The encrypted and decrypted values are differenced, and the resultant difference is de-grouped to form a set of 8-bit difference values, each corresponding to a respective data block in the original bit stream.

Each difference value is then inverse-transformed using IDCT similar to that used in the described encryption scheme. The position of each difference value in the respective IDCT is again determined by the output of a pseudorandom number generator, and all the other coefficients in the IDCT are set to zero. The pseudorandom number generator generates the same sequence of numbers as that generated by pseudorandom number generator used in the encryption scheme and, to that end, the pseudorandom number generator is controlled by the same pseudorandom seed as also used in the encryption scheme. This is generated by decrypting the ciphertext of the seed contained in the sent-out message using further RSA algorithm.

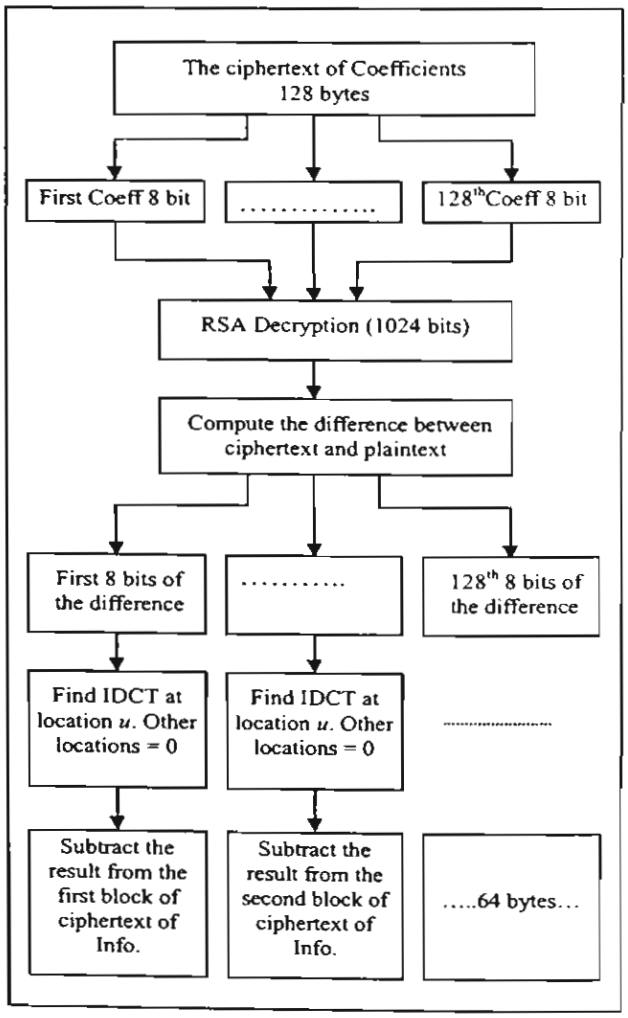

Figure 7. Flow Diagram of the Decryption Scheme in More Detail

4. The inverse-transform values derived from the difference values are subtracted from the ciphertext of the original data values contained in the sent-out message using modular arithmetic whereby to reconstitute the original digital bit stream.

4. PERFORMANCE ANALYSIS

According to our scheme, only one of 64 DCT coefficients derived from each block B_1, B_2.....B_{128} is subjected to the RSA encryption process. Thus, in the present example, where the digital bit stream consist of 128 data blocks, each containing 64 bytes (a total of 8192 bytes) the RSA encryption process need only be applied to a block of plaintext containing a mere 128 bytes. This gives a very considerable saving in computation time; indeed, compared with a conventional method in which the en/decryption operation is applied to all the data bytes. The described method gives a reduction rate of 1/64, corresponding to a reduction in computation time of 98.4375 %. In this connection it will be understood that the adding and subtracting operations used in the present method involve a negligible amount of computation time compared with the en/decryption operations used in the scheme.

In this example, the transformation operations are carried using an 8×8 DCT and an 8×8 IDCT. However, it will be appreciated that any other suitable transformation operation could alternatively be used, such as a discrete Fourier transform (DFT) and an inverse discrete Fourier transform (IDFT). The block size of transformation can also be varied which results in the reduction rate of $1/n$, where n is the maximum number of samples in transformation block.

Note that we do not claim that our method gains better performance than any other stream ciphers. On the contrary, we understand very well that one could use the pseudorandom number generator as a stream cipher to encrypt the entire data at little overhead for the performance compared to the proposed method. However, such a technique would not have the advantages of asymmetric algorithms. One may argue that a hybrid method might be used instead to achieve such a requirement. However, this is not the case since the entire encrypted data depends on the strength of the symmetric algorithm, either block cipher or stream cipher, used in the encryption scheme. On the other hand, from the proposed method, every single block of encrypted data is protected by the mathematical problem complexity inherent in asymmetric algorithms.

5. SECURITY ANALYSIS

In the given example, the RSA algorithm is used in the en/decryption process. The reliability of the RSA algorithm depends on the size of the generated ciphertext. In the described example, the ciphertext generated by the RSA encryption process consists of a 1024-bit number, which is considered to provide an adequate level of security. A ciphertext consisting of a 2048-bit number would give an even greater level of security; however, this would require much more computation time. In practice, the alternative block encryption algorithms could be used provided these give an adequate level of security. Another concern when employing block ciphers is the known-plaintext attack. Since cryptanalysis relies on exploiting redundancies in the plaintext, compressing the data before encryption reduces these redundancies and also speeds up the entire process.

Provided the pseudorandom seed remains secret it should not be possible to discover the pseudorandom sequence of numbers used to determine the location of coefficients in the transformation and inverse-transformation operations of the scheme. Assuming the attacker can produce the same pseudorandom sequence of numbers used in the encryption scheme, he or she still needs to break a block cipher in order to obtain the plaintext of each data block. Therefore, it can be considered that the security of the overall system depends on the strength of the algorithm used.

Finally, as already described, the encryption method adds a random value to the original data values. Accordingly, it is not possible to recover the original data values from the corresponding ciphertext directly, without at least having knowledge of the encryption algorithm used in the encryption scheme.

6. CONCLUSIONS

In this paper, the problem of encrypting large amounts of data, especially by using an asymmetric algorithm, has been addressed and an encryption method has been proposed to resolve such a problem. In the encryption scheme, the discrete linear transform is used to reduce the computation time required in the en/decryption process, while maintaining a high level of security. From the given example, the computational complexity when encrypting a file is reduced by 98.4375 % compared to the conventional encryption method. The security of the whole system mainly relies on the asymmetrical algorithm used in the scheme. This method is particularly well-suited for applications that require a high bit rate such as for subscription broadcast and digital TV services.

7. ACKNOWLEDGMENT

The author thanks the Thailand Research Fund for partly supporting this research work (PDF/27/2543).

8. REFERENCES

[1] R. L. Rivest, A. Shamir and L. Adleman, '*A Method for Obtaining Digital Signatures and Public Key Cryptosystems*', Communications of the ACM, Vol. 21, n. 2, February, 1978, pp. 120–126.

[2] B. Schneier, '*Applied Cryptography*', Wiley, New York, 1996.

[3] ANSI X 3.92, '*American National Standard for data Encryption Algorithm (DEA)*', American National Standards Institute, 1981

[4] J. Markus, P. S. Julien and Y. Moti, '*Scramble All, Encrypt Small*', Fast Software Encryption 1999, LNCS.

[5] M. M. Stephen, P. Mohammad and R. Allen, '*Encryption of Long Blocks Using a Short-Block Encryption Procedure*', IBM Technical Report, TR 29.2236, Research Triangle Park, North Carolina, March, 1997.

[6] M. M. Stephen, P. Mohammad, R. Allen and Z. Nev, '*Reversible Data Mixing Procedure for Efficient Public-key Encryption*', Computer and Security, V.17, n. 3, 1998, pp. 265-272.

[7] R. Revest, '*All-or-nothing Encryption and the Package Transform*', the 4th International Worksshop on Fast Software Encryption, Vol. 1267 of Lecture Notes in Computer Science, Springer-Verlag, 1997, pp.210-218.

[8] V. Boyko, '*On the security properties of OAEP as an all-or-nothing transform*', Proceedings of Crypto '99, Springer-Verlag, 1999.

[9] N. Ahmed, T. Natarajan and K. R. Rao, '*Discrete Cosine Transform*', IEEE Transaction on Computers, January 1974, pp. 90-93.

[10] R. J. Clarke, '*Digital Compression of Still Images and Video*', Academic Press, 1995.

[11] W. H. Chen, C. Harrison Smith and S. C. Fralick, '*A fast computational algorithm for the Discrete Cosine Transform*', IEEE Transaction on Communications, September, 1977, pp. 1004-1009.

Dual Level Access Scheme for Broadcasting Networks

Thumrongrat AMORNRAKSA
Multimedia Communications Laboratory, Department of Computer Engineering, Faculty of Engineering,
King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand

and

Peter SWEENEY
Centre for Communication Systems Research, School of Electronics, Computing and Mathematics,
University of Surrey, Guildford GU2 7XH, UK

ABSTRACT

This paper introduces a concept of dual level access scheme for broadcasting networks and describes an encoding scheme based on direct sequence spread spectrum technique for conveying some extra bits in the existing channel. By adding small amount of information into the encrypted signal to generate the output signal for transmission, any user at the receiver end is allowed to view this content, while only the users with the decryption key can view the encrypted content. In the decoding process, the added extra bits are first extracted from the received signal, and then used to recover the encrypted signal. The scheme was examined by simulation method and its performance was measured. Error control codes were applied to the extra bits before the encoding process so that the scheme's efficiency can be significantly improved, as seen in the experimental results. The scheme was also examined by transmitting the data through an AWGN channel to observe its performance when implemented in general applications. With the proposed scheme, the existing allocated bandwidth in the broadcast channel is utilized in a more efficient way.

Keywords: Direct sequence spread spectrum, Encoding method, Digital signal processing, Dual level access scheme, Broadcasting networks.

1. INTRODUCTION

An advantage of communications over the broadcasting network is that the transmitted signal from a source station can be received simultaneously by many destination stations. Digital TV broadcasting is one of the applications that uses this advantage. Since some digital TV programmes are pay-TV services, they will be encrypted before transmitting to every subscriber. Only the authorised subscribers who pay an extra fee can get access to those programmes. This technique does not give any value at all to other subscribers who have not paid for that particular programme. The allocated bandwidth is only used for broadcasting the encrypted signal to the authorised subscribers, which may be a small group compared to all subscribers in the network. It will be more efficient if we can devise an encoding scheme in which the authorised subscribers can access the encrypted signal and, at the same time, the other subscribers can receive something on the same channel, such as an advertisement. However, the scheme should not extend the existing allocated bandwidth.

In this paper, we describe such an encoding scheme that gives two levels of access to the subscribers in the broadcasting network. A techniques based on direct-sequence spread spectrum communications is implemented, together with error correcting codes, to add specific information (i.e. advertisements) to the access-limited signal, which is protected by encryption techniques. With this scheme, the allocated bandwidth for broadcasting is utilised more efficiently and more benefit is given to both the service providers, through advertising, and all subscribers in the network, since there will be programmes which they are not authorised to access but can see advertised. In Section 2, the method of constructing the encoding scheme is explained. The details of the encoding scheme are described, including the theory behind its operations. Section 3 describes the simulation model used to evaluate the performance of the scheme. The results from simulations and discussions are then given in Section 4. Finally, Section 5 provides some concluding remarks and directions for future work.

2. BACKGROUND

In spread spectrum (SS) communications [1, 2], a low-level wideband signal can easily be hidden within the same spectrum as a high power signal where each signal appears as noise to the other. The heart of these SS systems is a pseudo-random binary sequence (PRBS). For these direct sequence SS systems, the original baseband bit stream is multiplied by the PRBS to produce a new bit stream. Only those receivers equipped with the correct PRBS can decode the original message. At the receiver, the low level wideband signal will be accompanied by noise, and by using a suitable detector/demodulator with the correct PRBS, this signal can be squeezed back into the original narrow baseband. Because noise is completely random and uncorrelated, the wanted signal can easily be extracted [3].

part from applications in wireless communications, the rect sequence spread spectrum technique is widely used digital watermarking applications such as in [4, 5, 6]. y spreading the information bits and modulating them ith a PRBS, the spread signal can be obtained. This gnal is then embedded in the video signal, below the reshold of perception. The recovery of the embedded gnal can be accomplished by correlating the modified ideo signal with the same PRBS that was used in the rocess of constructing the spread spectrum signal. Correlation here is demodulation followed by summation ver the width of the chip-rate (the number of blocks over which each information bit is spread).

Based on this concept, an encoding scheme for dual level access was constructed to convey some information bits via an existing transmitted signal. Beside the benefit to the other subscribes, these extra bits can also be beneficial to various applications in many ways, for instance, they may be used to enhance the quality of the transmitted image or transport a control signal.

2.1. Description of the Scheme

In digital communication systems, channel coding is normally applied to the signal before transmission takes place, and this signal is considered as the encrypted signal in our encoding scheme. Using the spread spectrum technique as described in [5], the information bits will be embedded via an add operation to the encrypted signal after the channel coding process to obtain the resulting signal for transmission. Given a key to reproduce the same PRBS at the receiver's end, the information bits can be recovered. The encrypted signal can then be recovered by subtracting the information bits from the received signal. Any errors which occur at this stage such as communication channel errors will be detected and corrected by the channel decoder. The operation of the encoding scheme is shown in figure 1.

The basic steps of adding the extra bits to the encrypted signal are now described. We denote the sequence of extra bits we want to add to the encrypted signal by m_j, $m_j \in \{-1, 1\}$. This discrete signal is spread by a large factor cr, the chip-rate, to obtain the spread sequence (b_i), $b_i = m_j$, $j \bullet cr \le i < (j+1) \bullet cr$. The spread sequence is then modulated with a PRBS (p_i), $p_i \in \{-1, 1\}$ and added to the encrypted signal s_i, where each s_i block containing k bits, to yield the transmitting signal (s'_i), $s'_i = s_i + p_i \bullet b_i$.

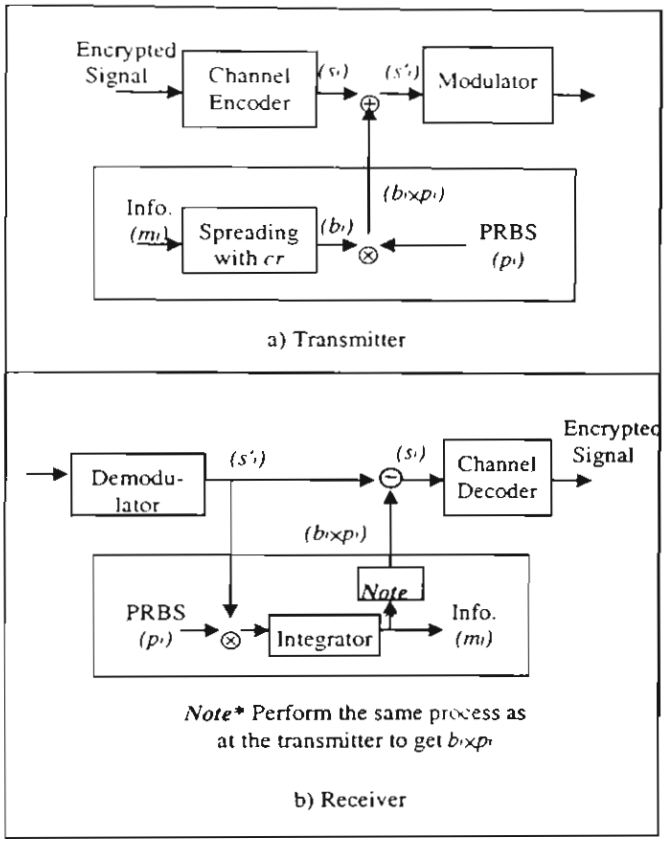

Figure 1. The operation of the Encoding Scheme

At the receiver, the recovery of the added bits is easily accomplished by multiplying the received signal with the same PRBS (p_i) that was used in the encoder. The summation over the correlation window, i.e. over cr, is as follows:

$$r_j = \sum_{i=j\bullet cr}^{(j+1)\bullet cr-1} p_i \bullet s'_i = \sum_{i=j\bullet cr}^{(j+1)\bullet cr-1} p_i \bullet s_i + \sum_{i=j\bullet cr}^{(j+1)\bullet cr-1} p_i^2 \bullet b_i \qquad (1)$$

The first term on the right-hand side of (1) vanishes if p_i and s_i are uncorrelated, and then $\sum_{i=j\bullet cr}^{(j+1)\bullet cr-1} p_i = 0$ [7]. Since $p_i^2 = 1$, r_j ideally becomes

$$r'_j = cr \bullet m_j \qquad (2)$$

and the recovered information bit $m'_j = sign\ (r'_j)$.

