INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
34100 TRIESTE (ITALY) · P.O.B. 586 · MIRAMARE · STRADA COSTIERA 11 · TELEPHONE: 2240-1
CABLE: CENTRATOM · TELEX 460392 · 1

SMR.379/13

COURSE ON BASIC TELECOMMUNICATIONS SCIENCE

9 January - 3 February 1989

Encryption

Reginaldo Palazzo Jr.

FEE-UNICAMP. Dept. Telematica. Campinas. SP. Brazil

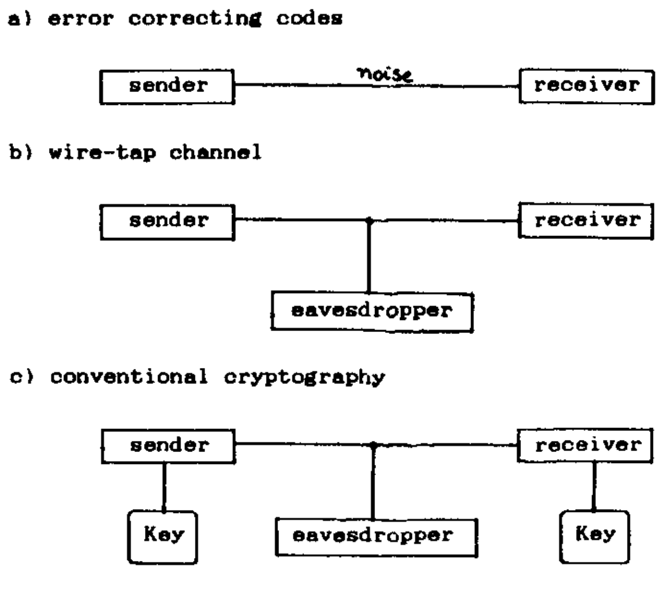
ENCRYPTION

by

Reginaldo Palazzo Jr. (*)

We are going to follow closely the material presented in "Error Correcting Codes and Cryptography", by N.J.Sloane which appeared in Cryptology, up to the case codes which detect deception. The last part is a proposed cryptosystem by the author.

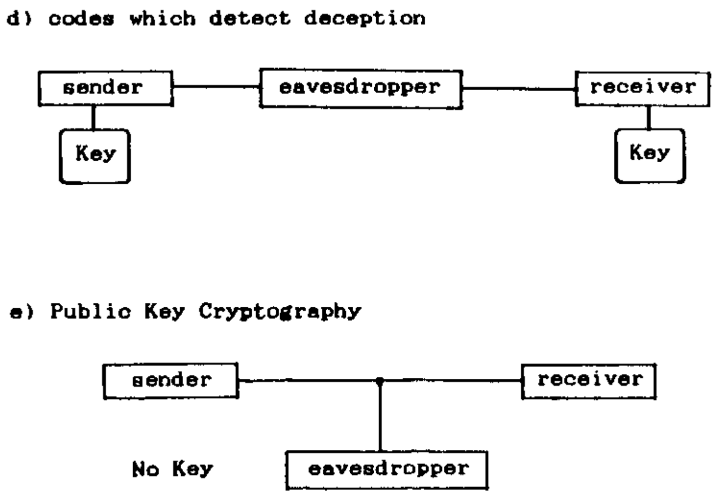
In Fig.1, it is shown 5 different communications systems which will be described in this lecture.



a) error correcting codes

b) wire-tap channel

c) conventional cryptography



d) codes which detect deception

e) Public Key Cryptography

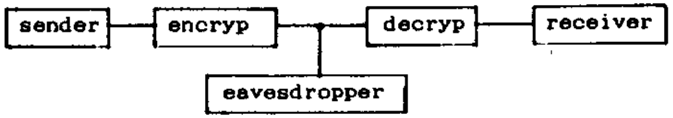
Fig. 1 - Communications Systems.

Case a), we have seen in previous lectures. So, we start with case b).

