7

A Cryptosystem Based on Error Control Codes and Stream Ciphers

E. Dawson
Information Security Research Centre
Queensland University of Technology
GPO Box 2434
Brisbane Queensland
Australia 4001
dawson@fit.qut.edu.au

D. Polemi
Institute of Communications and Computer Systems
National Technical University of Athens
Heroon Polytechniou 9
157 73 Zografou
Athens, Greece
polemi@softlab.ece.ntua.gr

Abstract.

A new single key cryptosystem is proposed based on error control binary codes and secure stream ciphers. The system provides both security and the capability of correcting channel errors.

Keywords

Codes, stream ciphers, cryptosystems, error control.

Communications and Multimedia Security Vol. 3 S. Katsikas (Ed.)

1 Introduction

In 1976 McEliece [20] introduced a promising public key cryptosystem based on *binary classical Goppa codes.* The security of this scheme is based on the NP-completeness of the decoding problem for these codes. There have been many unsuccessful attempts to cryptanalyze this scheme [1], [26], [22], [8], [15], [16], [2].

In relation to this cryptosystem we have the Niederreiter [21] and the Stern [29] schemes. Janwa and Moreno [12] strengthen the security of the above schemes by using the new and much larger class of *q-ary algebraic geometric (AG) Goppa codes.*

Secret key cryptography is widely used in securing communications since single key cryptosystems are very fast and highly suitable for confidentiality services. Secret key cryptosystems (using classical Goppa codes) variant of the McEliece scheme were proposed by Jordan [13], Rao [25], Rao and Nam [26], and Li-Wang [17]. These cryptosystems have been cryptanalyzed because of their common weakness which is the error selection procedure they follow [30].

In this paper we propose a new single key cryptosystem using *binary codes* and secure *stream ciphers.* Stream ciphers are a class of single key cryptosystems [27], [28]. These ciphers are systems which generate pseudo random sequences. Examples of these ciphers are given in [27], [28]. The cryptosystem will provide for both privacy of messages as well as the capability of correcting channel errors. The keystream generator will be selected to provide a high level of security. The error control code will be selected for channel error correction as well as providing an extra level of security by including the actual code selected as part of the key. This is distinct from previous schemes using error control codes where errors are added before transmission. We believe that it is better to use the error correction capability of code as it is intended to correct channel errors.

When we design secure cipher systems there are trade offs between easy implementation, security, speed. The security of our proposed cryptosystem is based primarily on the cryptographic strength of the chosen stream cipher. As will be shown the added complexity of keeping the actual error control code secret adds further security preventing possible known plaintext attacks on the stream cipher. It should be noted that the use of a linear code by

itself without a stream cipher offers a low level of security.

The paper is organized as follows: In §2 we briefly review the construction of binary codes and the general description of stream ciphers. In §3 we describe two models for our proposed single key cryptosystem. In §4 an analysis of the security of the cryptosystem is conducted. Conclusions and future directions for development of this scheme are contained in §5 .

2 Preliminaries

In this section we will briefly review the construction of error control codes and the structure of stream ciphers. For further reading on error control codes we refer the reader to [18] and [23], and on stream ciphers to [27].

2.1 Error Control Codes

There are two classes of error control codes, namely block and convolutional codes. In this paper we will only refer to linear block codes over the binary field. A code C with parameters denoted by [n,k,d] is a subspace of dimension k of the binary vector space of n tuples. The parameter d denotes the codeword in C of minimum nonzero weight. Such a code using a maximum likelihood decoding rule is capable of correcting up to t channel errors where t is the largest integer less than or equal to (d-1)/2.

For an [n,k,d] code C, a generator matrix, G, is a k by n matrix with a basis of C as its rows. If we divide the message into binary k tuples then the method to form codewords of length n for transmission is to multiply a message block by G.

There are numerous types of codes to select. The public key cryptosystem described by McEliece used Goppa codes. Since this scheme was public key a very large code was required i.e. in original scheme [20] it was proposed to use a Goppa code [1024,524,101]. This required storage of the order of 1/2 million bits for the public and secret keys. In the symmetric cipher described in this paper it is possible to use a smaller code. A code will be selected to offer high speed decoding and maximum error correction capability such as a BCH, Reed-Solomon or Goppa code.

2.2 Stream Ciphers

There are two classes of cryptographic algorithms, *symmetric (single key)* and *asymmetric (public key)* [5],[28], [27]. In symmetric algorithms we use the same key for both encryption and decryption, where in asymmetric we use different keys.

