Nevertheless, the binary matrix format is suitable for representing boolean functions or circuits. A straightforward way of representing a boolean circuit with a matrix would be to use the truth table directly, similar to the way that it is done in [6].

Unlike the protocol given in [1], our scheme does not assure the confidentiality of the input data *x* with respect to Alice. If the function *F* is invertible, Alice can always interpolate the input data *x* from *xF* and *F.*

On the other hand, confidentiality of the input data *x* with respect to a third party intruder during transmission, can be ensured if Bob adds a correctable random error vector to the result of the computation.

5: Conclusion and Future Work

This paper presented an original approach to the problem of function hiding based on Error Correcting Codes and evaluated the security of this approach.

The novelty of the approach consists of using ECC techniques to hide functions instead of encrypting data vectors. Future work will focus on more efficient representations for boolean functions and the extension of our protocol to a broader class of functions.

The aim of our protocol is to deal with the issue of secure evaluation of functions in potentially hostile environments. Even though the basic purpose of our scheme is confidentiality, the confidentiality of the function can also assure the integrity of its execution. In other words, if an attacker cannot disclose the original function, and if the final result is encrypted, he will not be able to tamper the function to his benefit. On the other hand, it is a step back for host protection due to the fact that the internal behaviour of the code is hidden.

In the future, studies will also focus on classes of codes and transformations which would be more suitable to our protocol and try to apply this protocol to the area of software reliability, specifically checking the results of computations, based on the error detecting capability. Such an approach is, to our knowledge, new to the area of result checking.

6: References

[1] Martin Abadi and Joan Feigenbaum. Secure circuit evaluation. *Journal of Cryptology*, 2(1):1–12, 1990.

[2] Martin Abadi, Joan Feigenbaum, and Joe Kilian. On hiding information from an oracle. *Journal of Computer and System Sciences*, 39(1):21–50, August 1989.

[3] Carlisle M. Adams and Henk Meijer. Security-related comments regarding McEliece's public-key cryptosystem. In Carl Pomerance, editor, *Advances in Cryptology—CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 224–228. Springer-Verlag, 1988, 16–20 August 1987.

[4] Rida A. Bazzi. Secure mobile agents. Technical report, 1998. Arizona State University, Dept. Computer Science and Enginnering.

[5] Thomas A. Berson. Failure of the McEliece public-key cryptosystem under message-resend and related-message attack. In Burton S. Kaliski Jr., editor, *Advances in Cryptology—CRYPTO '97*, volume 1294 of *Lecture Notes in Computer Science*, pages 213–220. Springer-Verlag, 17–21 August 1997.

[6] Gilles Brassard and Claude Crépeau. Zero-knowledge simulation of Boolean circuits. In A. M. Odlyzko, editor, *Advances in Cryptology—CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 223–233. Springer-Verlag, 1987, 11–15 August 1986.

[7] Ernest F. Brickell. Breaking Iterated knapsacks. In G. R. Blakley and David Chaum, editors, *Advances in Cryptology: Proceedings of CRYPTO 84*, volume 196 of *Lecture Notes in Computer Science*, pages 342–358. Springer-Verlag, 1985, 19–22 August 1984.

[8] 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*, 1998, pages 367-378.

[9] A. Canteaut and N. Sendrier. Cryptanalysis of the original mceliece cryptosystem. In *In Advances in Cryptology - ASIACRYPT'98, Lecture Notes in Computer Science. Springer-Verlag*, 1998, pages 187-199.

[10] F. Chabaud and J. Stern. The cryptographic security of the syndrome decoding problem for rank distance codes. In Kwangjo Kim and Tsutomu Matsumoto, editors, *Advances in Cryptology—ASIACRYPT '96*, volume 1163 of *Lecture Notes in Computer Science*, pages 368–381, Kyongju, Korea, 3–7 November 1996. Springer-Verlag.

[11] Florent Chabaud. On the security of some cryptosystems based on error-correcting codes. In Alfredo De Santis, editor, *Advances in Cryptology—EUROCRYPT 94*, volume 950 of *Lecture Notes in Computer Science*, pages 131–139. Springer-Verlag, 1995, 9–12 May 1994.