The main objective in this case is to send information to the receiver as quickly and reliable as possible and simultaneously to minimize what the wire-tap learns.

We are going to assume that the eavesdropper does not possess a good equipment and that the channel he is listenning is noisy, specifically it is a BSC with transition probability p from 0 to 1, and from 1 to 0.

The simplest model assumes that the direct channel is noiseless



(*) - FEE-UNICAMP, Dept Telematica P.O.Box 6101, 13081 Campinas, SP, Brazil/

The solution to this problem is: encrypt 0 as a long, randomly chosen string of 0's and 1's with an even number of 1's. Encrypt 1 as a long, randomly chosen string of 0's and 1's with an odd number of 1's.

Although it is a very good strategy its disadvantage is that the rate goes to zero. It is good in a sense that even though the eavesdropper knows the encryption rule (but not which sequence, even or odd, has been chosen) he is unable to decrypt precisely.

In order to improve the encryption rule such that the rate does not go to zero, let us assume that F^n is the set of all binary vectors of length n. Divide F^n in 2 subsets: the subset En containing all vectors with even number of 1's and the subset Dn containing all vectors with odd number of 1's. Note that Dn is a translate of En and so both subsets are linear codes. Thus, to transmit a 0 randomly choose a codeword from En and to transmit a 1 randomly choose a codeword from Dn.

Therefore, the general solution to the wire-tap problem is: Choose a good linear code C1 containing 2^{n-k} codewords of length n. Now , partition F^n into 2^k cosets of C1, that is,

$$F^n = C1 \cup C2 \cup C3 \cup \dots \cup C2^k$$

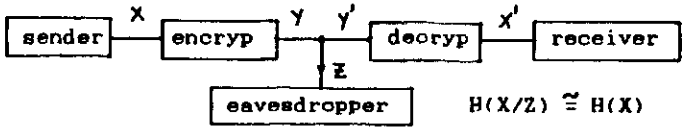
number the possible messages to be sent from 1 to 2^k , then: Encrypt the i-th message as a randomly chosen vector from Ci.

The receiver just has to compute the syndrome of the received vector whereas the eavesdropper is still unable to decrypt. Now the transmission rate is k/n.

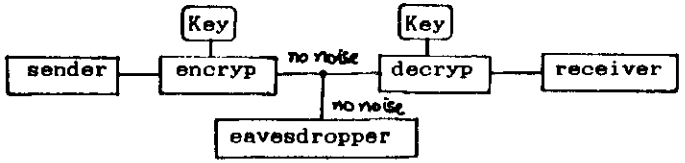
Thus, [Wyner] perfect secrecy is assured if p is the probability of error on the wire-tap, then it is possible to transmit at any rate below

$$- p.\log p - (1-p).\log (1-p)$$

while keeping the eavesdropper in ignorance from what is being transmitted.



Now, let us consider case c). Here we assume that the eavesdropper overhears what is being transmitted without distorion. In conventional cryptography the encryption schemes make use of a key which is known to both sender/receiver but not to the eavesdropper.



The solution to the perfect secrecy is the one-time pad scheme. A long string of randomly generated 0's and 1's are recorded on a tape. The length of this string is the same as the message to be transmitted and it is added mod 2 to the message. At the receiver, the string is added again to obtain the desired transmitted message. Once used it is destroyed. When another message is to be transmitted another string is generated and added to it and the same procedure follows.

The perfect secrecy comes from the following argument: since all the different strings K are possible and equally likely, so are all the possible messages. Therefore, the eavesdropper has learned essentially nothing. The only disadvantage is that it requires as much key as there is data to be transmitted. Even though this is a disadvantage, this scheme is widely used.

The art of designning a good encryption scheme is to find a way of expanding the key, that is, from a small amount of key used as seed to produce a much longer key string. The ways this can be achieved are by use of linear feedback shift-registers, and nonlinear shift-registers.

The DES, Lucifer and many other encryption schemes make use of the idea of a product cipher. This is basically a set of permutations and nonlinear operations. However, if the channel is noisy then one must use error correcting codes to clean up the channel to the encryption scheme to work properly.