A special class of symmetric algorithms is that of *stream ciphers.* Stream ciphers are preferable in communications because of their good performance in hardware implementation (speed) and low error propagation.

A stream cipher specifies a device with internal memory that enciphers the jth digit p_j of the plaintext stream into the jth digit c_j of the ciphertext stream using a function which depends on both the key k and the internal state of the stream cipher at time j. This function is called a keystream generator. In most cases both the plaintext and keystream are binary. A ciphertext bit is formed from combining, using modulo two addition, the corresponding plaintext and keystream bits.

The type of stream cipher used in this paper is the *synchronous* stream cipher. In synchronous ciphers the next state depends only on the previous state and not on the input so the succession of states is independent of the message stream. If a ciphertext bit gets lost the communicating parties have to resynchronize their generators.

Most keystream generators use sequences produced by one or more linear feedback shift registers (LFSR) [27]. An LFSR offers a high speed mechanism in either software or hardware for producing binary sequences. We can categorize such stream ciphers into *state filtered* generators which employ one or several LFSRs and the *clock-controlled* generators which employ some LFSRs to control other LFSRs. The key for such keystream generators is selected as the initial state vectors of the input LFSRs. As well, in some cases the tap settings of the LFSRs may be used as part of the key.

There are five standard measures used commonly to examine the level of security of a stream cipher.

Period : The key sequence should have long period.

Linear Complexity : The length of the shortest LFSR able to produce the key sequence must be large.

Statistical Tests : The key sequence should satisfy all statistical tests

(frequency, binary derivative, change point, poker, runs, sequence complexity, linear complexity, universal) [11].

Noise like characteristics: The key sequence should have noise like characteristics.

Correlation Immunity: The sequence should have high order correlation immunity.

Unfortunately a system can satisfy the above requirements and still be vulnerable to a cryptanalytic attack [9]. For example, the summation generator due to Rueppel [27] was thought to be secure if more than two LFSRs are used in the design. However, recent attacks [4], [10], [14] have shown that it may be possible to attack the summation generator even in the case where more than two input LFSRs are used.

3 Description of Cryptosystem.

The proposed cryptosystem consists of a composition of a stream cipher and an error control code. The key of the system will be made up of both the generator matrix of code and the key of the stream cipher.

As mentioned in the introduction the operation of the linear code is primarily to provide error correction. However, by keeping the actual code used secret we will show that we add an extra layer of security. We shall describe two models for the cryptosystem. In the first model (Model 1) a code is followed by a stream cipher, where in the second model (Model 2) we use a stream cipher first followed by a code. We will use the following notation:

Secret keys:
a) A generator matrix G with dimensions $k \times n$ of an [n,k,d] binary code C.
b) The key of a secure stream cipher.

Message (Plaintext): A binary vector (block) $\mathbf{p} = (p_1, ..., p_k)$ of length k.

MODEL 1 (Code first)

Encryption	
Step 1	Form $r = p\mathrm{G}$.
Step 2	Add modulo two the next n bits of the keystream with r to produce the next n bit ciphertext block c.
Decryption	
Step 1	Add modulo two the same n bits of the keystream to received n bit block. This gives $r + e$ where e represents a possible error pattern from channel.
Step 2	Apply decoding algorithm of code to recover p (assuming that number of errors is less than t).

MODEL 2 (Stream cipher first)

Encryption	
Step 1	Add modulo two the next k bits of the keystream to p to form k-bit block r.
Step 2	Form n-bit ciphertext block $c = r\mathrm{G}$.
Decryption	
Step 1	Apply decoding algorithm of code to received block to recover r (assuming that the number channel errors is less than t).
Step 2	Add modulo two the same k bits of keystream to r to recover p.

4 Security Analysis

In this section we shall conduct an analysis of the security of each model of the cryptosystem described in the previous section. We are assuming that the cryptanalyst knows the cryptosystem being used including the type of keystream generator and error control code, but does not know the actual key selected i.e. the initial state vectors of keystream generator and generator

matrix of code are unknown. The aims of the attacks that we describe are to recover these two keys.

It should be noted that, in general, the attacks on keystream generators such as the summation generator require known plaintext in addition to the intercepted cryptogram. This allows the cryptanalyst to derive the corresponding bits in the keystream.

The aim in such attacks is to derive the state vectors of the input shift registers. In this fashion we shall analyze the models based on the assumption that we can attack the keystream generator if it was used by itself provided m bits of the keystream are known.