As an example, let the bit-rate of the encrypted signal be 10 Mb/s, the chip-rate $cr = 500$ and let the block size k be 4 bits. Then, the rate at which extra bits can be added after the channel coding process is 5 kb/s. With this bit-rate, the extra bits could be an image signal, for instance, in a compressed form transmitted every 30s or so. Hence, we can transmit the total bit-rate of 3.005 Mb/s within the existing bandwidth allocation of 10 Mb/s.

) increase the bit-rate of the extra bits, the chip-rate and e block size should be reduced. However, a smaller ock size implies a greater likelihood that subtracting the dra bits from the received signal will not give the ncrypted signal. In addition, a smaller chip-rate implies a reater likelihood of error in decoding the extra bits. To educe this latter likelihood of error, an error control code an be applied to the information bits before the preading process.

3. SIMULATION MODEL

Simulations were carried out using C programming language. The block size k was varied from 2-7 bits to represent up to 128 values. The chip-rate was varied from 0 to a value that gives no error in the extracted information. However, it is obvious that some results from the addition of s_i and $p_i \bullet b_i$ are out of range of the values that the encrypted signal can represent, and thus more bandwidth will be required for transmitting the output signal. In order to keep the output bit-rate constant, the addition of s_i and $p_i \bullet b_i$ is performed as follows;

$$s'_i = s_i, \text{ if } s_i = 0 \text{ and } p_i \bullet b_i = -1,$$
$$\text{or } s_i = (2^k - 1) \text{ and } p_i \bullet b_i = 1$$
$$\text{Otherwise } s'_i = s_i + p_i \bullet b_i \qquad (3)$$

When the error control codes are applied to the extra bits, it will of course reduce the main throughput by a factor k/n, which one may think that this may be difficult to compensate by a smaller value of chip-rate in the decoding process. To demonstrate that the error control codes can improve the performance of the encoding scheme, various codes are applied to the extra bits before performing the spreading process, and theirs performances are then compared to the one without the codes. For example, Reed Solomon codes, Binary BCH code, Golay code and Convolutional code with rate 1/2 and $K = 7$. Consult [8, 9], for those who are not familiar with the subject.

However, at this state of our simulations, the encoding scheme will be performed in an error-free communication channel. That is, the errors that occurred in the encrypted signal came solely from the need to remain within the bandwidth of the transmission channel. The objective for doing this is to focus on only the errors that occur in the extracted information bits, which are mainly related to the performance of the scheme. In addition, generating the data to be used in the simulations can be accomplished by using a random number generator. One that produces a uniform distribution of numbers on the interval 0 to a maximum value is provided by a function *rand( )* in C language. Using this function, we can say that any value less than half of the maximum value is a zero; any value greater than or equal to half of the maximum value is a one, and then input into the constructed encoding scheme as described in figure 1.

After the proper code that gives the best performance is found, the proposed scheme will be simulated in a communication channel. At this step, an Additive White Gaussian Noise (AWGN) channel is chosen since it is a type of noise that most communication systems encounter [10]. An error control code i.e. convolutional code with rate 1/2 coding is also applied to the encrypted signal in order to observe the performance of the proposed scheme when implemented in practice. The simulation model used in the experiments is shown in figure below.

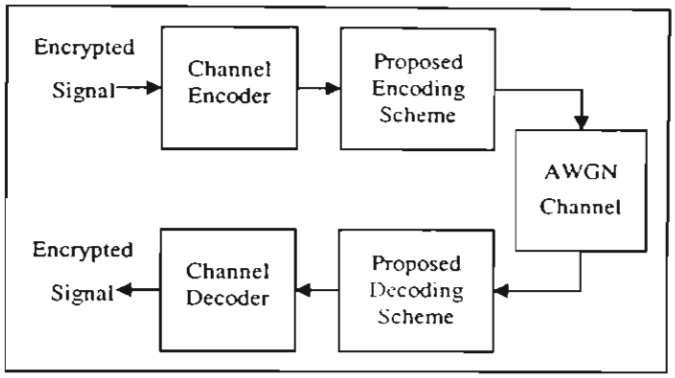

Figure 2. Simulation model in an AWGN channel

Adding noise to the transmitted channel symbols involves generating Gaussian random numbers, scaling the numbers according to the desired energy per symbol to noise density ratio, E_S/N_0, and adding the scaled Gaussian random numbers to the channel symbol values. For the uncoded channel, $E_S/N_0 = E_b/N_0$, since there is one channel symbol per bit. However, for the coded channel, $E_S/N_0 = E_b/N_0 + 10\log_{10}(k/n)$, where k and n are the number of input symbols and output symbols respectively. For example, for rate 1/2 coding, $E_S/N_0 = E_b/N_0 + 10\log_{10}(1/2) = E_b/N_0 - 3.01$ dB. The results from this simulation will be shown by a plot of the BER versus the E_b/N_0 as previously described.

4. SIMULATION RESULTS AND DISCUSSIONS

From the simulation results, the smallest chip-rate with no errors after the extracting process are shown in table 1.

Table 1. Values of the chip-rate with no errors after the extracting process, at different block sizes

Block Size k	2	3	4	5	6	7
Chip-rate cr	46	110	455	1100	4150	12000

s Table 1 shows, the smaller the block size, the larger alue the chip-rate required to recover the information its correctly. For these block sizes, other values of the hip-rate considered resulted in different values of Bit Error Rate (BER) in the extracted information bits, and hese values and the underlying line are shown in the igure below.

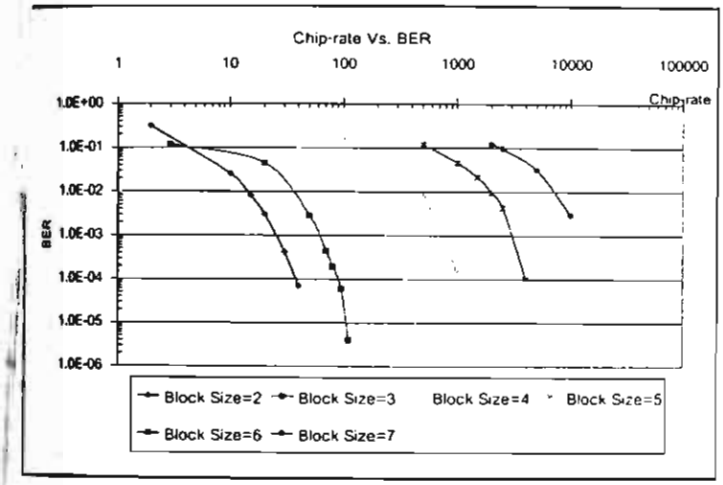

Figure 3. Bit error rate of extracted extra bits at different block sizes

From the figure 3, it can be seen that a larger block size needs a bigger chip-rate to retain the same BER. In addition, since one single bit error in the extracted information causes error propagation in the encrypted signal, any value other than a large chip-rate will result in a large BER. To further improve performance of the scheme, the error control codes were applied to the information bits before the spreading process. This reduced the amount of data rate to be embedded in the encrypted signal by a factor *k/n*. However, the amount of chip-rate required in the decoding process was decreased, and this resulted in an improvement in efficiency for the entire system.

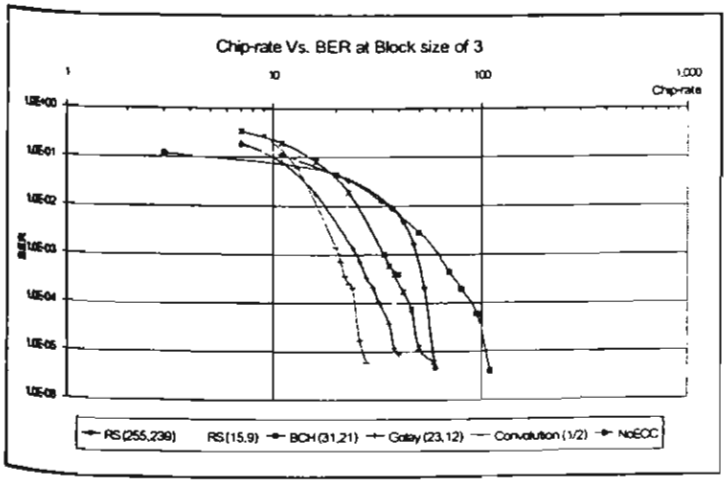

Figure 4. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 3

The smaller values of the chip-rate, when applied the error control codes, that gave different values of BER in the extracted extra bits are shown in figures 4 through 7 for the block sizes of 3, 4, 5, 6 respectively.

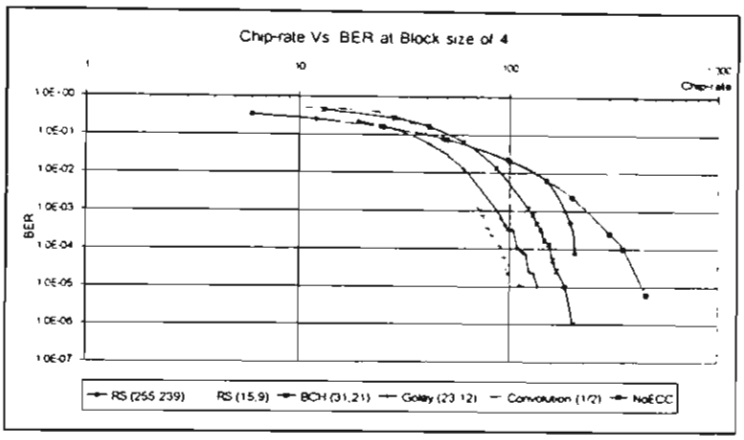

Figure 5. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 4

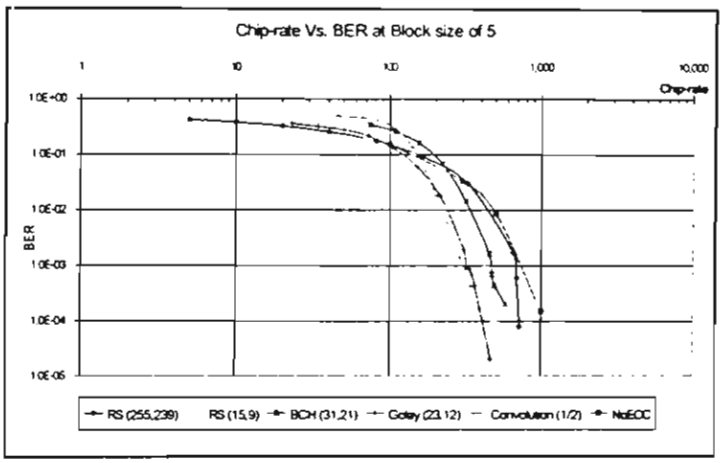

Figure 6. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 5

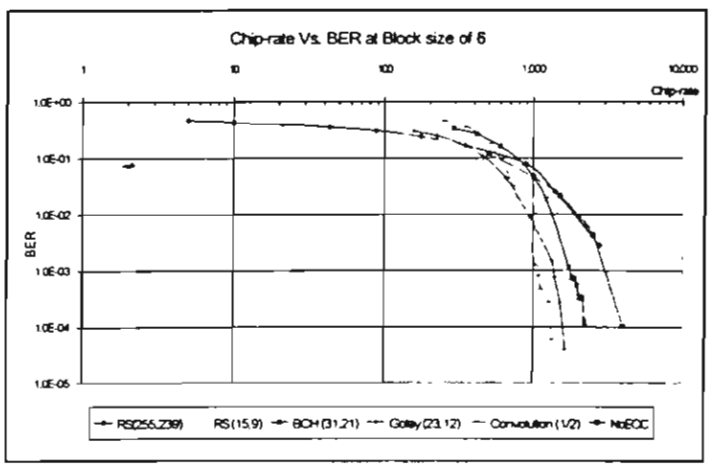

Figure 7. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 6

obviously shows, from the figures above, that the error ntrol codes provide the smaller values of the chip-rate. can also be seen the convolutional code gave the best rformance, compared to the others, and hence was used the next experiments. To illustrate the benefits of using ror control codes, the number of encrypted data ymbol) which is used to convey the extra bits is plotted gainst the BER. Figure 8 shows the performance omparison of the encoding scheme with and without the ror control codes at the block size of 4.

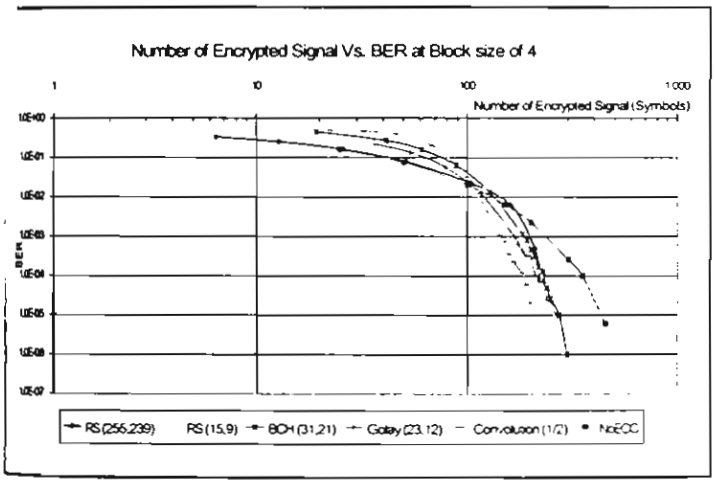

Figure 8. Performance comparison between the scheme with and without the error control codes at the block sizes of 4

Table 2 shows an example of the total amount of bandwidth saved by the use of the RS (15, 9) code at different block sizes. The results are presented in the form of efficiency improvement (%) of the scheme with the RS (15, 9) code, compared to the one without the code.

Table 2. Summary of efficiency improvement of the scheme with the (15, 9) RS code at different block

Block Size k	2	3	4	5	6	7
Efficiency (%)	33.3	33.3	39.1	38.9	46.1	44.2

It is clear that the larger the block size, the higher the efficiency of the scheme. For that reason, the error control codes can be very useful when the scheme is operated with a large block size. To observe the performance of the scheme when implemented in the AWGN channel, the simulations were conducted according to the model in figure 2. In the following figure 9, a plot of the BER versus the E_b/N_0 for the scheme using the convolutional code with rate 1/2 is given.

