

Cryptcoding - Encryption and Error-Correction Coding in a Single Step

Danilo Gligoroski¹ and Svein J. Knapskog¹ and Suzana Andova²

¹ Centre for Quantifiable Quality of Service in Communication Systems, Norwegian University of Science and Technology, O.S.Bragstads plass 2E, N-7491 Trondheim, NORWAY

gligoroski@yahoo.com, Svein.J.Knapskog@Q2S.ntnu.no

² Department for Telematics, Norwegian University of Science and Technology, O.S.Bragstads plass 2E, N-7491 Trondheim, NORWAY
suzana@item.ntnu.no

Abstract. *In this paper we re-open a 25 years old question of joint encryption and error-correction coding, named here as Cryptcoding. Cryptcoding is a procedure in which encryption/decryption and error-correction coding/decoding are performed in a single step. We discuss the advantages of this approach over the traditional First-Encrypt-Then-Encode approach. To our knowledge only three different realizations of cryptcoding can be found in literature: the McEliece's public key encryption system using Goppa codes, the Kak's joint encryption and error-correction coding using D-sequences and the recently developed quasigroup random error-correcting codes. We briefly discuss the first two and mainly focus on the last one. We give two examples in which cryptcoding is efficiently employed for secure document management.*³

Key words: cryptography, error-correction, secure document management.

1 Introduction

Contemporary cryptography and coding theory have more than 60 years of a successful history. Since the publication of the seminal works of Shannon in 1949 [1] and Hamming [2], these two scientific fields evolved with mutual borrowing of the cross fertilizing ideas (see for example [3]), but keeping clear distinction of the developed methods and algorithms. That is partly because cryptographic and error-correcting algorithms have opposite intentions with information processing they perform. While cryptographic algorithms in order to provide information security, in the process of decryption need an errorless input, error-correcting algorithms are meant to handle certain amount of errors in the input data, but they are not designed to provide any security of the data they process. However, there are many situations when both information security and error-correction are needed or required. In that case a combination of cryptographic algorithms and error-correcting codes, which basically means encryption, decryption, encoding and decoding, is realized by a sequential execution of two separate algorithms, one for encryption/decryption and the other one for coding/decoding.

In 1997, the National Scientific Foundation (NSF) of USA recognized the needs to bring together the two research communities in order to address mutual problems, challenges, future development, research and practice of cryptography and codes. A working group of some of the most prominent researchers from both fields was established. The report of their work [4] addresses important aspects both from an error-correcting

³ This work was carried out during the tenure of an ERCIM fellowships of Suzana Andova visiting Department for Telematics at Norwegian University of Science and Technology - Trondheim, Norway.

and from a cryptographic perspective, and gives several examples of scenarios where the development and methods from one field contributed to the development of the other. However, they took the general and widely accepted view, that coding techniques are mainly used for ensuring that encrypted data are stored (or transmitted) without errors. In other words, they did not consider any possibility of combining cryptography and error-correcting techniques in a single function.

In this paper we address the question of joining the two primitives into a single one, namely, whether it is possible to define a primitive (and algorithm) which at the same time performs the two functions: encryption and encoding. We will call such a primitive *cryptcoding*. Interestingly, more than 25 years ago McEliece tackled this question for the first time [5]. Since then, unfortunately, only few authors have tried to deal with this problem. Some authors argue that a possible reason the cryptcoding has not attracted more attention is the fact that error correction introduces data redundancy i.e. data expansion, which is usually not desirable in cryptography (e.g. [6]). However, when encrypted data is transmitted over a public network, a redundancy have to be introduced. In that situation the combined computational costs of First-Encryption-Then-Coding are usually bigger then those required by a cryptcoding procedure.

In addition, we discuss another advantage that one cryptcoding system may poses. Namely, the recently proposed cryptcoding system [7] has an arbitrarily chosen redundant information. More precisely, the system is defined in a way that the redundant information used for error-correction is not pre-determined by the nature of the error-correction part of the algorithm but it can be chosen arbitrarily out of the whole set of possible strings. We show how this property can be used for defining different schemes for secure management of digitally produced documents.

The paper is organized as follows. In Section 2 we discuss the traditional approach of composing encryption and encoding sequentially. In Section 3 we introduce the notion of cryptcoding and briefly discuss several cryptcoding systems that can be found in the literature. The Section 4 contains several examples of secure schemes for management of digitally produced document defined by means of cryptcoding. Finally, we conclude the paper with Section 5.