Now, we discuss case d), that is, codes which detect deception. The idea here is that the eavesdropper can listen the message and he is allowed to retansmit it. However, a strategy to overcome possible message alteration is to sign or authenticate the message in such a way that the eavesdropper is unable to replace the true message by a false one without being detected.

As an example: suppose a casino is managed by a bad guy who is cheating the owner by reporting the daily takings from the slot machines to be less than they actually are and keeping the difference for himself. To prevent this, the owner proposes to install in each slot machine a secret key K and a device which takes as its input the day's takings X and the key K and produces as output a signature or authenticator $Z = \Phi(X,K)$.

The device punches X and Z onto a paper tape. The bad guy mails the tape to the owner which will read X, recalculate Z from X and K and check this value of Z with that generated on the tape.

On the other hand, the bad guy knows X, Z and Φ (he knows how the device works) but not K and he wishes to replace X and Z by X' and Z' such that $Z' = \Phi(X',K)$.

If this is possible then he can take the difference X'-X. So, a good way to design an authenticator system is to ensure that there are a large number of possible keys corresponding to each message-authenticator pair. Even making use of this strategy does not guarantee that the probability of success will be very small. Indeed Gilbert, MacWilliams and Sloane have shown that this probability is $1/\sqrt{N}$, where N is the number of keys.

Finally, the last communication system to be discussed is the public key cryptosystem. We do this by using a proposed cryptosystem employing unit-memory convolutional codes.

Cryptographic Systems Based on Trellis Codes

1 - Introduction

Since the introduction of the concept of public-key cryptography by Diffie and Hellman [1], many cryptosystems have been proposed and can be found in the open literature in [2]-[5].

The procedure employed in each one of these cryptosystems follows the same basic principle when messages are encoded by use of block codes, that is, each message is encrypted in the former and encoded in the latter independently of the previous ones. Nevertheless, it is well known that encoding of messages by use of convolutional codes (linear trellis codes) avails system's performance when compared to those using linear block codes. Hence we are led to the following question: Is there any advantage in terms of having more secure cryptosystems by introducing memory in the encryption of messages? Under the complexity criterion, the answer is clearly yes since now it has been added to the decryption process another degree of complexity due to the state description process and interdependency between the messages.

Cryptosystems based on t-error correcting Goppa codes is believed to be difficult to break since its high efficiency of correcting errors is destroyed if the bits that make up a codeword are scrambled prior to transmission. We believe linear unit-memory convolutional codes (UM codes) and in general trellis codes are harder to break due to the following reasons: 1) it is shown by an example that by scrambling the bits that make up a codeword the efficiency of good UM codes in correcting errors is destroyed; 2)

the additional fact that good UM codes belong to the class of knapsack problems. Hence, under these premises good cryptosystems can also be found by use of trellis codes [6].

Following this line of using error correcting codes in cryptosystems, we propose in this paper to exploit the use of trellis codes as a means of encrypting messages in cryptographic systems.

It has been shown in [7] that finding good linear unit-memory convolutional codes is equivalent to solve a knapsack problem. It is well known that knapsack problems are NP-complete, thus well suited for cryptographic systems if a trap-door function can be established. In general, one encounters easy and hard knapsack problems. This concept still holds true for linear UM codes. The threshold between these two extreme classifications of knapsack problems (finding good UM codes) is directly related to the number of digits that are fed into the shift-registers since it is assumed that the UM encoder has b parallel K shift-registers.

The dimensionless rate of linear UM codes is defined as $r = b/n$, with b the number of digits to be encoded and n the number of encoded digits, b and n integers such that there is a multiplicity factor greater than or equal to 2 between b and n. Since UM codes can be represented by finite state machines, the number of states and the number of branches leaving any one of the states is exponentially dependent upon b more specifically $2^{b(K-1)}$ and 2^{b} respectively. Under the split state representation of UM codes of rate $r = b/n$, to determine the branch Hamming weights leaving and going into the zero state one is confronted with solving a knapsack problem (see [7]). This knapsack problem is harder to solve since it is not of the superincreasing type as devised by Diffie and Hellman. Therefore, well suited for cryptographic purposes. Solutions of this problem are also exponentially dependent on b where not all of them lead to implementable UM codes.