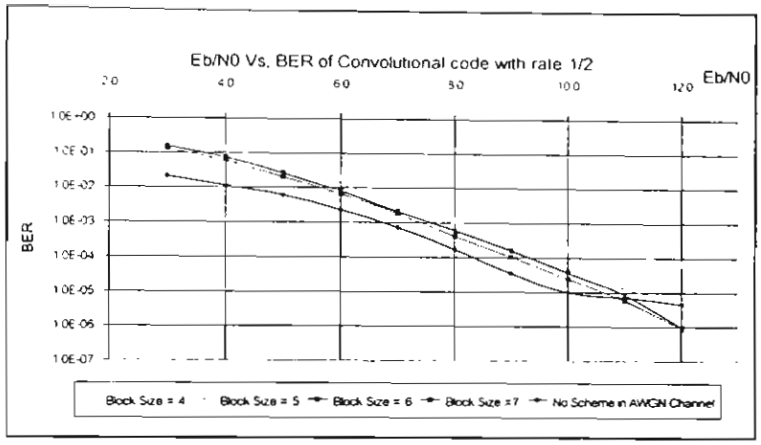

Figure 9. Performance comparison when the scheme is implemented in the AWGN channel at various block sizes

From the figure, it can be seen that errors occurred at the output of the encrypted signal were larger than the one from pure AWGN channel. This is because of the unrecoverable errors remained in the data after decoding process. However, when the value of E_b/N_0 was increased the BER became smaller, especially with the larger block size. It can be noticed that the scheme operated with a larger block size gave better performance. Nevertheless, the value of the chip-rate required for large block size is enormous and this choice should be carefully considered. From the obtained result so far, the proposed scheme is not fit well when implemented in the AWGN channel.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have shown a method of constructing an encoding scheme for dual level access to broadcasting network, based on the direct sequence spread spectrum technique. We have also shown experimentally and analytically that the scheme's performance was improved by applying the error control codes to the information bits before the encoding process. Our approach has showed an idea of how to utilize the existing allocated bandwidth in a more efficient way. For implementation aspect, when all parameters are properly selected, the proposed encoding scheme can be fitted with any application that employs channel coding, so that any errors which occur at the receiver's side, whether communication channel errors or errors resulting from the decoding process, will be detected and corrected by the channel decoder. Further work can be carried out by simulating the scheme in the presence of noises in various communication channels such as Rayleigh fading channels, or in some specific applications such as Digital Video Broadcasting (DVB), and observe its performance and reliability.

6. ACKNOWLEDGMENT

The authors would like to thank the Thailand Research Fund (TRF) for financial support throughout this work (Funding Code: PDF/27/2543).

7. REFERENCES

[1] R. Pickholtz, D. Schilling and L. Millstein, "Theory of Spread Spectrum Communications A Tutorial", *IEEE Transaction on Communication*, Vol. COMM-30, 1982, pp. 855-884.

[2] W. C. Y. Lee, "Spectrum Efficiency in Cellular", *IEEE Transactions on Vehicular Technology*, Vol. 38, No. 2, May 1989, pp. 69-75.

[3] R. C. Dixon, "*Spread Spectrum Systems with Commercial Applications 3rd Edition*", John Wiley & Son Inc., New York, 1994.

[4] M. George, J-V. Chouinard and N. Georganas, "Digital Watermarking of Images and Video using Direct Sequence Spread Spectrum Techniques", *Proceeding of the 1999 IEEE Canadian Conference on Electrical and Computer Engineering*, Vol. 1, 1999, pp. 116-121.

[5] F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video", *Signal Processing*, Vol. 66, no. 3 (Special issue on Watermarking), May 1998, pp. 283-301.

[6] Cox, J. Kilian, T. Leighton and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia", *IEEE Transactions on Image Processing*, Vol. 6, No. 12, December, 1997, pp. 1673-1687.

[7] H. Stark and J. W. Woods, "*Probability, Random Variables and Estimation Theory for Engineers*", Prentice Hall, Englewood Cliffs, N.J. 1986.

[8] P. Sweeney, "*Error Control Coding: An Introduction*", Prentice Hall, London, 1991.

[9] S. Lin and D. J. Jr. Costello, "*Error Control Coding: Fundamentals and Applications*", Prentice-Hall, Englewood Cliffs, NJ, 1983.

[10] A. M. Michelson and A. H. Levesque, "*Error Control Techniques for Digital Communication*", New York: John Wiley & Sons, 1985.

Access Limited Coding for Digital Video Streams

Wachiwan KANJANARIN, Pipat SUPASIRISUN and Thumrongrat AMORNRAKSA

Multimedia Communications Laboratory, Department of Computer Engineering, Faculty of Engineering,

King Mongkut's University of Technology Thonburi, Bangkok 10140, THAILAND

ABSTRACT

Generally, encryption techniques are applied to a distributed digital video stream to prevent unauthorized viewing. This paper proposes a new scrambling scheme, which is used in conjunction with ordinary encryption techniques, for protecting the digital video streams. The proposed scheme reduces the computation time and complexity of the entire system, compared to traditional encryption methods i.e. the whole video stream is totally encrypted. A strong collision one-way hash function and a pseudo-random number generator are applied to the video stream before entering the encryption process. Our scheme provides less complexity in computational process by encrypting only a single block, and at the mean time offers the same level of security as ordinary encryption methods do. In addition, the proposed scheme is suited to be implemented some applications that requires high bit rate.

Keywords: Encryption, Scrambling Method, Digital Video, Hash Function and Pseudo-random Number Generator.

1. INTRODUCTION

Digital video streams are often transmitted through insecure public channels. In some cases, such as Pay-TV system, video conferencing or video purchase via Internet, the senders can prevent their video streams from unauthorized viewing by using scrambling techniques to alter those video streams (referred to as an encryption technique in digital systems) . However the video information has much higher bit rate than other types of information such as text which is used in military or commercial banking systems [1]. This poses a limitation on encryption algorithm that could be applied to video information. For example, It would be difficult to implement the complicated types of symmetric or asymmetric encryption algorithms with video information because both algorithms require complicated and time-consuming operations in encryption process.

Usually the security of asymmetric algorithms relies on complex mathematical computations which make it even more time-consuming than the symmetric ones. An example of asymmetric encryption is RSA (Rivest-Shamir-Adelman) algorithm [2], which requires more computational time than the well-known symmetric block encryption, DES (Data Encryption Standard), used in digital television systems, by about 100 to 1000 times [3, 4]. Therefore, we can often see the use of complicated encryption algorithms in hybrid form. A frequently seen example of hybrid cryptosystems is the use of asymmetric algorithm to encrypt the secret key of a symmetric algorithm, and this secret is then used to encrypt the information directly.

In this paper, we first propose a digital video scrambling technique, which separates selected data into blocks. Then we propose a new design tool for block encryption used for digital video scrambling. The design purposes are to reduce overall processing time by decreasing the encryption process of the entire data into a mere one single block, and at the same time still maintaining the security level of the system. The proposed method can possibly be applied with any application which requires a high secure asymmetric encryptions for high bit-rate video streams.

2. BACKGROUND

Security algorithm for video information has first been implemented on TV systems to prevent the broadcast programs from unauthorized viewing. In the system, a source station broadcasts the scrambled video information to various receivers simultaneously via a public channel. At the receivers' end, each receiver has a device that enables the descrambling process of the scrambled signal back to the original video information [5].

For the digital video information, the encryption techniques are used in the same way as the scrambling techniques, and they are hence referred to as digital scrambling techniques in this paper. Basically, the technique starts with the process of

ncrypting a plaintext message to produce a iphertext. This ciphertext is then broadcast via a ublic channel to all receivers. Only the authorized eceivers with a key can decrypt this ciphertext and iew the original plaintext. Normally, there are two inds of digital scrambling technique that are ommonly used [6].

Block encryption: the plaintext is first separated into fixed size blocks. These blocks are then encrypted independently from the others. The resultant ciphertext is the same size as the plaintext input. One disadvantage of this encryption algorithm is that when the ciphertext is transmitted through a noisy channel, single error will propagate and make the recovered plaintext deteriorated. An example of block encryption is the popular DES algorithm [1].

Stream encryption: In this technique, the encryption is performed bit by bit or byte by byte and the plaintext is XORed with key k_i calculated from a pseudo-random number generator (PRG). The encryption uses a secret key as a seed of the PRG. This technique is more resistant to channel errors than the block encryption. However, the stream encryption is less secure to cryptanalysis due to the following rationale. It is highly possible for a known plaintext attack to occur, especially in the digital TV system, where the cryptanalysts can actually obtain plaintext from ciphertext without knowing the secret key. By subscribing to a pay-TV program, the cryptanalysts can know easily both the plaintext and the corresponding ciphertext. Thus, they can efficiently break the encryption based on PRG built by liner shift registers [1].

At present, the efficiency of data transmission and error correction technique has improved making the possibility of channel error smaller than before. Because of this reason, the block encryption has increasingly become more interesting. Compared to the conventional encryption such as that of text information, digital scrambling of video information has an additional important characteristics, that is transparency.

Transparency process is such that the ciphertext still looks somewhat like the original plaintext. In many applications, the encrypted information is required to be quite transparent: encrypted but still viewable to a particular degree. For example, the Pay-TV operators do not always want to prevent unauthorized viewers to receive their programs. Instead they want these viewers to see the poor version of the programs to promote paying subscription [7].

There are many interesting methods that have been applied to improve the performance of block encryption for use in different applications [8, 9, 10, 11]. A common concept is to remove the information pattern from the original message making each output bit appeared random. In this concept, each message would undergo a pre-process or masking process, which is reversible. The whole procedure begins with a message being divided into fixed size blocks called formatting blocks. These blocks then go through a masking process that makes them appear randomly. These processed blocks are now called masked blocks and have the same amount as the formatting blocks. The masked blocks altogether are called a masked message. Every bit in a masked message is equally important and is related to the original input message. If any one bit is missing or there is a bit error in the reversible masking process, one will not be able to recover the original message.

All-or-nothing transform uses a similar concept as the masking process to increase its tolerance to the brute force attacks. In this transform, a message undergoes a masking process in which all output is encrypted block by block making the recovering process difficult. To obtain a block of original message, one must decrypt all the ciphertext. This concept increases the competency of block encryption, however, it also increases the processing duration by two to three times when compared to direct encryption of the message [9].

Another design that has been proposed to reduce the encryption complexity is the Scramble All, Encrypt Small [10]. This design is able to maintain security level of the encryption while reducing the length of message encrypted. It starts by calculating a hash value from the entire message. Then it takes the original message that is now concatenated with the calculated hash value, through a process. This process makes every bit of output related with every bit of the original message. Next, it encrypts a portion of the output. In the reversible process, one must know every bit of the output before encryption. The process includes the work of a strong collision resistant one-way hash function.

Although the Scramble All, Encrypt Small has a high security level, the number of rounds required in hash function is high while the number of output bits is even higher than that of the original input,

ue to the concatenated hash value. This lengthens ie calculation time and, in some applications, the ıcreased bits become a burden to the system.

3. THE PROPOSED SCHEME

We now present a scrambling method for digital video streams that decreases the complexity required in the encryption process, compared to encryption of the entire video streams. Our intention is to reduce processing time and to open up possibilities of implementing highly secure encryption algorithm such as asymmetric encryption to digital video streams.

We briefly introduce our scheme, which comprises of three major steps, as follows: Firstly, the process starts with selecting the video bit streams to undergo scrambling process m bits at a time. These bits are then separated into fixed size blocks called formatted blocks. Each formatted block contains k bit and there are n blocks in the process labeled F_1, F_2, F_3... to F_n. Note that the blocks must be in even quantity. Secondly, we put the formatted blocks through a masking process to produce masked blocks (M_1, M_2, M_3... to M_n), and the masked blocks will have the same size as the formatted blocks. Lastly, we randomly choose certain blocks or certain bits in each block for encryption.

3.1. Selecting and separating video bit streams

In the first step, we present how the scheme selects the uncompressed video streams for scrambling process. We use the uncompressed streams for the following reasons. First, the streams are independent of compression algorithm. Second, they would have the transparency that is required in some applications. However, the compressed video streams could still be used in our scheme by separating m bit streams into n fixed size blocks. Figure 1 shows the selecting and separating process of m bit streams into n fixed size blocks.

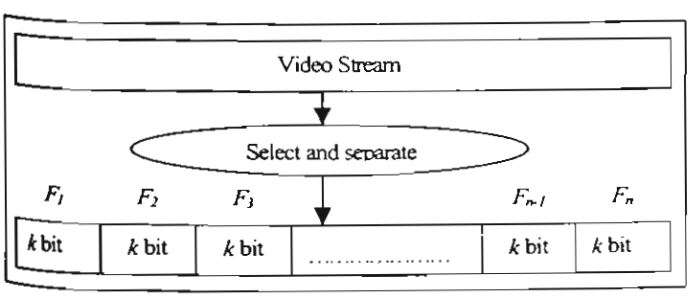

Figure 1: Bit selection and separation process

The uncompressed video streams comprise of many connected images. Each image contains many pixels, and each of them contains 24 bits, in which each eight-bit group represents the intensity of red, green and blue.

In an application that requires transparency or scrambling rapidity, we need not scrambling all bits in each pixel. Scrambling nine to twelve bits from each pixel would be sufficient. For example, in applications that require transparency, we would choose only twelve bits from each pixel by taking the four least significant bits from each color only. On the other hand, in applications that does not desire transparency, we would choose only twelve bits from each pixel by taking the four most significant bits from each color.

All selected bits from the video streams are separated into n blocks of an equal size. These n blocks then undergo a masking process to become a masked message. A portion of the masked message then goes through an encryption process. Finally, the entire masked message, including the encrypted part, is distributed back to their original position of uncompressed video bit streams.

3.2. Marking process

Marking process constitutes two important functions. First, it has a strong collision resistant one-way hash function (h), which calculates k-bit size output from arbitrary size input (for example, in SHA-1, k is equal to 160 bits). Second, it has a high-speed pseudo-random number generator (G), which calculates k-bit size output from k-bit size input. We try to minimize the responsibility of the one-way hash function while making the most use to the pseudo-random number generator. This is done to minimize the computation time in the marking process and maintain the high security level at the same time. There are six steps in the marking process.

1. Use a pseudo-random number generator (G), initial vector (IV) and formatted block to create F'_1, F'_2, F'_3,..,F'_n. Then use this result, together with a pseudo-random number generator(G), initial vector(IV) again, to define F''_1, F''_2, F''_3,.....,F''_n

$$F'_1 = F_1 \oplus IV$$

$$F'_i = F_i \oplus G(F'_{i-1}) \qquad \text{(Eq.1)}$$

$$F''_n = F'_n \oplus IV$$

$$F''_{i-1} = F'_{i-1} \oplus G(F''_i) \qquad \text{(Eq.2)}$$

where $i = 2,3,..n$

Concatenate F''_1, F''_2,... ,F''_n together. Then separate them into two equal parts, which we call *FR* and *FL*.

3. Define FR'_i as the result of a pseudo-random number generator (G) and a hash value calculated from a strong collision resistant one-way hash function (h) and *FR*.

$$FR'_1 = G(h(FR))$$
$$FR'_i = G(FR'_{i-1} \oplus i\text{-}1) \quad \text{(Eq.3)}$$
where $i = 2,3,..n/2$

4. Set *FR'* as the concatenation of FR'_i. Then we denote and compute *ML* using *FR'* and *FL*.

$$FR' = FR'_1 \| FR'_2 \| .. \| FR'_{n/2} \quad \text{(Eq.4)}$$
$$ML = FR' \oplus FL \quad \text{(Eq.5)}$$

5. Define ML'_i as the result of a pseudo-random number generator (G) and a hash value calculated from a strong collision resistant one-way hash function (h) and ML.

$$ML'_1 = G(h(ML))$$
$$ML'_i = G(ML'_{i-1} \oplus i\text{-}1) \quad \text{(Eq.6)}$$
where $i = 2,3,..n/2$

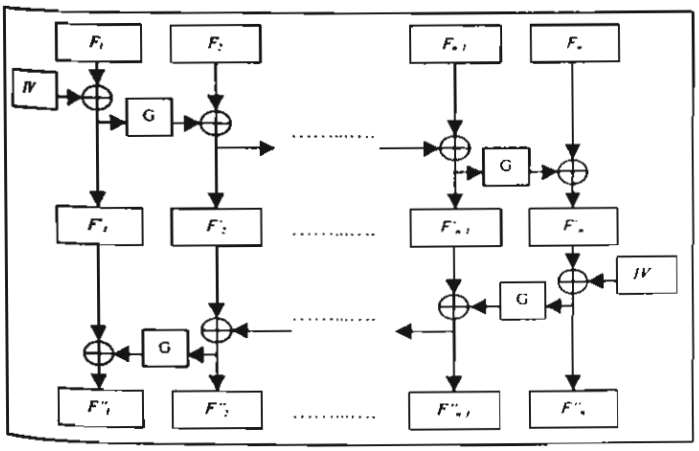

Figure 2: Step 1 of marking process

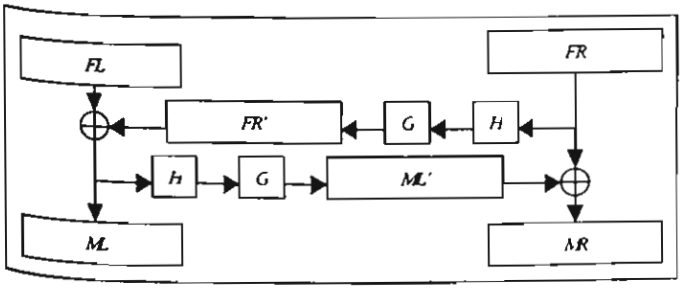

Figure 3: Steps 3 to 6 of marking process

6. Set *ML'* as the concatenation of ML'_i. Then we define and compute *MR* using *FR* and *ML'*.

$$ML' = ML'_1 \| ML'_2 \| .. \| ML'_{n/2} \quad \text{(Eq.7)}$$
$$MR = ML' \oplus FR \quad \text{(Eq.8)}$$

3.3. Encryption process

For encryption process, we can choose a single marked block from M_1 to M_n in either *ML* or *MR* to encrypt. Another option is to randomly select bits from marked message (*ML* concatenated with *MR*) by using a pseudo-random number generator to locate the position of these bits. Eventually, all of these processed bits are distributed back to their original position of the video bit streams.