4.1 Cryptanalysis of Model 1

In the first model it may be possible to attack the scheme using chosen plaintext. Here we are assuming that the cryptanalyst has the assistance of an insider and the encryption takes place within a tamperproof encryption box.

The insider is able to input chosen plaintext into encryption box. The procedure would be for the insider to put in the encryption box all zero blocks until the total length of these blocks is at least m bits. The cryptanalyst with intercepted ciphertext now has available m bits of keystream generator.

Then the cryptanalyst can apply attack on keystream generator to derive state vectors of input shift registers. Finally, given the encryption of a further k known plaintext blocks (provided these plaintext blocks are linearly independent) the cryptanalyst can derive the generator matrix of code and has broken the encryption scheme completely.

It should be noted that we are assuming that there are no channel errors. In the case where there are channel errors the cryptanalyst clearly would require additional bits of known plaintext.

In the case of a standard stream cipher, which only uses a keystream generator to encrypt, an attacker gains no advantage by using chosen plaintext instead of known plaintext. In both chosen and known plaintext attacks the cryptanalyst obtains the equivalent bits in the keystream (given the intercepted ciphertext).

In this fashion the cryptosystem outlined in Model 1 does enhance the security of the stream cipher, since in order to conduct the attack described above the cryptanalyst requires chosen plaintext.

4.2 Cryptanalysis of Model 2

In Model 2 the cryptanalyst is not able to conduct the chosen plaintext attack described above since the encryption by stream cipher first prevents this type of attack. Moreover multiplication by the matrix G in Step 2 has disguised the action of the keystream generator. However this does not prevent all known attacks.

In particular chosen ciphertext attack may be applied. In such attack the attacker needs to be able to control decryption device at receiver end, i.e. attacker needs to be able to insert ciphertext and have access to resulting decrypted text. The goal of the attack is to find state vector of stream cipher. The attacker would input all zeros as ciphertext. The decryption of all zeros ciphertext would be corresponding bits of keystream in Model 2. This may allow attacker to attack keystream generator similar to chosen plaintext attack in first model.

5 Conclusion

The complexity, error correcting capability and added redundancy of both models is the same. Both models provide approximately the same level of security in that the difficulty of conducting a chosen plaintext attack and a chosen ciphertext attack on a symmetric are about the same. The first attack requires an insider at the sender end while the latter requires an insider at the receiving end. Our scheme does not require large block length as in McEliece's scheme. We can use binary codes of smaller length. Hence it does not produce much computational overhead for encryption and decryption so it can be used for practical computer communications.

The synchronization of the stream cipher used for encryption and decryption can be an important problem in real life applications. Synchronization can be attained using various techniques (e.g a sequence count).

Further research. We would like to implement the proposed cryptosystem and compare it with the McEliece's software implementation in [24]. In this implementation we need to select a keystream generator and error control code. In particular we need to decide on which code to select to provide good error correction, decoding capability and security as well as adding a minimum of overhead in extra bandwidth requirements. There are extra

bandwidth requirements in the proposed system in increasing each plaintext block from k bits to ciphertext blocks of n bits. In addition there are extra bandwidth requirements in having to transmit the key consisting of the initial state vector of stream cipher and the generator matrix of the error control code over a secure channel. The generator matrix of the code would require the transmission of kn bits over a secure channel. Clearly, we must select k and n in such a manner to minimize these requirements.

References

[1] C.M. Adams and H. Meijer. Security-related comments regarding McEliece public-key cryptosystems. *Advances in Cryptology-CRYPTO, New York: Springer Verlag*, pages 224–228, 1987.

[2] M. Beth, T.and Frich and G.J. Simmons. Public key cryptography: State of the art and future directions. *LNCS, Springer Verlag*, 578, 1992.

[3] M. Bronstein, M. Hassner, A. Vasquez, and C.J. Williamson. Computer algebra algorithms for the construction of error correcting codes on algebraic curves. *IEEE Proceedings on Information Theory*, June 1991.

[4] E. Dawson and A. Clark. Divide and conquer attacks on certain classes of stream ciphers. *Cryptologia*, XVIII:25–40, 1994.

[5] D.E. Robling Denning. *Cryptography and Data Security.* Addison-Wesley Publishing Company, N.Y, 1983.

[6] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, 22 (6):644–654, 1976.