[12] D. W. Davies, editor. *Advances in Cryptology—EUROCRYPT 91*, volume 547 of *Lecture Notes in Computer Science*. Springer-Verlag, 8–11 April 1991.

[13] R. McEliece E. Berlekamp and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Information Theory*, IT-24(3):384–386, May 1978.

[14] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. Ideals over a non-commutative ring and their application in cryptography. In D. W. Davies, editor. Advances in Cryptology—EUROCRYPT 91, volume 547 of Lecture Notes in Computer Science. Springer-Verlag, 1991, pages 482–489.

[15] J. K. Gibson. Equivalent Goppa codes and trapdoors to McEliece's public key cryptosystem. In D. W. Davies, editor. Advances in Cryptology—EUROCRYPT 91, volume 547 of Lecture Notes in Computer Science. Springer-Verlag, 1991, pages 517–521.

[16] Keith Gibson. *Algebraic Coded Cryptosystems*. PhD

thesis, University of London- Royal Holloway and Bedford New College, 1995.

[17] Keith Gibson. The security of the Gabidulin public key cryptosystem. In Ueli Maurer, editor, *Advances in Cryptology—EUROCRYPT 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 212–223. Springer-Verlag, 12–16 May 1996.

[18] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, April 1984.

[19] R. Heiman. On the security of cryptosystems based on error correcting codes. Master's thesis, Feinberg Graduate School of the Weizman Institute of Science, 1987.

[20] Heeralal Janwa and Oscar Moreno. McEliece public key cryptosystems using algebraic-geometric codes. *Designs, Codes and Cryptography*, 8(3):293–307, June 1996.

[21] P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In Christoph G. Günther, editor, *Advances in Cryptology—EUROCRYPT 88*, volume 330 of *Lecture Notes in Computer Science*, pages 275–280. Springer-Verlag, 25–27 May 1988.

[22] P. Loidreau. Some weak keys in McEliece public-key cryptosystem. In *IEEE International Symposium on Information Theory, ISIT'98, Boston, USA*, 1998.

[23] F. MacWilliams and N. Sloane. *The Theory of Error-Correcting Codes*. North-Holland, 1977.

[24] R. McEliece. A public-key cryptosystem based on algebraic coding theory. In *Jet Propulsion Lab. DSN Progress Report*, 1978.

[25] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. In *Probl. Contr. and Information Theory*, vol. 15, pages 159-166, 1986.

[26] Tomas Sander and Christian Tschudin. On software protection via function hiding. In *Proceedings of the Second Workshop on Information Hiding*, Portland, Oregon, USA, April 1998.

[27] Tomas Sander and Christian Tschudin. Towards mobile cryptography. In *Proceedings of the 1998 IEEE Symposium on Security and Privacy*, Oakland, California, May 1998.

[28] Nicolas Sendrier. On the structure of a randomly permuted concatenated code. In *EUROCODE 94*, pages 169–173, Abbaye de la BussiËre sur Ouche, France, October 1994.

[29] V. Sidelnikov and S. Shestakov. On cryptosystems based on generalized reed-solomon codes. *Diskret. Mat.*, 4:57–63, 1992.

[30] Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *Advances in Cryptology—CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21. Springer-Verlag, 22–26 August 1993.

[31] Johan van Tilburg. On the McEliece public-key cryptosystem. In S. Goldwasser, editor, *Advances in Cryptology—CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 119–131. Springer-Verlag, 1990, 21–25 August 1988.

[32] Johan van Tilburg. *Security-Analysis of a Class of Cryptosystems Based on Linear Error-Correcting Codes*. PhD thesis, Technische Universiteit Eindhoven, 1994.

[33] Robert H. Deng Yuan Xing Li and Xin Mei Wang. On the equivalence of mceliece's and niederreiter's public key cryptosystems. *IEEE Trans. on Information Theory*, 40(1):271–273, January 1994.