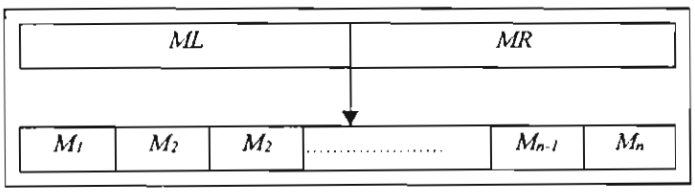

Figure 4: Position of marked blocks

3.4. Descrambling process

The descrambling process in our scheme is the reverse of the scrambling process. It starts by selecting the scrambled bits and then performing decryption only on the encrypted parts. After that the reversible marking process occurs. The output bits of this process is distributed back to their selected position of the video bit streams making the streams of images visible and clear again.

3.5. The reverse of marking process

1. Concatenate M_1 to M_n altogether and divide the sum into *MR* and *ML*. Then compute *FR* from *MR* and *ML*.

$$ML'_1 = G(h(ML)$$
$$ML'_i = G(ML'_{i-1} \oplus i\text{-}1) \quad \text{(Eq.9)}$$
where $i = 2,3, n/2$

$$ML' = ML'_1 \| ML'_2 \| .. \| ML'_{n/2} \quad \text{(Eq.10)}$$
$$FR = ML' \oplus MR \quad \text{(Eq.11)}$$

2. Compute *FL* from *FR* and *ML*

$$FR'_1 = G(h(FR)$$
$$FR'_i = G(FR'_{i-1} \oplus i\text{-}1) \quad \text{(Eq.12)}$$
where $i = 2,3,..n/2$
$$FR' = FR'_1 \| FR'_2 \| .. \| FR'_{n/2} \quad \text{(Eq.13)}$$
$$FL = FR' \oplus ML \quad \text{(Eq.14)}$$

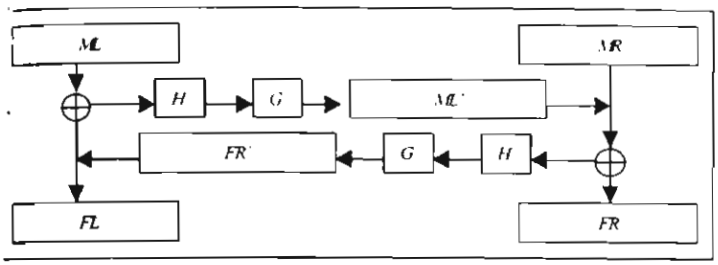

Figure 5: Steps 1and 2 of the reverse of marking process

3. Put FR and FL together and separate the sum into n blocks (F''_1, F''_2, F''_3,.....,F''_n). Then use IV and F''_1, F''_2, F''_3,.....,F''_n to recover the original video information. (F_1, F_2, F_3 to F_n.)

$$F'_n = F''_n \oplus IV$$
$$F'_{i-1} = F''_{i-1} \oplus G(F''_i) \quad \text{(Eq.15)}$$
$$F_1 = F'_1 \oplus IV$$
$$F_i = F'_i \oplus G(F'_{i-1}) \quad \text{(Eq.16)}$$

where $i = 2,3,\ldots n$

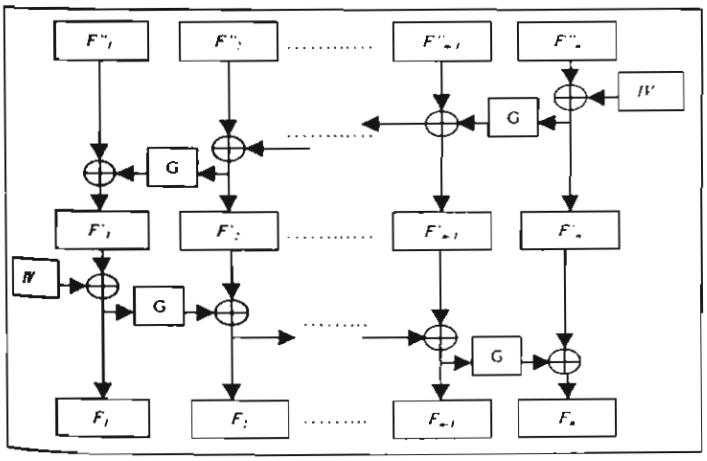

Figure 6: Step 3 of the reverse of marking process

4. PERFORMANCE AND SECURITY ANALYSIS

Our scheme, when applied to uncompressed video streams, can decrease encryption work to a single block only. This allows the scheme to be implemented on a highly secure device such smart card, which has a low calculation capacity. In fact, our scheme allows the use of smart cards in a more efficient way. Instead of using the card for keeping the secret key only, we use it for performing encryption because the encryption process is reduced to just one single block. In addition, our scheme lets the users control transparency of the encrypted streams by way of selecting the encrypted bits. Moreover, the scheme can be used with the compressed video streams and/or non-video information such as text file. For these types of data, we can divide bit streams into k-bit blocks and perform marking process and encryption.

When comparing a time used in marking process, which comprised a strong collision resistant one-way hash function and a high-speed pseudo-random number generator and a time used to encrypt one single blocks, our scheme uses less time than the total time used to encrypt the entire message in block by lock fashion. The scheme is not dependent on any encryption algorithm so it can be used with any encryption algorithm. This advantage makes our scheme very flexible.

Moreover, no bit in the formatted blocks can be determined unless every bit in the masked blocks is known. Also, as a result of step 1 and 2 of marking process, each bit in the masked block is a function of each and every bit in the formatted blocks. Both advantages enable us to choose to perform encryption to any part of the marked message while still holding same security level as encrypting the entire original message using cipher-block chaining (CBC mode)[9].

Security of our scheme depends on the choice of encryption algorithm. Chosen plaintext attacks are not possible with our scheme because the initial vectors are different for every scrambling process. When the same video stream is scrambled twice, it will produce completely different marked messages each time. Therefore, the attackers cannot compare the scrambling outputs that resulted from the same original video streams.

We can consider the application of initial vectors as a form of secret keys in the system. A change in initial vectors can happen as frequently as every m bit. The increase in secret key change frequency increases the system security [12]. A change in only one bit in the marked message would make the output of reversible marking process look completely different from the original video streams. This makes it easier for attackers to try to attack the encrypted part instead of the entire marked message, which contains a higher number of bits. Therefore, the system security relies on the choice of encryption algorithm. To break the system, attackers must break the selected encryption algorithm without any additional knowledge.

5. CONCLUSIONS

iere are several major requirements when rambling the digital video streams: the security vel, the computation time required and the mplexity of scrambling devices. Our scheme has itisfied all three requirements by using the existing evice efficiently, such as the smart card. It also rovides sufficiently high security level while educing the computation time when compared to ncryption of the entire video streams. Because we an choose among different algorithms for our ystem, this makes it flexible and adaptable to echnological changes and future developments.

7. ACKNOWLEDGMENT

The authors thank the Thailand Research Fund for partly supporting this research work (PDF/27/2543).

6. REFERENCES

[1] B. M. Macq and J. J. Quisquater, *'Cryptology for Digital TV Broadcasting'*, Proceedings of the IEEE, Vol. 83, No. 6, 1995, pp.944-957.

[2] R. L. Rivest, A. Shamir and L. Adleman, *'A Method for Obtaining Digital Signatures and Public Key Cryptosystems'*, Communications of the ACM, Vol. 21, No. 2, February, 1978, pp. 120-126.

[3] ANSI X.3.92, *'American National Standard for data Encryption Algorithm (DEA)'*, American National Standard Institute, 1981.

[5] B. Schneier, *'Applied Cryptography'*, Wiley: New York, 1996.

[6] R. F. Graf and W. Sheets, *'Video Scrambling & Descrambling for Satellite & Cable TV'*, the 4th Edition, Howard W. Sams & Company, 1989.

[7] W. Mooij, *'Advances in Conditional Access Technology'*, International Broadcasting Convention, No. 447, September, 1997, pp. 461-464.

[8] S. R. Ely and S. R. Shuttleworth, *'Conditional Access Scrambling Techniques for Terrestrial UHF Television Broadcasts'*, International Broadcasting Convention, 1988 pp. 318-322.

[9] M. Bellare and P. Rogaway, *' Optimal Asymmetric Encryption '*, EUROCRYPT 94, No. 950, Springer- Verlag, 1994, pp. 92-111.

[10] R. Revest, *'All-Or-Nothing Encryption and the Package Transform'*, the 4th International Workshop on Fast Sortware Encryption, Vol. 1267 of Lecture Notes in Computer Science, Springer- Verlag, 1997, pp. 210-218.

[11] J. Markus, P. S. Julien and Y. Moti, *'Scramble All, Encrypt Small'*, Fast Software Encryption, 1999.

[12] D. Johnson and S. Matyas, *'Asymmetric Encryption : Evolution and Enhancements'*, CryptoBytes, Vol 2, No.1, Spring, 1996

[13] F. K. Tu, C. S. Laih, and H. H. Tung, *'On Key Distribution Management for Conditional Access System on Pay-TV System'*, IEEE Transactions on Consumer Electronics, Vol.45, February, 1999, pp. 151-158.

Scrambling and Key Distribution Scheme for Digital Television

Wachiwan Kanjanarin and Thumrongrat Amornraksa
Multimedia Communications Laboratory, Department of Computer Engineering
King Mongkut's University of Technology Thonburi, Bangkok, Thailand
e-mail: wachiwan@hotmail.com and t.amornraksa@cpe.eng.kmutt.ac.th

Abstract

The scrambling scheme is a part of the conditional access system (CAS) that is used to prevent unauthorized access to Pay-TV systems. In this paper, we propose a new scrambling scheme and key distribution scheme. The scrambling scheme is used in conjunction with ordinary encryption techniques, for protecting the digital video streams from unauthorized viewing. A hash function and a pseudo-random number generator are used to prepare the video stream before being encrypted. The proposed scheme helps reduce computational time and complexity while providing the same level of security as encrypting the entire video stream. In addition, the proposed secure key distribution scheme can be used with any scrambling scheme e.g. with our scrambling scheme. By using the Chinese Remainder Theorem (CRT) for distributing parameters in the scrambling process, the security of the scheme can be increased. Moreover, our scheme can prevent two common problems, namely smart card cloning and McCormac Hack.

1. Introduction

Generally, digital video streams are transmitted through insecure public channels. In some cases, such as Pay-TV system, video conferencing or video purchase via the Internet, the senders would like to prevent unauthorized viewing of their video streams. Pay-TV service providers employ Conditional Access System (CAS), which uses scrambling, to protect their investments [1,2]. For the digital system, we implement encryption with video streams as the scrambling scheme. In general video information has much higher bit rate than other types of information such as text information that is used in the military or commercial banking system [1]. This poses a limitation on encryption algorithm that could be applied to video information. It is difficult to implement more complicated types of encryption algorithms on video information because it would be too computationally complex and time-consuming. The CAS is performed inside the decoder box (sometime called set top box) and the smart card. Therefore, the algorithm in decoder box and smart card and the process between both components have direct effects on system security. McCormac Hack and smart card cloning are problems that happen when one card can be used in different decoder boxes of the same type.

In this paper, we present a CAS for Pay-TV systems that can be separated into two parts: scrambling scheme for digital video information and scrambling key distribution scheme. First we propose a new design tool for block encryption used for digital video scrambling. The design purposes are to reduce overall processing time and to decrease the encryption complication by encrypting only one single block while still maintaining the same security level as encrypting the entire message. In the second part we present a reliable scrambling key distribution scheme that ensures the detection of any possible fraud in the decoder box. Our scheme can solve McCormac Hack and smart card cloning problems that can happen to systems that use both smart cards and decoder boxes. We also propose an authentication between the smart card and the decoder box, which helps to confirm that only authorized subscribers who have the authorized smart card and decoder box can receive the proper scrambling key.

2. Background

In general, Pay-TV Systems use the CAS to improve their security [2]. In the CAS, only the authorized subscribers who paid a subscription fee can watch the program. The security of a CAS depends merely on the scrambling algorithm and the scrambling key distribution scheme [3].

2.1. Scrambling method

Scrambling is a cryptographic algorithm on video information using a secret key, called scrambling key (sometimes called "control word"). The algorithm makes such signals unwatchable to unauthorized viewers. The

authorized subscribers need this scrambling key to descramble the received signal and reconstruct the original program [4]. For the digital video information, encryption is used to as the scrambling scheme. There are two kinds of encryption that are commonly used [1].

2.1.1. Block encryption. Here, plaintext is separated into blocks of fixed size. These blocks are then encrypted independently from one another. The resulting ciphertext is the same size as the plaintext input. An example of block encryption is the popular DES algorithm [5], which has been used in digital television encryption [6].

2.1.2. Stream encryption. Here, encryption is performed bit by bit or byte by byte and plaintext is X-ORed with key K_i calculated from a pseudo-random number generator (PRG). The encryption uses a secret key as a seed of PRG.

For digital television, stream encryption is less secure than block encryption because it is more vulnerable to known-plaintext attacks. In stream encryption, cryptanalysts can easily recover both the plaintext and the corresponding ciphertext, which will allow them to break the encryption, based on PRG built by linear shift register [1]. The cryptanalysts can receive the plaintext by subscribing to the digital Pay-TV service. Because this weakness in stream encryption, researchers have become more interested in block encryption.

2.2. Scrambling key distribution

Authorized subscribers need scrambling keys to descramble the scrambled programs. The scrambling keys are secretly sent to all subscribers so unauthorized parties cannot see them. CAS security depends on scrambling key distribution, which is a part of CAS. As a result, the choice of scrambling key distribution scheme is as important as the choice of cryptographic algorithm used in scrambling. In general, the scrambling key is encrypted using an encryption algorithm [7]. Then the ciphertext of a scrambling key is sent together with program signals that are scrambled with the key. There have been many proposals for key hierarchy models for key distribution. These models enable efficient key management so Pay-TV providers are able to refresh scrambling key as often as they desire in order to ensure a high level of system security [2,8].

2.3. Implementation of smart cards in Pay-TV system

For many years, smart cards have been used along with decoder boxes to extracts certain important information from the box. A smart card is replaceable at anytime by operators and is inserted into the decoder box for operation [9]. In general, a card can be used for any different decoder box of the same type. This results in two frequently encountered problems.

2.3.1. McCormac Hack. This problem occurs when the data line from smart card to decoder box is tapped and the data from this line is directed to another decoder box that acts as if it has the same smart card inside.

2.3.2. Smart card cloning. In this problem, a legal smart card is copied to make many illegal cards with the same ID number. These copies can be used in any decoder box of the same type allowing unauthorized usage of signal.

3. Previous works

There are many interesting schemes that have been applied to improve the performance of block encryption in different applications [10,13]. A common concept is to remove the information pattern from the original message making each output bit appeared random. In this concept, each message would undergo a pre-process or masking process, which is reversible. Every bit in a masked message is equally important and is related to the original input message. If any one bit is missing or there is a bit error in the reversible masking process, one will not be able to recover the original message.