2 The traditional encryption-then-coding approach

Building protocols that deal with errors in presence of cryptographic operations is not new. A combination of these two techniques is used for various purposes, such as for use and protection of biometric data [8–13], for cryptographic protection and authentication of digitally produced documents [14–17], as well as in image coding [18]. As a matter of fact, encryption and error-correction are inseparable parts of any protocol for secure communication via a noisy channel.

While each of these methods addresses different problems, they all have the following in common: they all comprise both encryption and error-correction coding as separate procedures. The two procedures are executed sequentially, first the input message is encrypted into an encrypted output, and then a redundancy is introduced by the error-correction algorithm. The composition of the two procedures results in an encrypted message which is error resistant up to a certain degree. The inverse operations, decoding and decryption are done in two separate steps as well, in a reverse order, first decoding then decryption. Informally, the composition of these two algorithms can be described by the diagram given in Figure 1.

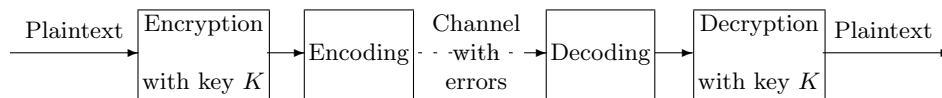


Fig. 1. An illustration of the traditional approach - First-Encryption-Then-Coding.

Using the standard terminology from cryptography and coding theory, the diagram in Figure 1 can be described by four algorithms: *Encr* and *Decr* which denote encryption and decryption algorithms, and *Enco* and *Deco* which denote encoding and decoding algorithms.

Then the traditional First-Encryption-Then-Coding approach can be described in the following way. For a given message M and an encryption key K :

1. $Encr(K, M) = C_1$: Encryption step in which ciphertext C_1 is produced;

2. $Enco(C_1) = C_2$: Encoding step in which the ciphertext C_1 is encoded;
3. $Deco(C_2) = C_1$: Decoding step in which the ciphertext C_1 is restored after possible errors that occurred during transmission;
4. $Decr(K, C_1) = M$: Decrypting step in which the initial message is obtained by decryption of C_1 .

$(Encr, Decr)$ together with a given key-generator define a secure encryption scheme, and as such they satisfy the following two conditions:

Correctness condition: For all messages M and for all keys K , $Decr(K, Encr(K, M)) = M$.

Security condition: For every message M it is computationally infeasible to distinguish between $Encr(K, M)$ and a message from uniform random source, for an arbitrary encryption key K .

$(Enco, Deco)$ defines an error-correction scheme. Thus, the following condition is satisfied:

Coding: For every message $C_2 = Enco(C_1)$ there exists a positive natural number $t > 0$ such that for every C' which has Hamming distance from C_2 less or equal then t (i.e. $Hamming(C_2, C') \leq t$), $Deco(C') = C_1$.

3 Cryptcoding - Cryptographic error-correcting approach

As argued in the previous section, there are enormous number of problems in which encryption and error-correction are combined, but still very little attention has been given to a possibility of combining both features in one single primitive. As a matter of fact, the first attempts to deal with this question were done almost three decades ago [5, 20, 19]. Since then, it seems (almost) nobody has even questioned whether it is necessary to use two separate schemes, and consequently have higher computational costs.

Mainly there are three reasons to investigate the question of cryptcoding:

- We find that the question of combining the two primitives into a single one is of fundamental importance, theoretical as well as practical.
- Recently a completely novel error-correction algorithm has been proposed [7] which may provide an efficient solution of the problem we are posing here.
- As already mentioned, we believe that cryptographic error-correcting algorithms will significantly reduce computational costs and increase efficiency, especially in practical realizations.

A *cryptcoding* is a method that performs both, the functions of encryption and error-correction coding as a single cryptographic primitive. It consist of a pair of algorithms $(\mathcal{E}, \mathcal{D})$ where \mathcal{E} is called *encryptcoding* algorithm and \mathcal{D} is called *decryptcoding* algorithm. $(\mathcal{E}, \mathcal{D})$ satisfy the following conditions:

Encryption: $(\mathcal{E}, \mathcal{D})$ is an secure encryption scheme, which means it satisfies the correctness and the security conditions as given in Section 2

Encoding: $(\mathcal{E}, \mathcal{D})$ is an error-correction scheme and satisfies the coding condition given in Section 2.