In the following sections, we introduce public-key and conventional cryptosystems based on UM codes, and in general trellis codes, for use in cryptographic systems.

2 - One Level Knapsack Problem for Public-Key Cryptosystems

Since solving the linear UM code knapsack problem for large values of b is rather difficult, thus with no use in cryptography, one wonders to solving an easy knapsack problem first. To this end, we assume that the two b x n encoding submatrices G_o and G_1 of a "good" UM code for relatively small b have been determined. By "good", we mean a code that attains the largest minimum Hamming distance among all codes. These submatrices and their equivalent ones are easily found for values of $b \leqslant 4$, by hand calculation, when a network flow approach is employed [8].

As mentioned in the introduction, by applying proper transformations to the generating functions of a UM code, G_o and G_1, its efficiency of correcting errors is destroyed. Hence, a trap-door function can be found by choosing A and B, b x b and n x n invertible matrices respectively, and apply them to the generating functions G_o and G_1. We call this procedure a direct trapdoor function. Thus, the direct trapdoor function gives new generating functions G_o' and G_1' as follows

$$G_i' = A \cdot G_i \cdot B \qquad i = 0, 1 \tag{1}$$

These generator matrices are put in a public file as

$$G' = [G_o' \,;\, G_1'] \tag{2}$$

The encoding process by use of a UM code is defined by

$$y_t = x_t \cdot G'o + x_{t-1} \cdot G'1 \quad \text{with} \quad t \geqslant 0 \quad \text{and} \quad x_{-1} = \underline{0} \tag{3}$$

with $x_t \in GF(2)$. Prior to decoding, we apply B^{-1} to y_t to obtain

$$y_t \cdot B^{-1} = (x_t \cdot A) \cdot G_o + (x_{t-1} \cdot A) \cdot G_1 \tag{4}$$

the operation "+" in (3) and (4) are modulo 2. Finally, x_t is obtained by use of the Viterbi or sequential decoding algorithms as in the usual way. Note that in order to break this scheme, the cryptanalyst has to solve a knapsack problem related to the generating functions G_o and G_1 and find the transformations A and B.

As a simple example of this cryptosystem, let us assume the UM code has rate $r = 2/4$. A good UM code in the set of UM codes with rate $r = 2/4$ can be found by solving an easy knapsack problem corresponding to its generating functions. From [7], we have that this knapsack is given by

$$a_1 \cdot d_1 + a_2 \cdot d_2 + \dots + a_B \cdot d_B = n \cdot 2^b \tag{5}$$

where a_i are the number of branch having Hamming weights d_i leaving the zero state and going into it after 2 branches for $i = 1, 2, 3$ with B being a constant. For UM code of rate $r = 2/4$, we have from equation (6) of [7] that $d_1 = 5$, $d_2 = 6$, $d_3 = 7$, and so on, and the right hand side of (5) equals 16. Therefore, a possible solution given a good UM code has $a_1 = 2$ and $a_2 = 1$. This solution generates a UM code with generating functions G_o and G_1 given by

$$G_o = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad G_1 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

The minimum distance of this code is easily shown to be $d_{min}\{G_o, G_1\} = 5$. Let the transformations A and B be given by

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Applying A and B to G_o and G_1 as given by (1), we have G'_o and G'_1 given by

$$G'o = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad G'1 \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The minimum distance of this new UM code is $d_{min}\{C'_o, C'_1\} = 3$. Hence, the efficiency of correcting errors is destroyed. For in this case only one bit in error can be corrected, whereas in the former up to two bits in error can be corrected. If a known error pattern with any two digits taking value 1 is added to the encoded sequence using a UM code with generating functions G'o and G'1 only the sender is able to decode correctly the information sequence. Improved error corrections can be achieved by increasing the multiplicity factor between b and n keeping b constant or increasing b.