All-or-nothing transform uses a similar concept as the masking process to increase its tolerance to the brute force attacks. In this transform, a message undergoes a masking process in which all output is encrypted block by block making the recovering process difficult. To obtain a block of original message, one must decrypt all the ciphertext [11]. This concept increases the competency of block encryption; however, it also increases the processing time compared to direct encryption of the message.

Another design that has been proposed to reduce the encryption part is called Scramble All, Encrypt Small. This design is able to maintain security level of the encryption while reduces the length of encrypted message [12]. Even though Scramble All, Encrypt Small has a high security level, the number of output bits is higher than that in the original input due to the concatenated hash value. In some application, the increased bits become a burden to the system.

4. The proposed scheme

Our proposed CAS scheme for Pay-TV systems can be divided into two parts: the scrambling scheme for digital video streams and the key distribution scheme. Our scrambling scheme for digital video streams decreases the amount of encrypted data when compared to encryption

of the entire video streams. We intend reduce processing time and to open up possibilities of implementing highly secure encryption algorithm such as asymmetric encryption on digital video streams (an example of asymmetric encryption is the popular RSA algorithm [14]. Our key distribution scheme describes the way to distribute descrambling parameters to decoder box and smart card. Each decoder box in our scheme has its smart card pair, and it cannot receive the correct descrambling parameter from any other smart card. All descrambling parameters are hidden in the sent-out message (X) that is sent together with the related scrambled video information.

4.1. Scrambling scheme

Our scrambling scheme comprises of three major steps. It starts with separating the video bit streams to undergo masking process m bits at a time. Then all of the m bits undergo a masking process to produce masked message. The last step involves choosing certain bits in masked message for encryption.

4.1.1. Separating video bit streams. These m bits of the video bit streams are separated into fixed size blocks called formatted blocks. Each formatted block contains k bits and there are n formatted blocks labeled F_1, F_2, F_3 to F_n. The blocks must be in even number. These n blocks then undergo a masking process to become a masked message. This masking process is explained in the following section.

4.1.2. Masking process. Masking process constitutes two important functions. First, it has a strong collision resistant one-way hash function (h), which calculates k-bit size output from arbitrary size input (for example, in SHA-1, k is equal to 160 bits). Second, it has a high-speed pseudo-random number generator (G), which calculates k-bit size output from k-bit size input. We try to minimize the responsibility of the one-way hash function while making the most use to the pseudo-random number generator. This is done to minimize the computation time in the masking process and maintain the high security level at the same time. There are six steps in the masking process.

1. Use a pseudo-random number generator (G), initial vector (IV) and formatted block to create F'_1, F'_2, F'_3,..,F'_n. Then use this result, together with G, IV again, to define F''_1, F''_2, F''_3,.....,F''_n.

$$F'_1 = F_1 \oplus IV$$
$$F'_i = F_i \oplus G(F'_{i-1}) \quad \text{(Eq.1)}$$
$$F''_n = F'_n \oplus IV$$
$$F''_{i-1} = F'_{i-1} \oplus G(F''_i) \quad \text{(Eq.2)}$$
$$\text{where } i = 2,3,..n$$

2. Concatenate F''_1, F''_2,, ...F''_n together. Then separate them into two equal parts, which we call FR and FL.

3. Define FR'_i as the result of a pseudo-random number generator (G) and a hash value calculated from a strong collision resistant one-way hash function (h) and FR.

$$FR'_1 = G(h(FR))$$
$$FR'_i = G(FR'_{i-1} \oplus i\text{-}1) \quad \text{(Eq.3)}$$
$$\text{where } i = 2,3,..n/2$$

4. Set FR' as the concatenation of FR'_i. Then we denote and compute ML using FR' and FL.

$$FR' = FR'_1 \backslash\backslash FR'_2 \backslash\backslash .. \backslash\backslash FR'_{n/2} \quad \text{(Eq.4)}$$
$$ML = FR' \oplus FL \quad \text{(Eq.5)}$$

5. Define ML'_i as the result of a pseudo-random number generator (G) and a hash value calculated from a strong collision resistant one-way hash function (h) and ML.

$$ML'_1 = G(h(ML))$$
$$ML'_i = G(ML'_{i-1} \oplus i\text{-}1) \quad \text{(Eq.6)}$$
$$\text{where } i = 2,3,..n/2$$

6. Set ML' as the concatenation of ML'_i. Then we define and compute MR using FR and ML'.

$$ML' = ML'_1 \backslash\backslash ML'_2 \backslash\backslash .. \backslash\backslash ML'_{n/2} \quad \text{(Eq.7)}$$
$$MR = ML' \oplus FR \quad \text{(Eq.8)}$$

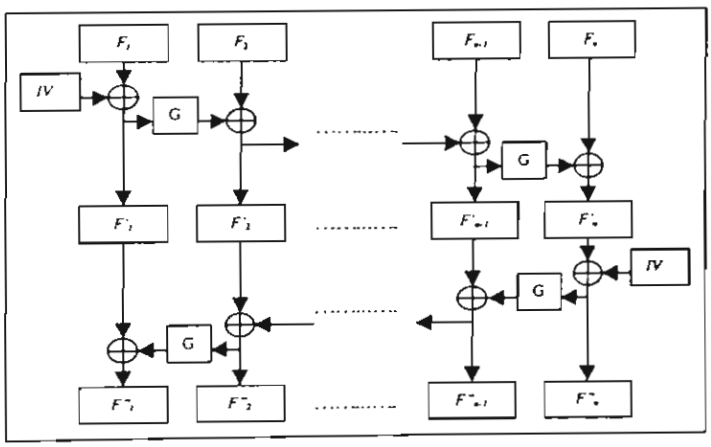

Figure 1. Step 1 of masking process

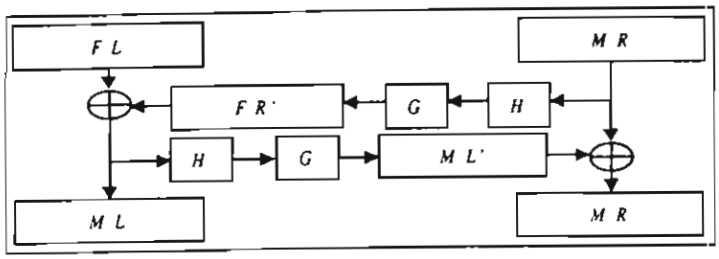

Figure 2. Steps 3 to 6 of masking process

4.1.3. Encryption scheme. The operation of our encryption scheme is described below.

1. Masked message ($ML\backslash\backslash MR$) is divided into blocks of equal size called masked blocks. There are n masked blocks (called M_1, M_2, M_3, ...M_n) with k bits in each.
2. Only the adjacent j bits from each masked block are selected. These bits are then put together to form a plaintext, which is n times j bits in size. The plaintext position of each selected j bit from each masked block is determined pseudo-randomly by means of a pseudo-random number generator operation in accordance with a pseudo-random seed (S).
3. The block of plaintext is encrypted with an encryption algorithm using a scrambling key (K_s) to become a ciphertext, which is also n times j bits in size.
4. Each bit of ciphertext is distributed back to its original position in the masked blocks. Then all blocks are concatenated to one another. The result is m-bit scrambled video information.

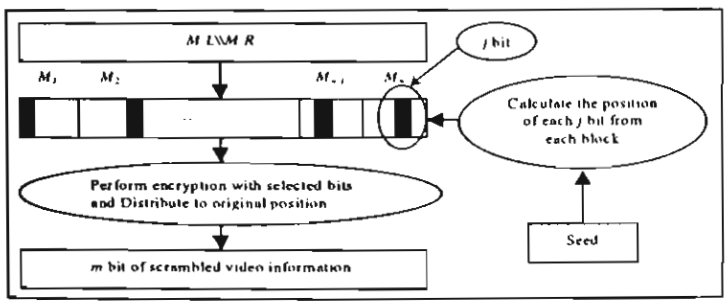

Figure 3. Encryption scheme

4.1.4. Descrambling process. This is the reverse of the scrambling process. Here scrambled video information is descrambled m bits at a time. The parameters used in descrambling are IV, S and K_s. The descrambling process is as follows.
1. The m-bit scrambled video information is divided into n blocks of equal size. Only j bits from each block are then selected using S and a pseudo-random number generator. Next, all chosen bits from every block are put together to become a ciphertext block.
2. The ciphertext block is decrypted using K_s to get a plaintext. Each bit in the plaintext is then distributed back to its original position in each block. The position was computed by S and a pseudo-random number generator. The result is masked blocks numbering M_1 to M_n.
3. The masked blocks numbering M_1 to M_n and IV go through a reversal of the masking process to become the m-bit original video information.

4.2. Key distribution scheme

In our scheme, the broadcaster sends a sent out message (X) along with the related scrambled video information. The smart card and the decoder box calculate S, IV and K_s that are used to descramble the related scrambled video information from the sent out message. With the pairing system, each smart card uses a different secret number to compute the data that is sent to its decoder box pair. Only the decoder box pair that has the same secret number will receive the correct S from the data. For this reason, McCormac Hack and smart card cloning attacks are ineffective in our scheme.

A technique based on Chinese Reminder Theorem (CRT) is used in constructing message X. Before presenting the construct method, we need to understand the mathematical background of the CRT [15]. Let p_1, p_2,..., p_t be positive integers that are pair wise relatively prime, and let R_1, R_2,..., R_t be positive integers, and let $N = p_1 * p_2 * ... * p_t$. Then the set of congruous equations

$$X = R_t \bmod p_t \quad (t = 1,2,3,....) \qquad \text{(Eq.17)}$$

have a common solution X which is ($1 \le X \le N-1$) and

$$X = (\Sigma^{t}_{i=1}(N/p_i) * R_i * f_i) \bmod N \qquad \text{(Eq.18)}$$
$$where\ 1 \equiv f_i * (N/p_i) \bmod p_i$$

The technique for constructing X is as follows. Let R be a random number. C_1 is a ciphertext of R and S, which is encrypted by a secret key (K_{card}) in the smart card. C_2 is a ciphertext of R and IV, which is encrypted by a secret key (K_{box}) in the decoder box. C_3 is a ciphertext of K_s, which is encrypted by secret key (K_{box}) Let p_1 ($>C_1$), p_2 ($> C_2$) and p_3 ($>C_3$) be relatively prime integers. Consider the following congruence equations:

$$X = C_1 \bmod p_1$$
$$X = C_2 \bmod p_2$$
$$X = C_3 \bmod p_3$$

In our scheme, all smart cards have an identical secret key K_{card} and a prime number p_{card}. Furthermore, each smart card keeps an individual secret number ($IN_{i\text{-}card}$) of each user i (U_i). For the decoder boxes, they all have an identical secret key K_{box} and two identical prime numbers p_{box1} and p_{box2}. In addition, each decoder box keeps an individual secret number ($IN_{i\text{-}box}$) of each user i (U_i). For the smart card and the decoder box that are paired, $IN_{i\text{-}card}$ of each user is the same as $IN_{i\text{-}box}$ of each user.

4.2.1. Signals transmission from the broadcaster.
1. The broadcaster generates a random number (R)
2. The broadcaster encrypts the concatenation of random number (R) and seed (S) with secret key K_{card} to obtain C_1. Then he encrypts the concatenation of random number(R) and initialization vector (IV) with secret key K_{box} to obtain C_2. he also encrypts the scrambling key (K_s) with secret key K_{box} to obtain C_3

$$C_1 = E_{Kcard}(R\backslash\backslash S) \qquad \text{(Eq.19)}$$
$$C_2 = E_{Kbox}(R\backslash\backslash V) \qquad \text{(Eq.20)}$$
$$C_3 = E_{Kbox}(K_s) \qquad \text{(Eq.21)}$$

3. Calculate sent out message (X) from C_1,C_2 , C_3 ,p_{card} ,p_{box1} and p_{box2} using CRT.
4. Send sent out message (X) and the related scrambled video information to all subscribers.

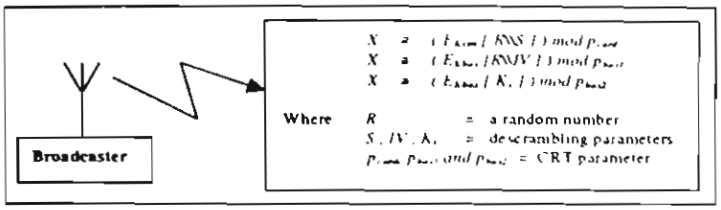

Figure 4. Steps involved in generating X

4.2.2. Operations in the decoder box and the smart card. The operation in smart card and decoder box after receiving the sent out message (X) and the related scrambled video information is as follows.
In the smart card
1. Compute R and S from the sent out message (X) using p_{card} and K_{card}.

$$R\backslash\backslash S = D_{Kcard}(X \bmod p_{card}) \qquad \text{(Eq.22)}$$

2. Define Y_i which is a the value computed from R, $IN_{i\text{-}card}$ and the pseudo-random number generator (G) of user i (U_i)

$$Y_i = S \oplus G(R \oplus IN_{i\text{-}card}) \qquad \text{(Eq.23)}$$

3. Send the computed Y_i to the decoder box.
In the decoder box
1. Calculate R, IV and K_s from the sent out message (X) by using K_{box}, p_{box1} and p_{box2}.

$$R\backslash\backslash V = D_{Kbox}(X \bmod p_{box1}) \qquad \text{(Eq.24)}$$
$$K_s = D_{Kbox}(X \bmod p_{box2}) \qquad \text{(Eq.25)}$$

2. Compute S by using R, $IN_{i\text{-}box}$ in the decoder box and Y_i from the smart card.

$$S = Y_i \oplus G(R \oplus IN_{i\text{-}box}) \qquad \text{(Eq.26)}$$

3. Use IV, K_s and S to descramble related scrambled video information

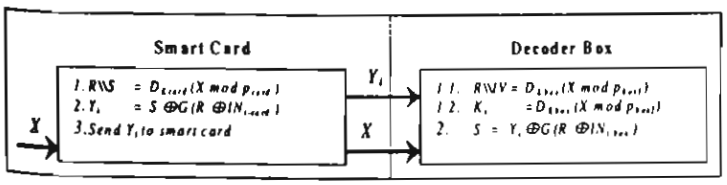

Figure 5. Operations between the decoder box and the smart card

5. Performance and security analysis

Both scrambling scheme and key distribution scheme are equally important to the system security and performance. This is because attackers can obtain the video information without the knowledge of the scrambling scheme if they can break the key distribution scheme to find the descrambling process parameter.

For block-by-block encryption, if t_e is the time used to encrypt a block, the total time used to encrypt n blocks is equal to nt_e. For the masking process of n-block message, t_h and t_r are the time used to compute a hash function and the high speed pseudo-random number generator once, respectively. The total time used in masking process in our scheme is $(t_h + 3t_r)n$. Comparing the time used in masking process and the time used to encrypt one single block, our scheme also uses less time than the total time used to encrypt the entire message in block by block fashion. The computation time for hash function and pseudo-random number generator are less than the computation time for encryption ($t_e >> t_h >> t_r$). For these reasons, the total time reduced by our scheme when compared to encrypting in the block-by-block fashion is equal to $(t_e - t_h - 3t_r)(n-1)$. The time reduction allows Pay-TV Operators to choose more complex encryption algorithm that needs long computation time. We can also reduce the overall performance time when our scrambling scheme is used on high-bit rate video streams.

The decision regarding number of blocks, n, is also important. The efficiency of our scheme, when compared to encrypting the entire original message, can improve at a faster pace as the number of n grow larger. However, there are drawbacks to this concept. The increase in number of n means more memory capacity is required to store the formatted blocks and masked blocks before processing. The effect error propagation would also increase when number of n is larger. The future trend of less expensive memory capacity and efficiency in information transfer that will decrease in error rate will likely make these drawbacks more tolerable, allowing a larger number of n to be used.

Our scheme is based on the Feistel Structure. Feistel gives a best form of permutation and makes no bit in the formatted blocks determinable unless every bit in the masked blocks is known. Also, as a result of step 1 of masking process, each bit in the masked block is a function of each and every bit in the formatted blocks. Both advantages enable us to choose to perform encryption to any part of the masked message while still holding same security level as encrypting the entire original message. The security of the scrambling scheme depends on the attackers' search for encrypted bit position and the chosen encryption algorithm.