A direct consequence of the above requirements is that *cryptcoding* is different than traditional composition of *encryption-then-encoding*. Our decision to introduce the new term *cryptcoding* is justified with the ability of this approach to accomplish both, the functions of secure encryption and error-correction in a single logical operation, exactly as described by the two conditions above.

The cryptcoding is illustrated in Figure 2. If we use \mathcal{E} and \mathcal{D} to denote the encryptcoding and decryptcoding algorithm, respectively, we have that the cryptcoding is realized in only two steps:

1. $\mathcal{E}(K, M) = C$: Encryptcoding step in which an error resistant ciphertext C is produced using the encryption key K .
2. $\mathcal{D}(K, C') = M$: Decryptcoding step in which the original message M is restored from the erroneous ciphertext C' after transmission ($Hamming(C, C') \leq t$).

In what follows we briefly discuss the cryptcoding system of McEliece from 1978 [5], and the Kak's proposal from 1983 [20].

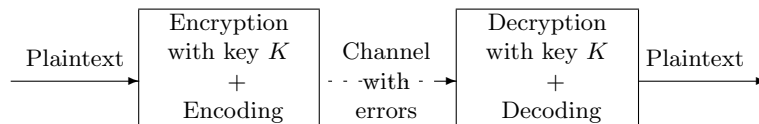


Fig. 2. An illustration of Cryptocoding approach.

McEliece's algebraic based cryptosystem. In [5] McEliece suggested that error correcting codes are excellent candidates for providing public key cryptosystems. He defined a cryptosystem based on Goppa codes, a subset of polynomial codes that are easy implementable and for which exists a fast decoding algorithm. The encoding algorithm multiply a message vector onto a so-called generator matrix G to form the codeword which is transmitted via a noisy channel. McEliece suggested a special form of a generator matrix, G' , which in his cryptosystem is used as a public key (so-called public generator matrix). Thus, if G is a generator matrix from the encoding algorithm, then $G' = SG P$ where S is a random dense non-singular matrix and P is a random permutation matrix. Defined in this way, G' generates a linear code with the same rate and minimum distance as the code generated by G . Thus, G' is used for encryption and P^{-1} and S^{-1} are used to produce the codeword and for decryption.

In [19] Rao proposed several modifications of the McEliece public-key cryptosystem. First he gave a symmetric variant of the McEliece cryptosystem and then he discussed some possibilities for efficiency improvement that do not decrease the security properties.

Kak's cryptocoding method. In [20] Kak proposed a different approach to the problem of joint encryption and error-correction coding. His solution is based on D-sequences, which are decimal expansions of fractions. He proved that the considered encryption operation was equivalent to the operation of exponentiation in finite field, which lies in the core of the definitions of several public-key schemes. In the paper he also pointed out several directions for further research in the area of joint encryption and error-correction coding. Unfortunately, his work did not attract any further attention.

Quasigroup random error-correcting codes. Recently in [21] a new cryptocoding method has been proposed which by its features differs from any other previously defined method.

The method is based on a so-called technique of *quasigroup string transformation*. It is parameterized by a quasigroup (Latin square) of order 16, $(Q, *)$, that is used when a pair of an encryptcoding and a decryptcoding functions $(\mathcal{E}, \mathcal{D})$ is generated. We stress that $(Q, *)$ is chosen out of at least 2^{430} possibilities. One of the most important properties of this systems which makes it to have an advantage over the other previously described cryptocoding systems is the possibility to choose an arbitrary redundant information. Here we give only a very brief introduction to this cryptocoding system. For the precise definitions and more details we refer to [7, 21, 22].

The encryptcoding \mathcal{E} and the decryptcoding \mathcal{D} functions are defined as:

$$\mathcal{E} : Q^N \times Q^m \rightarrow Q^{m+r},$$

and

$$\mathcal{D} : Q^N \times Q^{m+r} \rightarrow Q^m,$$

where Q is a finite alphabet, $N \in \mathbb{N}$ is the (fixed) length of the initial secret key string K , m is the length of the message M , r is the length of the redundant message M_R and $R = \frac{m}{m+r}$ is the rate of the error-correction.