3 - Two Levels Knapsack Problem for Public-Key Cryptosystems

In keeping up with the idea of increasing the complexity and therefore obtaining more secure cryptosystems, the next scheme to be presented has two levels of knapsack involved. The first level, deals with the knapsack problem of finding the generating functions of the UM code, or equivalently, the one level knapsack problem of section 2. Once the generating functions have been determined, the second level makes use of the idea that each row of G_o and G_1 (or G'o and G'1) is a binary representation of an integer number. Like the "block" knapsack proposed by Diffie and Hellman, we have a set of integers numbers, the knapsack, which will be used in a convolutional way to encrypt messages to be transmitted. We call this a <u>direct two level knapsack</u>.

Let us assume the first level has been solved, that is, the generating functions G_o and G_1 are known. The second level represents each row of the generating functions by its corresponding integer number followed by a proper transformation which will lead to members of another knapsack. This transformation consists of selecting two large numbers p and w such that the greatest common divisor is one, gcd(p, w) = 1. We apply the transformation p mod w to G'_o and G'_1 as given by equation (1), to obtain

$$\bar{G}_i = \left[(G'_i \cdot p) \bmod w\right] \qquad i = 0,\ 1 \tag{6}$$

These generator matrices are put in a public file as

$$\bar{G} = \left[\bar{G}_o\ ;\ \bar{G}_1\right] \tag{7}$$

The encoding of a UM code is defined by

$$y_t = x_t \cdot \bar{G}o + x_{t-1} \cdot \bar{G}1 \qquad \text{with} \quad t \geqslant 0 \quad \text{and} \quad x_{-1} = 0 \tag{8}$$

where $x_t \in GF(2)$. Prior to decoding, we apply $q \bmod w$ $(q = p^{-1})$ to y_t to obtain

$$(y_t \cdot q) \bmod w = x_t \cdot \left[(\bar{G}_o \cdot q) \bmod w\right] + x_{t-1} \cdot \left[(\bar{G}_1 \cdot q) \bmod w\right] \tag{9}$$

and B^{-1} to (9) to get

$$y_t . B^{-1} = (x_t . A) . G_o + (x_{t-1} . A) . G_1 \qquad (10)$$

the "+" operations in (8)-(10) are the real addition operation. Finally, x_t is obtained by use of the Viterbi or sequential decoding algorithms as in the usual way. Note that without knowing A, B, p, w, and G'_i the decryption becomes very difficult due to the fact that it is equivalent to solve a sequence of knapsack problems.

As a simple example to show the procedures involved, let us take the previous UM code of rate $r = 2/4$. The generator matrices G_o and G_1 are given by

$$Go = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \qquad G1 = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$$

Now, we take the rows of G_o and G_1 as the binary representation of an integer number, that is,

$$Go = \begin{bmatrix} 5 \\ 14 \end{bmatrix} \qquad G1 = \begin{bmatrix} 7 \\ 10 \end{bmatrix}$$

Let $p = 3950$ and $w = 8443$, then $q = 2550$. Applying p mod w to G_o and G_1 above, we have

$$\tilde{Go} = \begin{bmatrix} 2864 \\ 4642 \end{bmatrix} \qquad \tilde{G1} = \begin{bmatrix} 2321 \\ 5728 \end{bmatrix}$$

The encoding is

$$y_t = x_t . \tilde{G}_o + x_{t-1} . \tilde{G}_1$$

Let the input sequence be 01, 10, 11, 01, ... This input sequence generates a path through the trellis where the corresponding branch output sequence is 4642, 8592, 9627, 12691, ... Note that at this point the cryptanalist is confronted with a sequence of interdependent knapsacks to solve. At the decoder, we apply q mod w to each branch output from the incoming sequence to obtain 14, 15, 26, 31, ... By use of the Viterbi algorithm we finally have 01, 10, 11, 01, ... as the original data input sequence.