Ciphers are usually regarded as acceptable and secure if they can withstand the known-plaintext and the chosen plaintext attacks [12]. It is difficult for these kinds of attacks to occur in our scheme because the initial vectors (IV) are different for every scrambling process. We will get a completely different masked message every time the

video stream is scrambled. In our scheme, even if the attackers know the original video streams, they must find the plaintext from the original video signal with probability of $1/2^k$ to break our scheme using know plaintext or chosen plaintext attacks. For this reason, our scheme is invertible with probability $1/2^{k(\delta+1)}$ for some constant δ (to an attacker who can see concrete given bound of plaintext-ciphertext pair)[12]. Our proposed scrambling scheme is therefore acceptable by this standard.

For the key distribution scheme, attacks may happen in two situations. First, attackers may intercept the broadcaster's message X. Second, attackers may intercept the data sent between smart card and decoder box. Assuming that attackers can intercept message X, it would still be difficult to know IV, S and K_s required to break the system. This is because if the attackers want to know the ciphertext of IV, S and K_s in message X, they must know p_{card}, p_{box1} and p_{box2} in order to extract the ciphertexts. Moreover, they must be able to break the chosen encryption algorithm to obtain IV, S and K_s while they do not have any information about plaintext-ciphertext relationships. It can be clearly seen that the security of the proposed system depends on the encryption algorithm used, and the ability to break the CRT without knowing p_{card}, p_{box1} and p_{box2}.

Assuming that attackers can intercept Y_i, which is the data sent between smart card and decoder box, it would still be difficult to use Y_i with another decoder box or to use it to break the system. This is because Y_i values calculated from different smart card are different, and only the decoder box-pair can calculated the correct S from Y_i it received. If attackers want to know S that is hidden along with Y_i, they must know R and $IN_{i\text{-}card}$ in the smart card. It is very hard for attackers to know R and $IN_{i\text{-}card}$ since they must be able to first break the chosen encryption algorithm used to encrypt R and also be able to break CRT to get ciphertext of R without any knowledge of p_{card}. Besides, to obtain information of $IN_{i\text{-}card}$ in the smart card, attackers must be able to break the chosen pseudo-random number generator. From the two situations here, attackers must, at least, be able to break CRT and the chosen encryption algorithm in order to break the key distribution scheme. This makes the security of our scheme dependent on the chosen encryption algorithm.

6. Conclusion

In this paper, we have proposed a scrambling scheme and a key distribution scheme. Our scrambling scheme, used for a high bit rate information, has succeeded in reducing the encryption time. The schemes is also beneficial to digital Pay-TV providers who can make use of the highly secure but more complicated and time consuming asymmetric encryption algorithms in their systems. Moreover, our scrambling scheme can function like a block encryption for other types of digital information. Our key distribution scheme can be used to distribute scrambling keys for other types of scrambling scheme. It provides high security and can prevent McCormac Hack and smart card cloning.

7. Acknowledgment

The authors thank the Thailand Research Fund for partly supporting this research work (PDF/27/2543).

8. Reference

[1] B.M. Macq and J.J. Quisquater, 'Cryptology for Digital TV Broadcasting', Proceedings of the IEEE, Vol. 83, No. 6, June, 1995, pp. 944-957.

[2] F.K. Tu, C.S. Laih, and H.H. Tung, 'On Key Distribution Management for Conditional Access System on Pay-TV System', IEEE Transactions on Consumer Electronics, Vol.45, February, 1999, pp. 151-158.

[3] F. Coutrot and V. Michon, 'A Single Conditional Access System for Satellite-Cable and Terrestrial TV', IEEE Transactions on Consumer Electronics, 1989, pp. 464-468.

[4] S. R. Ely and S. R. Shuttleworth, 'Conditional Access Scrambling Techniques for Terrestrial UHF Television Broadcasts', IBC, 1988, pp. 318-322.

[5] ANSI X.3.92, 'American National Standard for data Encryption Algorithm (DEA)', American National Standard Institute, 1981.

[6] W. Mooij, 'Advances in Conditional Access Technology', IBC, No. 447, September, 1997, pp. 461-464.

[7] J. S. Saini, 'The BBC Select decoder', IBC, 1992, pp. 410-413.

[8] A.G. Mason, 'Conditional Access for Broadcasting', IBC, 1988, pp. 328-333.

[9] P. Peyret, G. Lisimaque and T.Y. Chua, 'Smart cards provide very high security and flexibility in subscribers management', IEEE Transactions on Consumer Electronics, Vol. 36, 1990, pp. 744-752.

[10] M. Bellare and P. Rogaway, ' Optimal Asymmetric Encryption ', EUROCRYPT 94, No. 950, Springer- Verlag, 1994, pp. 92-111.

[11] R. Revest, 'All-Or-Nothing Encryption and the Package Transform', the 4th International Workshop on Fast Sortware Encryption, Vol. 1267 of Lecture Notes in Computer Science, Springer- Verlag, 1997, pp. 210-218.

[12] J. Markus, P.S. Julien and Y. Moti, 'Scramble All, Encrypt Small', Fast Software Encryption, 1999

[13] D. Johnson and S. Matyas, 'Asymmetric Encryption: Evolution and Enhancements', CryptoBytes, Vol 2, No.1, Spring, 1996.

[14] R. L. Rivest, A. Shamir and L. Adleman, 'A Method for Obtaining Digital Signatures and Public Key Cryptosystems', Communications of the ACM, Vol. 21, No. 2, February, 1978, pp. 120-126.

[15] G.H. Chiou and W. T. Chen, 'Secure Broadcasting Using the Secure Lock', IEEE Transaction on Software Engineer, Vol.15, 1989, pp. 929-934.

Applying Spread Spectrum Technique for Transmitting Extra Bits over AWGN Channel

Thumrongrat AMORNRAKSA
Multimedia Communications Laboratory, Department of Computer Engineering,
King Mongkut's University of Technology Thonburi, Bangkok 10140, Thailand
e-mail: t.amornraksa@cpe.eng.kmutt.ac.th

Peter SWEENEY
Centre for Communication Systems Research, School of Electronics,
Computing and Mathematics, University of Surrey, Guildford GU2 7XH, UK
e-mail: p.sweeney@eim.surrey.ac.uk

Abstract

This paper describes an encoding scheme based on direct sequence spread spectrum technique for conveying some extra bits in a communication channel. In the encoding process, small amount of information is added into the original transmitted signal to generate the output signal for transmission, and the user at the receiver end is able to obtain both contents. In the decoding process, the added extra bits are first extracted from the received signal, and then used to recover the original signal. The scheme was examined by simulation method and its performance was measured. Error control codes were applied to the extra bits before the encoding process so that the scheme's performance can be significantly improved. The scheme was also examined by transmitting the data through an AWGN channel to observe its performance when implemented in general applications. With the proposed scheme, the existing allocated bandwidth in the broadcast channel can be utilized in a more efficient way.

1. Introduction

In spread spectrum (SS) communications [1, 2], a low-level wideband signal can easily be hidden within the same spectrum as a high power signal where each signal appears as noise to the other. The heart of these SS systems is a pseudo-random binary sequence (PRBS). For these direct sequence SS systems, the original baseband bit stream is multiplied by the PRBS to produce a new bit stream. At the receiver, the low level wideband signal will be accompanied by noise, and by using a suitable detector/demodulator with the correct PRBS, this signal can be squeezed back into the original narrow baseband. Because noise is completely random and uncorrelated, the wanted signal can easily be extracted [3]. In other words, only those receivers equipped with the correct PRBS can decode the original bit stream.

Based on these concepts, we may construct an encoding scheme for conveying some extra information bits via a transmission channel without requiring extra bandwidth. That is, some extra bits will be added into the original signal before the transmission process takes place. These extra bits can give benefits to various applications in many ways. For instance, in multimedia applications, they may be used to enhance the quality of the transmitted image or transport a control signal. However, at the receiver, both the extra bits and the original signal are required for correct recovery.

In this paper we describe a possible approach to achieve the above requirement. A method based on the direct sequence SS technique is proposed and then used to construct an encoding scheme, which enables transmission of extra bits over the existing allocated bandwidth. In Section 2, the method of constructing the encoding scheme is explained. The details of the encoding scheme are described, including the theory behind its operations. Section 3 describes all possible adding methods that can be used in the scheme, and the simulation model used to evaluate the performance of the scheme. The results from simulations and discussions are then given in Section 4. Finally, Section 5 provides some concluding remarks and directions for future work.

2. Description of the scheme

In digital communication systems, channel coding is normally applied to the signal before transmission takes place, and this signal is considered as the original signal in our encoding scheme. Using the SS technique as described in [4], the extra bits will be added via an add operation to the original signal after the channel coding process to obtain the resulting signal for transmission. Given a key to reproduce the same PRBS at the receiver's end, the extra bits can be recovered. The original signal can then be recovered by subtracting the extra bits from the received signal. Any errors which occur at this stage such as communication channel errors will be detected and corrected by the channel decoder. The operation of the encoding scheme is shown in the figure below.

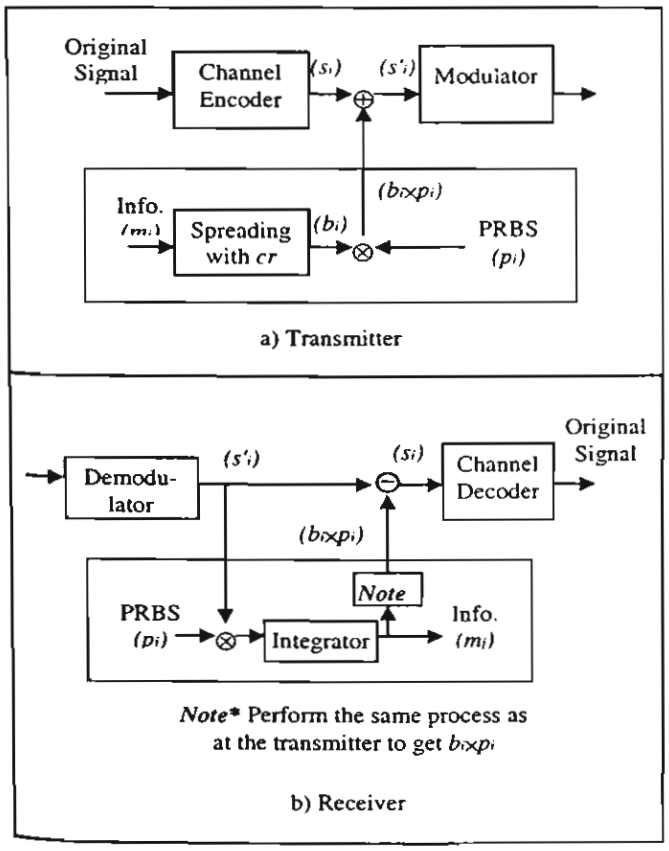

Figure 1. The operation of the encoding scheme

The basic steps of adding the extra bits to the original signal are now described. We denote the sequence of extra bits we want to add to the original signal by m_j, $m_j \in \{-1, 1\}$. This discrete signal is spread by a large factor cr, the chip-rate, to obtain the spread sequence (b_i), $b_i = m_j$, $j \bullet cr \le i < (j+1) \bullet cr$. The spread sequence is then modulated with a PRBS (p_i), $p_i \in \{-1, 1\}$ and added to the original signal s_i, where each s_i block containing x bits, to yield the transmitting signal (s'_i),

$$s'_i = s_i + p_i \bullet b_i. \quad (1)$$

At the receiver, the recovery of the added bits is easily accomplished by multiplying the received signal with the same PRBS (p_i) that was used in the encoder. The summation over the correlation window, i.e. over cr, is as follows:

$$r_j = \sum_{i=j\bullet cr}^{(j+1)\bullet cr-1} p_i \bullet s'_i = \sum_{i=j\bullet cr}^{(j+1)\bullet cr-1} p_i \bullet s_i + \sum_{i=j\bullet cr}^{(j+1)\bullet cr-1} p_i^2 \bullet b_i \quad (2)$$

The first term on the right-hand side of eq. (2) vanishes if p_i and s_i are uncorrelated, and then $\sum_{i=j\bullet cr} p_i = 0$ [5]. Since $p_i^2 = 1$, r_j ideally becomes

$$r'_j = cr \bullet m_j \quad (3)$$

and the recovered extra bit $m'_j = sign\ (r'_j)$.

As an example, let the bit-rate of the original signal be 10 Mb/s, the chip-rate $cr = 500$ and let the block size x be 4 bits. Then, the rate at which extra bits can be added after the channel coding process is 5 kb/s. With this bit-rate, the extra bits could be an image signal, for instance, in a compressed form transmitted every 30s or so. Hence, we can transmit the total bit-rate of 3.005 Mb/s within the existing bandwidth allocation of 10 Mb/s.

To increase the bit-rate of the extra bits, the chip-rate and the block size should be reduced. However, a smaller block size implies a greater likelihood that subtracting the extra bits from the received signal will not give the original signal. In addition, a smaller chip-rate implies a greater likelihood of error in decoding the extra bits. To reduce this latter likelihood of error, an error control code can be applied to the extra bits before the spreading process.

3. Simulation model

Simulations were carried out using C programming language. The block size x was varied from 2-7 bits to represent up to 128 values. The chip-rate was varied from 0 to a value that gives no error in the extracted information. According to eq. (1), the addition between s_i and $p_i \bullet b_i$ can be performed in five different methods, yielding five operations, as follows:

i) $s'_i = s_i + p_i \bullet b_i$
ii) $s'_i = s_i$, if $s_i = 0$ and $p_i \bullet b_i = -1$,
or $s_i = (2^x - 1)$ and $p_i \bullet b_i = 1$,
Otherwise $s'_i = s_i + p_i \bullet b_i$

iii) $s'_i = s_i$, if $s_i = 0$ and $p_i \bullet b_i = -1$,
otherwise $s'_i = (s_i + p_i \bullet b_i) \bmod 2^x$

iv) $s'_i = s_i$, if $s_i = (2^x - 1)$ and $p_i \bullet b_i = 1$,
otherwise $s'_i = (s_i + p_i \bullet b_i) \bmod 2^x$

v) $s'_i = (s_i + p_i \bullet b_i) \bmod 2^x$

Table 1 shows five possibilities of s'_i resulting from different adding methods (*i-v*), which can be used in the encoding scheme. Since each method gives different levels of performance in the decoding processes, they will be investigated to determine a suitable one to be used in practice.

Table 1: Possible values result from the different adding methods in eq. (1) for block size $x = 2$

	$s_i + p_i \bullet b_i$							
s_i	0	0	1	1	2	2	3	3
$p_i \bullet b_i$	-1	1	-1	1	-1	1	-1	1
i.) s'_i	-1	1	0	2	1	3	2	4
ii.) s'_i	0	1	0	2	1	3	2	3
iii.) s'_i	0	1	0	2	1	3	2	0
iv.) s'_i	3	1	0	2	1	3	2	3
v.) s'_i	3	1	0	2	1	3	2	0

As the table 1 indicates, the method *i* produces some results that are out of the range of the values that the original bit stream can represent, e.g. the value of 4 cannot be represented by 2-bit number, and thus this method will not be used in the simulation since it will increase the bandwidth of the transmitted signal. For the remaining methods, the different values of s'_i exist when performing the addition between $s_i = 0$ and $p_i \bullet b_i = -1$, or $s_i = (2^x - 1)$ and $p_i \bullet b_i = 1$. In the first part of the experiments, the methods *ii-v* were used in the simulations, with the aim of demonstrating how an encoding scheme may be constructed as well as how well it performs. The differences when applying each method were then analyzed, based on the simulation results obtained.