The pair $(\mathcal{E}, \mathcal{D})$ has the following properties:

1. (Invertibility of \mathcal{E} and \mathcal{D}) For every key-string $K \in Q^N$ and every message string $M \in Q^m$,

$$\mathcal{D}(K, \mathcal{E}(K, M, M_R), M_R) = M.$$

2. (Cryptographic properties of \mathcal{E} and \mathcal{D}) If the key K of length N is not known to the adversary, then under the adaptive chosen attack, the minimum number of computing operations needed for recovering the message M from message $C = \mathcal{E}(K, M)$ is $O(|Q|^N)$ i.e. exponential on the length of the key.
3. (Error-correcting properties of \mathcal{E} and \mathcal{D}) There exist positive natural number $t > 0$, such that for every string C' that is within Hamming distance t from the string $C = \mathcal{E}(K, M, M_R)$ (i.e. $\text{Hamming}(C, C') \leq t$) $\mathcal{D}(K, C', M_R) = M$.

In their original proposal, Gligoroski, Markovski and Kocarev use a redundant information that consists only of zeros, i.e. $M_R = 00 \dots 0$. However, the redundant information can be arbitrary. In the next section we will show that this freedom of choice of the redundant information M_R makes their system a powerful tool for solving problems related to secure document management, commitment schemes, time stamps, tamper resistance, and many others. Moreover, in the light of the remarks given in Section 1 about the preferences encrypting algorithms not to manifest data expansion, their algorithm can be used also as an ordinary encryption scheme by taking the empty string as a redundant message.

4 Cryptcoding in secure document management

In this section we give two examples of a secure document management. These examples represent a real-life problems which are usually solved with some other techniques. Here we propose solutions that use properties of the cryptcoding system of [21].

Before we give the definitions of the concrete schemes, we give a general framework in which they will be defined. Its design is based and uses the parameters of the quasigroup random error-correcting cryptcoding system. As a matter of fact, this framework provides a possibility many other security schemes to be defined.

Our definition uses a trusted third party (TTP) denoted by \mathcal{N} (in real life it can be a notary, lawyer, certificate authority, bank, some governmental organization, or any other institution). He guarantees that the content of a document certified by him, is authentic and identical to the originally signed document by two clients Alice and Bob.

\mathcal{N} generates (or possesses) an algebraic structure $\mathcal{Q}_{\mathcal{N}} = (Q, *)$ over the set Q , which has a role of a key. \mathcal{N} also uses a unique counter *Counter* (equivalent to the conventional archive number) for every processed document. Alice and Bob both have some keys K_A and K_B (which are strings of characters from the set Q). Each of these three keys, $\mathcal{Q}_{\mathcal{N}}$, K_A and K_B can be given publicly or kept as a secret. We assume that the *Counter* is always public. The *key of the document*, K is produced as a cryptographic hash digest of the values: *Counter*, $\mathcal{Q}_{\mathcal{N}}$, K_A and K_B , that is,

$$K = Hash(\mathcal{Q}_{\mathcal{N}}, Counter, K_A, K_B).$$

This is done by the TTP \mathcal{N} . As usual, if K_A or K_B or any other data is a secret, then the owner of that key can either share this key with the TTP, or she/he maps the content of the secret by some hash function, and provides the result of the hashing.

Basically, the key K serves as a symmetric encryption key. To complete the operations of encryptcoding and decryptcoding functions, the redundant information M_R have to be provided. As already mentioned, in this method a redundant information can be freely chosen. Thus, it may contain different statements, commitments or any other meaningful information (names, dates, bank account numbers, car plates number, etc.). The content of M_R can be chosen by one or more parties involved in the process and can be given publicly or it can be in a private possession of \mathcal{N} , Alice and/or Bob. We assume that, later in the process of verification, it may be revealed to other involved parties if necessary.

Depending on the initial knowledge of all parties (“who knows what”) different schemes can be defined. For the purposes of this paper we consider only two of them, and give real-life examples where they can be applied. The reader can easily construct the other schemes giving a different initial knowledge to the involved parties.

In the following example we use two instead of three parties. It is easy to observe that if all information provided by Bob, for instance, are public, then the problem reduces to a two-party problem.

Example 1. (Parking ticket) In this example the two involved parties are Alice, who purchases a yearly parking ticket for a parking place in the downtown, on one side, and the authority \mathcal{N} who issues the ticket on the other side. If the ticket is just a paper document, there is a trivial possibility for Alice to make a forgery. Namely, she can print another ticket with similar shape but with different (extended) date. The authorities can prevent the forgery by use of a communication network. A parking police officer can check the validity of every issued ticket by contacting the central database and comparing the data in the database with those on the ticket. Of course, this solution is expensive in terms of communication costs, speed and time.