In the previous cryptosystem, the basic idea was to operate directly in a convolutional way on the "knapsack" provided by the rows of G_o and G_1 as integers. The variation to be described follows the idea of operating not directly in a convolutional way on the integers provided by G_o and G_1, but on the corresponding integers from the mod 2 operation between the rows of the generating matrices. We call this undirect two level knapsack.

Since in the undirect two level knapsack one can not operate directly on the set of integer numbers (knapsack) another trapdoor function has to be devised. This new trapdoor function, which operates on the integer numbers that originate from the operation modulo 2 on the rows of Go and Gl, can be any one of the public key schemes devised in [1], [3], and [4]. The on that fits best is the RSA public-key cryptosystem. Under the RSA cryptosystem, one selects two prime numbers $\tilde{p}$ and $\tilde{q}$ and define $\tilde{n}$ as $\tilde{n} = \tilde{p} . \tilde{q}$. Once the Euler's function $\phi(\tilde{n}) = (p-1) . (q-1)$ is known, one selects an integer number E between 3 and $\phi(\tilde{n})$ with no common factor with $\phi(\tilde{n})$. Then, one can find an integer number D which is the "inverse" of E mod $(\tilde{n})$. Next, E and $\tilde{n}$ are displayed in a public file.

The encryption of a message M, represented by a corresponding cyphertext C, which is an integer number between 0 and $\tilde{n} - 1$ follows

$$C = M^E \bmod \bar{n}$$

Decryption of the cyphertext C follows by using the secret decryption number D as

$$M = C^D \bmod \bar{n}$$

Now we introduce the variation as follows : let E, D, $\bar{n}$, and $\{\bar{G}_o, \bar{G}_1\}$ the encyphering, decyphering, an integer, and the generating functions of the UM code with rate $r = b/n$ be given. Let the displayed information in a public file consists of

$$E, \bar{n}, \text{ and } \{\bar{G}_o, \bar{G}_1\}$$

Any one interested in communicating with user A, takes b input digits at a time and encode them by use of $\{\bar{G}_o, \bar{G}_1\}$. The resulting string of binary numbers is transformed to a corresponding integer number M_i. This number is encyphered by use of the RSA cryptosystem as

$$C_i = M_i^E \bmod \bar{n}$$

Upon arriving at the destination, C_i is decrypted as

$$M_i = C_i^D \bmod \bar{n}$$

Next, a binary representation of M_i is taken. Finally, this binary sequence is decoded by use of the Viterbi or any other sequential decoding algorithms. Note that in order for a cryptanalist to break this cryptosystem he has to break first the RSA cryptosystem, to find the proper direct trapdoor function, and finally to solve the UM knapsack problem.

4 - Nonlinear UM Codes for Use in Conventional Cryptosystems

In conventional cryptographic systems keeping secret a selected "key" from an ensemble of keys is a must operation. Consistant with this premise, one can make use of the unknown "initial state", the content of the first b shift-registers of UM codes (linear or nonlinear) in the parallel representation, as the secret key to be employed with the condition that no zeros are appended to the data input sequence.

This observation comes from the fact that in conventional use of convolutional codes with rate $r = b/n$ and constraint length K the initial state is set up to zero (or to any other known state) and that after an input data sequence of length $L \cdot b$ digits has been encoded $b \cdot (K-1)$ zeros are appended to it. This implies that independently if error(s) are introduced or not by the channel to the encoded output sequence there will be a path in the trellis that diverges from the zero (or from a known initial) state and comes back to the zero state $L + K - 1$ branches later.