Then the error control codes were applied to the extra bits. This will of course reduce the main throughput by a factor k/n, where k and n are the number of input symbols and output symbols respectively, which one may think that this may be difficult to compensate by a smaller value of chip-rate in the decoding process. To demonstrate that the error control codes can improve the performance of the encoding scheme, various codes are applied to the extra bits before performing the spreading process, and theirs performances are then compared to the one without the codes. For example, Reed Solomon codes, Binary BCH code, Golay code and Convolutional code with rate 1/2 and $K = 7$. Consult [6, 7], for those who are not familiar with the subject

At this step of our simulations, however, the encoding scheme will be performed in an error free communication channel. That is, the errors that occurred in the original signal came solely from the need to remain within the bandwidth of the transmission channel. The objective for doing this is to focus on only the errors that occur in the extracted extra bits, which are mainly related to the performance of the scheme. In addition, the MPEG encoded stream was used to carry the extra bits while generating the data to be transmitted through the channel can be accomplished by using a random number generator. One that produces a uniform distribution of numbers on the interval 0 to a maximum value is provided by a function *rand( )* in C language. Using this function, we can say that any value less than half of the maximum value is a zero; any value greater than or equal to half of the maximum value is a one, and then input into the constructed encoding scheme as described in the figure 1

After the proper code that gives the best performance is found, the proposed scheme will be simulated in a communication channel. At this step, an Additive White Gaussian Noise (AWGN) channel is chosen since it is a type of noise that most communication systems encounter [8]. An error control code i.e convolutional code with rate 1/2 coding is also applied to the original signal in order to observe the performance of the proposed scheme when implemented in practice. The simulation model used in the experiments is shown in the figure below.

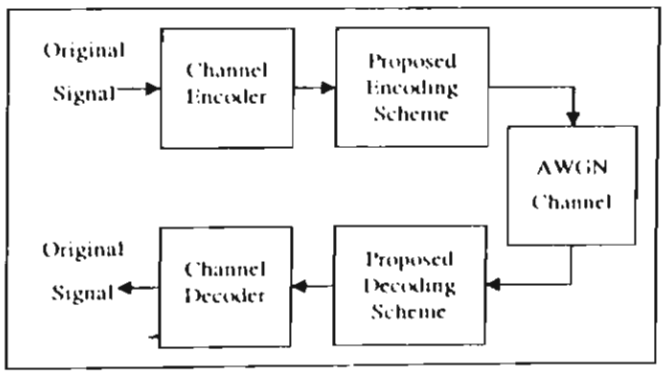

Figure 2. Simulation model in an AWGN channel

Adding noise to the transmitted channel symbols involves generating Gaussian random numbers, scaling the numbers according to the desired energy per symbol to noise density ratio, E_s/N_0, and adding the scaled Gaussian random numbers to the channel symbol values. For the uncoded channel, $E_s/N_0 = E_b/N_0$, since there is one channel symbol per bit. However, for the coded channel, $E_s/N_0 = E_b/N_0 + 10\log_{10}(k/n)$. For example, for rate 1/2 coding, $E_s/N_0 = E_b/N_0 + 10\log_{10}(1/2) = E_b/N_0 - 3.01$ dB.

Since C language only provides a uniform random number generator, *rand()*, in order to obtain Gaussian random numbers, we take advantage of relationships between uniform, Rayleigh, and Gaussian distributions:

Given a uniform random variable U, a Rayleigh random variable R can be obtained by

$$R = \sqrt{2\sigma^2 \ln(1/(1-U))} = \sigma\sqrt{2 \ln(1/(1-U))} \quad (4)$$

where σ^2 is the variance of the Rayleigh random variable, and given R and a second uniform random variable V, two Gaussian random variables G and H can be obtained by

$$G = R \cos U, \text{ and } H = R \sin V \quad (5)$$

In the AWGN channel, the signal is corrupted by additive noise, $n(t)$, which has the power spectrum $N_0/2$ watts/Hz. The variance σ^2 of this noise is equal to $N_0/2$. If we set the energy per symbol E_S equal to 1, then $E_S/N_0 = 1/2\sigma^2$. So $\sigma = \sqrt{1/(2(E_S/N_0))}$ [9].

4. Simulation results and discussions

From the simulation results, the smallest chip-rate with no errors after the extraction process using different adding methods (*ii-v*) are shown in the table 2.

Table 2: Values of the chip-rate with no errors after the extracting process, at different block sizes

Chip-rate *cr*	Block Size *x*					
	2	3	4	5	6	7
Method *ii*	46	110	455	1100	4150	12000
Method *iii*	190	400	1450	5100	15200	45000
Method *iv*	210	410	1400	4500	16000	43500
Method *v*	α	α	α	α	α	α

It is clear from the table 2 that the adding method *ii* gave the best performance, i.e., needs the smallest value of the chip-rate, especially in the larger block sizes, compared to other methods. Therefore, from this point, the adding methods *ii* was chosen for the simulations to measure the performance of the encoding scheme at various block sizes. Furthermore, for these block sizes, other values of the chip-rate considered resulted in different values of BER in the extracted extra bits, and these values and the underlying line are shown in Figure 3 for the adding methods *ii*.

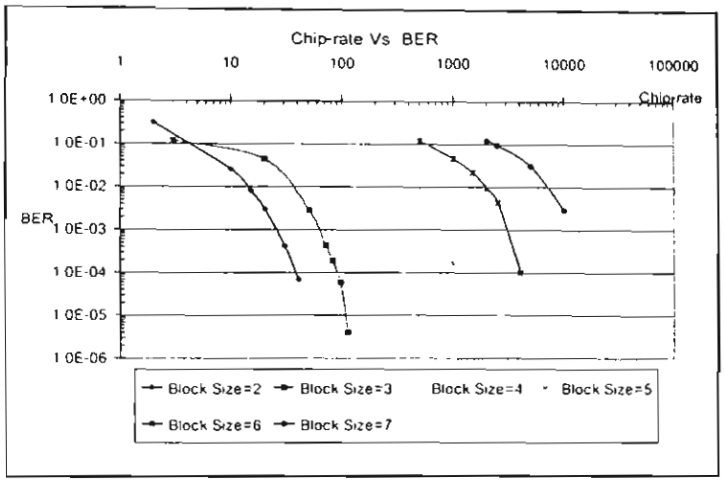

Figure 3. Bit error rate of extracted extra bits at different block sizes using adding method *ii*

First of all, note that the method *v*, according to the table 1, provides reversible decoding, i.e. the original bit stream can be correctly recovered. By using the knowledge of $p_i \bullet b_i$, every value of s'_i can be referred back to s_i, in the same way as one-to-one mapping, while the remaining methods cause some errors in the decoding process. However, the results from the table 2 showed that no matter how large the chip-rate is, when method *v* is used, the added bits will not be correctly extracted. The reason is because of inaccurate results from the summation over the correlation window, shown in eq. (2), in the decoding process; that is, the decoder will give a wrong sign of r'_j, and translate to a wrong value of m_j. This event can be noticed by considering the original bit stream as a random sequence and observing whether its distribution is flat or not. If so, it is likely that the summation results from eq. 2 will lead to a wrong value of m_j.

A good example that indicates this notification is shown in the table 1, where the original bit stream is equally distributed; i.e., each value (sample) has the same probability of occurrence. It can now be seen that the summation term of all possible values of $s'_i \times p_i \bullet b_i$ in each adding method is 8, 6, 3, 3 and 0, respectively. For example, in method *ii*, the summation term can be calculated as follows: (0×-1) + (1×1) + (0×-1) + (2×1) + (1×-1) + (3×1) + (2×-1) + (3×1) = 6. It is obvious that the larger the value of the summation term, the smaller the chip-rate needed to correctly extract the extra bits. This analytical observation can be proven by the simulation results from the table 2. In contrary, a smaller value of the summation term results in more incorrectly recovered bit stream at the same bit rate. The explanation is given

elow. However, method *v* should not be practically used n the encoding scheme.

As mentioned earlier, although the added bits are correctly obtained when a proper adding method is used, he recovered original bit stream, after subtracting the extra bits from the received data, still contains some errors. The reason for this is implicitly shown in the table 1. That is, for example in the adding method *ii*, when $s'_i = 0$ and $p_i \bullet b_i = -1$, the decoder will not be able to determine whether s_i is 0 or 1, and this gives the possibility of making a wrong decision up to 50%. If the block size x is used, the errors occurring in the recovered original signal will be approximately $1/(2^x)$ %. However we can reduce this error rate by using different adding methods e.g. method *iii*. According to the table 1, the remaining errors will be approximately $1/(2^{x+1})$ %. Nevertheless, when the adding method *iii* is used, the chip-rate needs to be increased in order to prevent any error in the process of extracting extra bits. The simulation results in the table 2 already verified this fact. The same explanation can also be applied to the adding method iv.

From the figure 3, it can be seen that a larger block size needs a bigger chip-rate to retain the same BER. In addition, since one single bit error in the extracted information causes error propagation in the original signal, any value other than a large chip-rate will result in a large BER. To further improve performance of the scheme, the error control codes were applied to the extra bits before the spreading process. This reduced the amount of data rate to be added in the original signal by a factor k/n. However, the amount of chip-rate required in the decoding process was decreased, and this resulted in an improvement in efficiency for the entire system. The smaller values of the chip-rate, when applied the error control codes, that gave different values of BER in the extracted extra bits are shown in the figure 4 for the block sizes of 4.

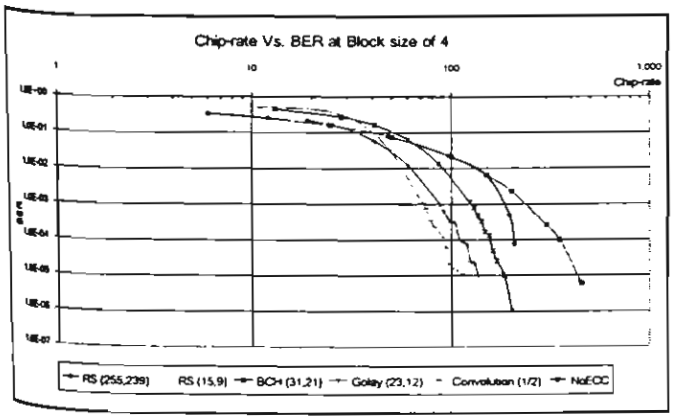

Figure 4. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 4

It obviously shows, from the figure 4, that the error control codes provide the smaller values of the chip-rate. It can also be seen the convolutional code gave the best performance, compared to the others, and hence was used in the next experiments. To illustrate the benefits of using error control codes, the number of original data (symbol) which is used to convey the extra bits is plotted against the BER. Figure 5 shows the performance comparison of the encoding scheme with and without the error control codes at the block size of 4.

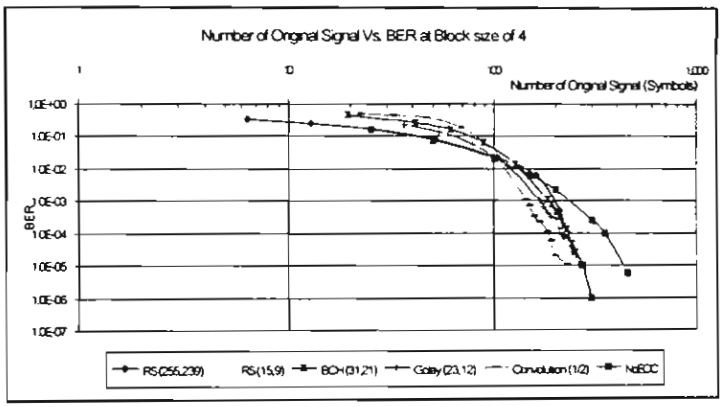

Figure 5. Performance comparison between the scheme with and without the error control codes

Table 3 shows an example of the total amount of bandwidth saved by the use of the RS (15, 9) code at different block sizes. The results are presented in the form of efficiency improvement (%) of the scheme with the RS (15, 9) code, compared to the one without the code.

Table 3. Summary of efficiency improvement of the scheme with the (15, 9) RS code at different block size

Block Size x	2	3	4	5	6	7
Efficiency (%)	33.3	33.3	39.1	38.9	46.1	44.2

It is clear that the larger the block size, the higher the efficiency of the scheme. For that reason, the error control codes can be very useful when the scheme is operated with a large block size. To observe the performance of the scheme when implemented in the AWGN channel, the simulations were conducted according to the model in the figure 2. In the following figure 6, a plot of the BER versus the E_b/N_0 for the scheme using the convolutional code with rate 1/2 is given.

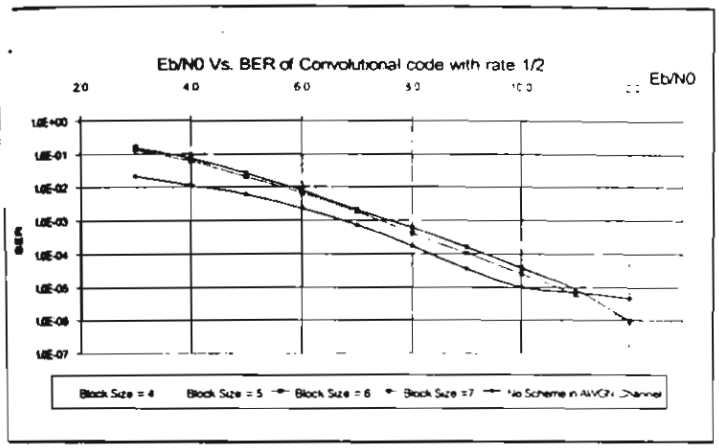

Figure 6. Performance comparison when the scheme is implemented in the AWGN channel at various block sizes

From the figure, it can be seen that errors occurred at the output of the original signal were larger than the ones from pure AWGN channel. This is because of the unrecoverable errors remained in the data after decoding process. However, when the value of E_b/N_0 was increased the BER became smaller, especially with the larger block size. It can be noticed that the scheme operated with a larger block size gave better performance. Nevertheless, the value of the chip-rate required for large block size is enormous and this choice should be carefully considered. From the obtained result so far, the proposed scheme is not fit well when implemented in the AWGN channel.

5. Conclusions and future work

In this paper we have shown a method of constructing an encoding scheme for transmitting extra bits via an existing transmitted signal, based on the direct sequence spread spectrum technique. We have also shown that different adding methods used in the encoding scheme not only gave different values of chip-rate that enables the extra bits to be correctly extracted, but also gave different performances of the scheme. Furthermore, we have shown experimentally and analytically that the scheme's performance was improved by applying the error control codes to the extra bits before the encoding process. Our approach has showed an idea of how to utilize the existing allocated bandwidth in a more efficient way. Further work can be carried out by simulating the scheme in some specific applications such as Digital Video Broadcasting (DVB), where powerful error control scheme is used, and observe its performance and reliability.

6. Acknowledgment

The authors would like to thank the Thailand Research Fund (TRF) for financial support throughout this work (Funding Code: PDF/27/2543).

7. References

[1] R. Pickholtz, D. Schilling and L. Millstein, "Theory of Spread Spectrum Communications A Tutorial", *IEEE Transaction on Communication*, Vol. COMM-30, 1982, pp 855-884.

[2] W. C. Y. Lee, "Spectrum Efficiency in Cellular", *IEEE Transactions on Vehicular Technology*, Vol. 38, No 2, May 1989, pp. 69-75.

[3] R. C. Dixon, "*Spread Spectrum Systems with Commercial Applications 3rd Edition*", John Wiley & Son Inc., New York, 1994.

[4] F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video", *Signal Processing*, Vol 66, no 3 (Special issue on Watermarking), May 1998, pp. 283-301.

[5] H. Stark and J. W. Woods, "*Probability, Random Variables and Estimation Theory for Engineers*", Prentice Hall, Englewood Cliffs, N.J. 1986.

[6] P. Sweeney, "*Error Control Coding: An Introduction*", Prentice Hall, London, 1991.

[7] S. Lin and D. J. Jr. Costello, "*Error Control Coding: Fundamentals and Applications*", Prentice-Hall, Englewood Cliffs, NJ, 1983.

[8] M. Michelson and A. H. Levesque, "*Error Control Techniques for Digital Communication*", New York: John Wiley & Sons, 1985.