We define the following scheme which will be used for solving this problem. To issue parking tickets, the authority \mathcal{N} possesses a parking ticket key $\mathcal{Q}_{\mathcal{N}}$. The Alice's key K_A is public. The (redundant) information M_R can be either public (it is known to Alice and the authority, for instance, the unique identification number of Alice, her date of birth, etc.) or it can be a secret of the authority (Alice does not know the value of M_R though she may know some data contained in it, her date of birth for instance). Under these assumption we can conclude that: 1.) The authority \mathcal{N} has guarantees that the content of the document (the ticket in this example) cannot be changed. Thus, Alice cannot forge the ticket; 2.) Both, Alice and the authority \mathcal{N} have guarantees that the document can be stored securely and that it can be recovered in case of (a certain degree of) intentional or non-intentional errors. For instance, if Alice loses the ticket the same ticket can be issued again. Or if the ticket gets damaged the information can be restored without looking in the central base as we will see.

With this setting, Alice gets a ticket in which the information about its expiration date, together with the information about the licence plates of the car are printed as a paper document. This information appears in both forms, disclosed (that Alice is able to read) and encrypted (that a parking police officer can read and verify the information on the ticket). The encrypted information can be printed in a data-dense format such as a 2D barcode, for instance PDF417 [23]. In order to be able to read the encrypted data, the parking police officer needs a 2D barcode reader. He only needs to read the barcode (encrypted data) and compare the disclosed (original) information on the ticket with the information on the display of his barcode reader. There is no need to communicate the central database in order to check the validity of the ticket.

Obviously, in this scenario the method is used for three purposes: 1.) as an encryption algorithm, 2.) as an error-correcting algorithm 3.) as a method for digital signing of paper documents. In the literature similar solutions can be found (e.g [24]) but they all use separate algorithms for all three steps. As our method can perform all three functions in a single step, we argue that it offers much better solution, particularly if it is implemented in a combined hardware-software solution as a barcode reader.

Example 2. (Non-disclosed testament) Before we state the example, we define the used scheme. We assume that the authority key $\mathcal{Q}_{\mathcal{N}}$ is public, the Alice's key K_A is secret, the Bob's key K_B is public, the content of M_R is known to Alice and to be given later to Bob. In this scenario, only Alice can prove the authenticity of the document. Any change she may make in the document, can be discovered by Bob and \mathcal{N} together, even if they have only the encrypted version of the document. Bob can not change the content of the document. If Bob is given only the encrypted version of the document he will be able to produce the original document when he gets M_R . Alice and Bob together decide in which format the document is stored, in its disclosed (original) or in its encrypted version.

Now we place Alice and Bob in the following real-life situation. Alice wants to write a testament to declare she passes all her fortune to her son Bob after her death. Of course, the Alice's attorney is playing the role of the authority. Alice takes certain precautions and she does not want Bob (maybe even her attorney) to know the content of the testament before she dies. Thus she rather keeps a copy of the document in her safe, but only in an encrypted form. As such, nobody will be able to reproduce the document or even parts of the document since she is the only one who knows the secret information. Moreover, some intentional or non-intentional changes of the encrypted document can be tolerated and the proper decryption can be done. Thus, Alice is guaranteed that neither Bob nor her attorney (nor both of them together) can make changes in her testament. Alice can produce a letter in which she declares some commitments and statements from which the redundant information M_R can be reproduced. Besides, M_R may contain information from which the attorney can check the identity of Bob after Alice passes away.

In the literature one can find a solution of this problem based on commitment schemes. Those solutions requires additional encryption algorithm, and possibly encoding algorithm. Again, our method provides a solution which uses only one single algorithm.

5 Conclusion

In this paper we defined a notion of *cryptcoding*, a cryptographic primitive which joins together, encryption and error-correction. We gave a short overview of the very few cryptcoding systems that can be found in the literature. We focused on the recently developed cryptcoding system based on quasigroup string transformation. We emphasized the advantages of this method and used them to design security schemes

which can be used for secure document management. We showed how these schemes can be used in real-life situations. In our opinion, cryptocoding can significantly increase performances, such as communication costs, speed, security of digital and paper documents. A deeper investigation on this subject as well as the formal definition of the schemes described in this paper is an ongoing work.