[7] G.L. Feng and R.R.N. Rao. Decoding of algebraic geometric codes up to designed minimum distance. *IEEE Trans. Inf. Theory*, 39:37–46, 1993.

[8] J.K. Gibbon. Equivalent Goppa codes and trapdoors to McEliece's public key cryptosystem. *EUROCRYPT'91, Lect. Notes in CS*, 547:68–70, 1991.

[9] J. Golić. Linear cryptanalysis of stream ciphers. *Preneel (ed.): Fast Software Encryption: Proceedings of 1994 Cambridge Security Workshop. Lecture Notes in Computer Science, Springer*, 1008:154–169, 1995.

[10] J. Golić, M. Salmasizadeh, E. Dawson and A. Khodkar. Cryptanalysis of the summation generator with three input LFSRs. *Proceedings of ISITA'96, Victoria, Canada*, Sept. 1996, pp 343-346.

[11] H. Gustafson, E. Dawson, L. Nielsen, and W. Caelli. A computer package for measuring the strength of encryption algorithms. *Computers and Security*, 13:687–697, 1994.

[12] H. Janwa and O.Moreno. McEliece public key cryptosystem using algebraic-geometric codes. *Journal of Design Codes and Cryptography*, 1995. (to appear).

[13] J.P. Jordan. A variant of a public key cryptosystem based on Goppa codes. *Sigact news*, 15:61–66, 1983.

[14] A. Klapper and M. Goresky. Cryptanalysis based on 2-Adic rational approximation. *Advances in Cryptology - Proceedings of Crypto'95 (D. Coppersmith, ed), Lecture Notes in Computer Science, Springer Verlag*, 963:262-273, 1995.

[15] V.I. Korzhik and A.I. Turkin. Cryptanalysis of McEliece's public-key cryptosystem. *EUROCRYPT'91, LNCS*, 547:68–70, 1991.

[16] P.J. Lee and E.F. Brickell. An observation on the security of McEliece's public-key cryptosystem. *Advances in Cryptology-EUROCRYPT'88, Springer LNCS*, 330:275–280, 1988.

[17] Y.X. Li and X.M. Wang. A joint authentication and encryption scheme based on algebraic coding theory. *Algebraic Algorithms and Error Correcting Codes 9, Lecture Notes in Computer Science*, 539:241–245, 1991.

[18] F.J. MacWilliams and N.J.A. Sloane. The Theory of Error-Correcting Codes. *North-Holland Publishing Company*, 1977.

[19] U.M. Maurer and J.L. Massey. Cascade Ciphers: The Importance of Being First. *Journal of Cryptology*, Vol 6, Number 1, 1993, pp 55-61.

[20] R.J. McEliece. A public key cryptosystem based on algebraic coding theory. *DSN Progress Report, Jet Population Laboratory*, pages 114–116, 1978.

[21] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory*, 15:159–166, 1986.

[22] C.S. Park. Improving code rate of McEliece's public-key cryptosystem. *Electronics Letters*, 25 (21):1466–67, 1989.

[23] A. Poli and L. Huguet. Error correcting codes: Theory and Applications. *Prentice Hall International Ltd*, 1992.

[24] B. Preneel, A. Bosselaers, R. Govaerts, and J. Vandelwalle. A software implementation of the McEliece's public key cryptosystem. *Proceedings of the Thirteen Symposium on Information Theory, Benelux*, pages 119–126, 1992.

[25] T.R.N. Rao. Cryptosystems using algebraic codes. *Int. Conf. on Computer Systems & Signal Processing*, 1984.

[26] T.R.N. Rao and Kil-H. Nam. Private key algebraic code encryption. *IEEE Trans. Inform. Th.*, IT-35, 1989.

[27] R. Rueppel. *Stream Ciphers.* in Contemporary Cryptology The Science of Information Integrity, G. Simmons (ed.), pp. 65–134, IEEE Press, 1992.

[28] B. Schneier. *Applied Cryptography, Protocols, Algorithms and Source Code in C.* John Wiley and Sons, Second Edition, N.Y, 1996.

[29] J. Stern. A new identification scheme based on syndrome decoding. *Advances in Cryptology -Proceedings of Crypto'93 (D.R. Stinson, ed.) Lecture Notes in Computer Science, Springer Verlag*, 773:13–21, 1994.

[30] J. van Tilburg. *Security Analysis of a Class of Cryptosystem Based on Linear Error-Correcting Codes.* Ph.D thesis, 1994.