This is not the case when $b \cdot (K-1)$ zeros are not appended to the data input sequence even though the initial state is known. For now the decoder has to look at all possible paths starting from a known initial state and leading to all ending states after L branches. On the other hand, if the initial state is not known, the decoder has to look at all possible paths starting in any one of the initial states and leading to all possible ending states after L branches. However, in this case the last data input digits do not have the same error protection as when $b \cdot (K-1)$ zeros are appended. To avoid this one can insert into the shift-registers the initial or last b bits of each message stream of length $(L + b)$ bits followed by the $(L + b)$ data input. Note that the initial and ending states are unknown every time $(L + b)$ data input is ready for transmission. In any case, at most

$2^{b(K-1)}$ runs of the Viterbi or any of the Sequential decoding algorithms are necessary to decode a particular data input sequence. It is clear that for UM codes with moderated values of b the number of states and consequently the number of runs is extremely large. In general, it takes about 5 constraint length to have a good estimate of the data input sequence from a known initial state.

In the linear UM codes case we have that the encoding functions operate on the digits (message) to be encoded together with the previous ones to encrypt the next block of digits in a linear fashion (mod 2 operation). For nonlinear UM codes this may not be the case since now the operation(s) are arbitrary. In case that at least one of these operations is a nonlinear function the resulting codewords, in general, form a nonlinear code.

The difference between nonlinear and linear UM codes, with no zeros appended to the L. b data input with known or unknown initial state, is that the decoder for nonlinear UM codes has to make pairwise path comparisons of length L starting and ending in any one of the possible states in the trellis independently if the initial state is known or unknown. Why these pairwise comparisons have to be made is due to the fact that paths or a set of paths may or may not have different accumulated metrics. This simple fact adds new difficulties to a cryptanalist since now he has to compare all possible codewords that start and end in any one of the possible states in the trellis.

This analysis is effectively done by use of the superstate approach. Note that for moderately values of b the number of superstates and path comparisons is the squared value of the original number of states in the trellis.

On the other hand, it is known that this problem is solved in polynomial time if all accumulated metrics are equal or all lengths are equal even though the number of comparisons is large. Otherwise, this problem is known to be NP-complete since it is equivalent to the "shortest weight-constrained path" [9]. Consequently, we have that the security of this scheme relies on the fact that it belongs to the class of "hard" problems. Therefore, for conventional cryptosystems using linear or nonlinear UM codes for moderate values of b, the "initial state" chosen as the key is secure.

5 - Conclusions

We have presented some cryptographic schemes for use in public-key and in conventional cryptography by use of linear and nonlinear unit.memory convolutional codes respectively. Due to the "equivalence" of convolutional codes and UM codes these schemes can easily be adapted to the former in the encryption of messages. The security of these schemes lies on the premises that they belong to the class of NP-complete problems. Therefore, the complexity inherent to them is the main factor to rely on.

References

[1] W. Diffie and M.E. Hellman, "New direction in cryptography," IEEE Trans. Inform. Theory, vol. IT-22, pp. 644-654, Nov. 1976.

[2] R.C. Merkle, "Secure communication over an insecure channel," Common. Assoc. Comput. Mach., vol. 21, pp. 294-299, Apr. 1978.

[3] R.C. Merkle, and M.E. Hellman, "Hiding information and signatures in trapdoor knapsacks," IEEE Trans. Inform. Theory, vol. IT-24, pp. 525-530, Sept. 1978.

[4] R.L. Rivest, A. Shamir, and L. Adleman, "On digital signatures and public key cryptosystems," Common. Assoc. Comput. Mach., vol. 21, pp. 120-126, Feb. 1978.

[5] R.J. McEliece, "A public key system based on algebraic coding theory," JLP DSN Progress Rep., 1978.

[6] R. Palazzo, Jr., "Unit-memory convolutional codes as a means in cryptographic systems," Second SIAM Conference on Applied Linear Algebra, Raleigh, North Carolina, April 29 - May 2, 1985.

[7] R. Palazzo, Jr., "Linear unit-memory codes - a knapsack problem?," 2nd Swedish-USSR International Workshop on Information Theory, Granna, Sweden, April 14-19, 1985.

[8] R. Palazzo, Jr., "New short constraint length convolutional codes derived from a network flow approach," IEEE Intern. Symp. on Inform. Theory, Brighton, England, June 23-28, 1985.

[9] M.R. Garey, and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Co., San Francisco, 1979.