References

1. C.E. Shannon, *A Mathematical Theory of Communication*, Bell Sys. Tech. J., 27:379–423, 623–656, 1948.
2. R.W. Hamming, *Error detecting and error correcting codes*, Bell Sys. Tech. J., 29:147–60, 1950.
3. J.L. Massey, *Some Applications of Coding Theory in Cryptography*, in *Codes and Cyphers: Cryptography and Coding IV* (Ed. P. G. Farrell). Essex, England: Formara Ltd., 1995, pp. 33–47.
4. *Report of the Working Group on Cryptology and Coding Theory*, National Science Foundation, April 17–18, 1997, available on <http://www.nsf.gov/pubs/1998/nsf9814/nsf9814.htm>
5. R. J. McEliece, *A public-key cryptosystem based on algebraic coding theory*, Deep Space Network Progress Report, Nos. 42-44, Jet Propulsion Labs, Pasadena, CA. 1978, pp. 114-116.
6. J. Seberry and J. Pieprzyk, *Cryptography: An Introduction to Computer Security*, ISBN 0-13-194986-1, Prentice Hall, 1989.
7. D. Gligoroski, S. Markovski, L. Kocarev, *New Directions in Coding: From Statistical Physics to Quasigroup String Transformations*, International Symposium on Nonlinear Theory and its Applications, NOLTA2004, Fukuoka, Japan, November 29 - December 3, 2004.
8. L.A. Ray, R.N. Ellson, *Method and Apparatus for Credit Card Verification*, U.S. Patent 5,321,751, June 1994.
9. A. Juels, M. Wattenberg, *A Fuzzy Commitment Scheme*, In G. Tsudik, ed., Sixth ACM Conference on Computer and Communications Security, pp 28–36, ACM Press. 1999.
10. Y. Dodis, L. Reyzin, A. Smith, *Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data*, Eurocrypt 2004 LNCS 3027, pg. 523–540, 2004.
11. Y. Dodis, A. Smith, *Correcting errors without leaking partial information*, STOC '05: Proc. of the 37 ACM symposium on Theory of computing, 2005.
12. G. Di Crescenzo, R. Graveman, R. Ge, G. Arce, *Approximate Message Authentication and Biometric Entity Authentication*, Proc. 9th Int. Conf. Financial Cryptography and Data Security, FC2005, LNCS 3570.
13. E.C. Chang, Q. Li, *Small Secure Sketch for Point-Set Difference*, Cryptology ePrint Archive, Report 2005/145, 2005, <http://eprint.iacr.org/>
14. L. O’Gorman, I. Rabinovich, *Photo-image authentication by pattern recognition and cryptography*, Int. Conf. Pattern Recognition (ICPR ’96), Vienna, Aug. 1996, pp. 949–953.
15. L. O’Gorman, I. Rabinovich, *Photo-ID encryption and pattern recognition for counterfeit resistance* CardTech/SecurTech ’96, Atlanta, May 1996, pp. 253–261.
16. L. O’Gorman, I. Rabinovich, *Secure Identification Documents Via Pattern Recognition and Public-Key Cryptography*, IEEE Trans. Pattern Anal. Mach. Intell. 20(10): 1097–1102 (1998)
17. M. Ruhl, M. Bern, D. Goldberg, *Secure notarization of paper text documents*, Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (Washington, D.C., United States, January, 2001).
18. C. J. Kuo, C. S. Huang, *A Novel Image Coding Technique for Noisy Communications*, Communications, Computers and Signal Processing, 1993., IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, vol. 1, 1993, pp. 260–263.
19. T. R. N. Rao, *Joint encryption and error correction schemes*, SIGARCH Comput. Archit. News, 12(3), 1984, pp. 240–241.
20. S. C. Kak, *Joint encryption and error correction coding*, IEEE Conference on Security and Privacy, 1983, pp. 55-60.
21. Gligoroski, D., Markovski, S., Kocarev, L. *Error-Correcting Codes Based on Quasigroups*, Unpublished, Submitted to IEEE Information Theory, October 2005.
22. U.S. Provisional Patent Application Serial No. 60/618,173, filed October 13, 2004, under 35 U.S.C. 119. (now disclosed) *Cryptographic Primitives, Error Coding, and Pseudo-Random Number Improvement Methods Using Quasigroups*.
23. <http://en.wikipedia.org/wiki/PDF417>
24. Microsoft technical overview, *Counterfeit-Resistant Optical Fiber*, <http://www.microsoft.com/mscorp/ip/tech/opticalfiber.asp>