[9] K. J. Larsen, "*Short Convolutional Codes with Maximal Free Distance for Rates 1/2, 1/3, and 1/4*", IEEE Transactions on Information Theory, vol. IT-19, May, 1973, pp. 371-372

TRANSMITTING EXTRA BITS OVER DVB SYSTEMS

T. Amornraksa and P. Sweeney

Multimedia Communications Laboratory,
Department of Computer Engineering,
King Mongkut's University of Technology
Thonburi, Bangkok 10140, Thailand.
Phone:+66-2-4709083, Fax:+ 66-2 -872-5050
Email: t.amornraksa@cpe.eng.kmutt.ac.th

Centre for Communication Systems Research,
School of Electronics, Computing and
Mathematics, University of Surrey,
Guildford GU2 7XH, UK
Phone:+44-1483-879123, Fax:+ 44-1483-876011
Email: p.sweeney@eim.surrey.ac.uk

ABSTRACT

This paper describes a concept of dual level access scheme for conveying some extra bits in the broadcasting networks. By adding small amount of information, using spread spectrum techniques, into the encrypted signal to generate the output signal for transmission, any user at the receiver end is allowed to view this content, while only the users with the decryption key can view the encrypted content. In the decoding process, the added extra bits are first extracted from the received signal, and then used to recover the encrypted signal. In this paper, the scheme's efficiency was improved by applying error control codes to the extra bits before the encoding process. Moreover, the scheme was implemented in DVB applications, by a simulation method, by transmitting the encoded MPEG-coded stream through an AWGN channel, to observe its performance. With the improved scheme, the existing allocated bandwidth in the broadcast channel is utilized in a more efficient way.

1. INTRODUCTION

A dual level access scheme is an encoding scheme which gives two levels of access to the users in the network. For example, in pay-TV services, some digital TV programmes will be encrypted before transmitting to every subscriber. Only the authorized subscribers who pay an extra fee can get access to those programmes. This technique does not give any value at all to other subscribers who have not paid for that particular programme. The allocated bandwidth is only used for broadcasting the encrypted signal to the authorized subscribers, which may be a small group compared to all subscribers in the network. With the dual level access scheme proposed in [1], the authorized subscribers can access the encrypted signal and, at the same time, the other subscribers can receive something on the same channel, such as an advertisement, so that more benefit is given to both the service providers and all subscribers in the network, and of course, the scheme shall not extend the existing allocated bandwidth.

In this paper, such scheme was developed by applying error control codes to the extra information bits before being added to the encrypted signal in order to improve the performance of the scheme. Various error control codes were tested by simulation method to determine the best suited one to be used with the scheme. Moreover, The improved scheme was tested for the practical use purpose by transmitting the extra bits on Digital Video Broadcasting (DVB) systems, where a powerful error control scheme is applied, through an Additive White Gaussian Noise (AWGN) channel to observe its performance. In the next section, the method of constructing the encoding scheme is explained. The details of the encoding scheme are described, including the theory behind its operations. Section 3 describes the simulation model used to evaluate the performance of the scheme. The results from simulations and discussions are then given in the Section 4, and finally, Section 5 provides some concluding remarks.

2. DESCRIPTION OF THE SCHEME

The principle of the scheme is based on direct sequence spread spectrum (SS) technique [2] which is used to add specific information i.e. advertisements, referred to as extra bits, to the access-limited signal, which is protected by encryption techniques. The heart of the SS systems is a pseudo-random binary sequence (PRBS). For these direct sequence SS systems, the original baseband bit stream is multiplied by the PRBS to produce a new bit stream. Only those receivers equipped with the correct PRBS can decode the original message. At the receiver, the low level wideband signal will be accompanied by noise, and by using a suitable detector/demodulator with the correct PRBS, this signal can be squeezed back into the original narrow baseband. Because noise is completely random and uncorrelated, the wanted signal can easily be extracted. Several watermarking techniques are also based on these ideas such as those proposed in [3, 4].

Using a similar technique to the above, the extra bits will be added to the encrypted signal (after the channel coding process) to give the signal for transmission. Given a key to reproduce the same PRBS at the receiver's side, the extra bits can be recovered. Then the encrypted signal can be

recovered by subtracting the extra bits from the transmitted signal [1]. Any errors that occur at this stage will be detected and corrected by the channel decoder. The operation of the encoding scheme is shown in Figure 1 below.

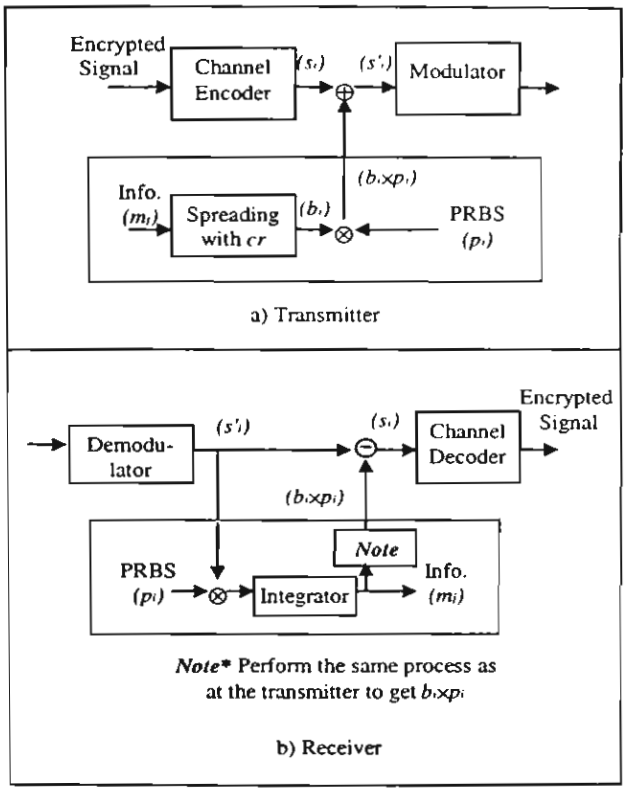

Figure 1. The operation of the encoding scheme

The basic steps of adding the extra bits to the original signal are now described. We denote the sequence of extra bits we want to add to the original signal by m_j, $m_j \in \{-1, 1\}$. This discrete signal is spread by a large factor cr, the chip-rate, to obtain the spread sequence (b_i), $b_i = m_j$, $j \bullet cr \le i < (j+1) \bullet cr$. The spread sequence is then modulated with a PRBS (p_i), $p_i \in \{-1, 1\}$ and added to the original signal s_i, where each s_i block containing k bits, to yield the transmitting signal (s'_i),

$$s'_i = s_i + p_i \bullet b_i \qquad (1)$$

At the receiver, the recovery of the added bits is easily accomplished by multiplying the received signal with the same PRBS (p_i) that was used in the encoder. The summation over the correlation window, i.e. over cr, is as follows:

$$\begin{aligned} r_j &= \sum_{i=j \bullet cr}^{(j+1) \bullet cr-1} p_i \bullet s'_i \\ &= \sum_{i=j \bullet cr}^{(j+1) \bullet cr-1} p_i \bullet s_i + \sum_{i=j \bullet cr}^{(j+1) \bullet cr-1} p_i^2 \bullet b_i \end{aligned} \qquad (2)$$

The first term on the right-hand side of (2) vanishes if p_i and s_i are uncorrelated, and then $\sum_{i=j \bullet cr}^{(j+1) \bullet cr-1} p_i = 0$ [5]. Since $p_i^2 = 1$, r_j ideally becomes

$$r'_j = cr \bullet m_j \qquad (3)$$

and the recovered extra bit $m'_j = sign\ (r'_j)$.

As an example, let the bit-rate of the encrypted signal be 10 Mb/s, the chip-rate $cr = 500$ and let the block size k be 4 bits. Then, the rate at which extra bits can be added after the channel coding process is 5 kb/s. With this bit-rate, the extra bits could be an image signal, for instance, in a compressed form transmitted every 30s or so. Hence, we can transmit the total bit-rate of 3.005 Mb/s within the existing bandwidth allocation of 10 Mb/s.

To increase the bit-rate of the extra bits, the chip-rate and the block size should be reduced. However, a smaller block size implies a greater likelihood that subtracting the extra bits from the received signal will not give the encrypted signal. In addition, a smaller chip-rate implies a greater likelihood of error in decoding the extra bits. To reduce this latter likelihood of error, an error control code can be applied to the information bits before the spreading process.

3. SIMULATION MODEL

Simulations were carried out using C programming language. The block size k was varied from 2-7 bits to represent up to 128 values. The chip-rate was varied from 0 to a value that gives no error in the extracted information. However, it is obvious that some results from the addition of s_i and $p_i \bullet b_i$ are out of range of the values that the encrypted signal can represent, and thus more bandwidth will be required for transmitting the output signal. In order to keep the output bit-rate constant, the addition of s_i and $p_i \bullet b_i$ is performed as follows;

$$\begin{aligned} s'_i = s_i, &\text{ if } s_i = 0 \text{ and } p_i \bullet b_i = -1, \\ &\text{or } s_i = (2^k - 1) \text{ and } p_i \bullet b_i = 1 \\ &\text{Otherwise } s'_i = s_i + p_i \bullet b_i \end{aligned} \qquad (4)$$

When the error control codes are applied to the extra bits, it will of course reduce the main throughput by a factor k/n, which one may think that this may be difficult to compensate by a smaller value of chip-rate in the decoding process. To demonstrate that the error control codes can improve the performance of the encoding scheme, various codes are applied to the extra bits before performing the spreading process, and theirs performances are then compared to the one without the codes. For example, Reed Solomon codes, Binary BCH code, Golay code and Convolutional code with rate 1/2 and $K = 7$.

However, at this state of our simulations, the encoding scheme will be performed in an error-free

communication channel. That is, the errors that occurred in the encrypted signal came solely from the need to remain within the bandwidth of the transmission channel. The aim of doing this is to focus on only the errors that occur in the extracted information bits, which are mainly related to the performance of the scheme.

After the proper code that gives the best performance is found, the proposed scheme will be simulated in a communication channel. At this step, an AWGN channel is chosen since it is a type of noise that most communication systems encounter [6]. The simulation model used in the experiments is shown in figure below.

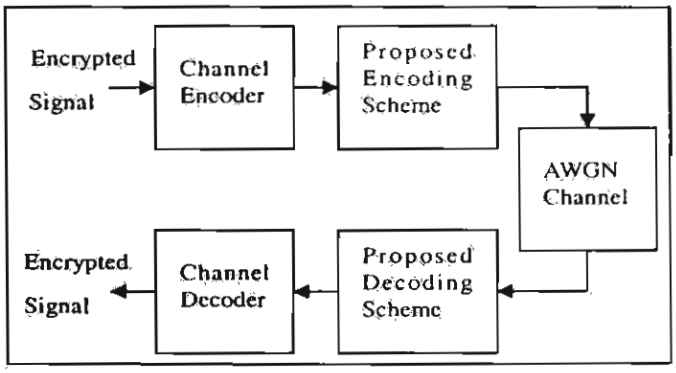

Figure 2. Simulation model in an AWGN channel

Moreover, the encoding scheme will be implemented with the MPEG coded stream by simulation method in the DVB systems where the powerful concatenated error control scheme is used. Commonly, the error control process in the DVB comprises three stages of forward error correction (FEC) coding, namely, outer coding, interleaving and inter coding, followed by a modulation stage where Quadrature Phase Shift Keying (QPSK) has been chosen. Normally, the Reed-Solomon (RS) 204:188 is used as an outer code, while the convolutional code with rate 1/2 is used as an inner code [7]. Figure 3 illustrates the concatenated error control scheme.

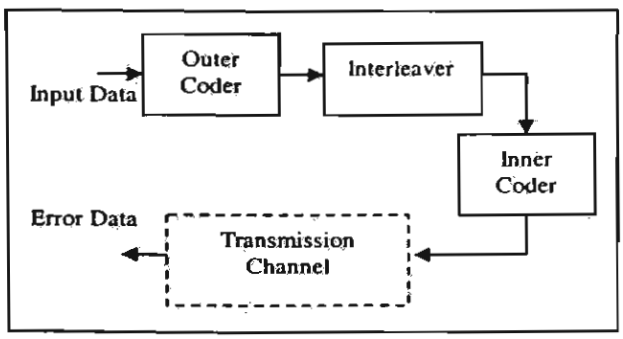

Figure 3. Concatenated error control scheme

4. RESULTS AND DISCUSSIONS

From the simulation results, the smallest chip-rate without errors after the decoding process was 46, 110, 455, 1100, 4150 and 12000 for a block size of 2, 3, 4, 5, 6 and 7 respectively. It can also be seen that the smaller the block size, the larger value the chip-rate required to recover the information bits correctly. For these block sizes, other values of the chip-rate considered resulted in different values of Bit Error Rate (BER) in the extracted information bits, and these values and the underlying line are shown in the figure below.

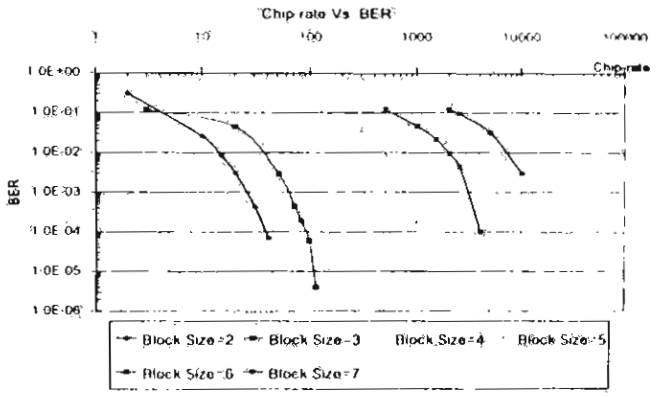

Figure 4. Bit error rate of extracted extra bits at different block sizes

From the figure 4, it can be seen that a larger block size needs a bigger chip-rate to retain the same BER. In addition, since one single bit error in the extracted information causes error propagation in the encrypted signal, any value other than a large chip-rate will result in a large BER. To illustrate the benefits of using error control codes, the number of encrypted data which is used to convey the extra bits is plotted against the BER. Figure 5 shows the performance comparison of the encoding scheme with and without the error control codes at the block size of 4.

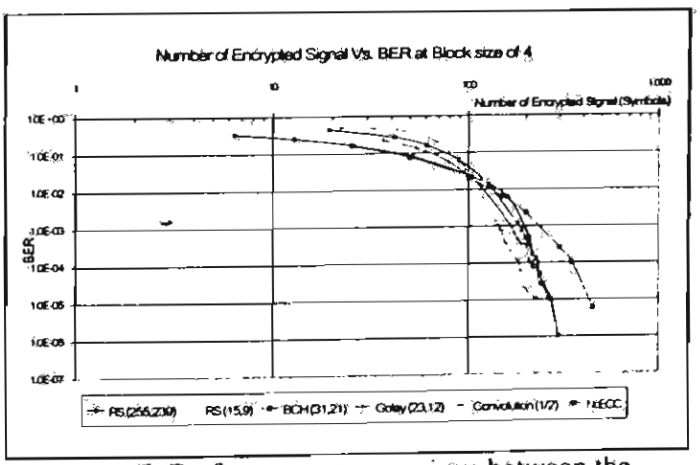

Figure 5. Performance comparison between the scheme with and without the error control codes at the block sizes of 4

It obviously shows, from Figure 5, that the error control codes provide the smaller values of the chip-rate. It also shows that the convolutional code gave the best performance, compared to the others. The summary of advantage of applying the error

control code i.e. the convolutional code with 1/2 rate is given in the table 1 below.

Table 1. Number of extra bits that can be transmitted with and without the convolutional code with 1/2 rate, at different block sizes

Block size	If Channel = 10Mbit/S, We can send the extra bits	
	With Convolutional code at m =	Non ECC at m =
3	51 kb/s	2.8 kb/s
4	16 kb/s	5.4 kb/s
5	2.7 kb/s	1.3 kb/s
6	520 bit/s	333 bit/s

To observe the performance of the scheme when implemented with DVB systems in the AWGN channel, the simulations were conducted according to the model in Figure 2. In the following Figure 6, a plot of the BER versus the E_b/N_0 for the scheme using the concatenated error control scheme mentioned in Section 3 is given.

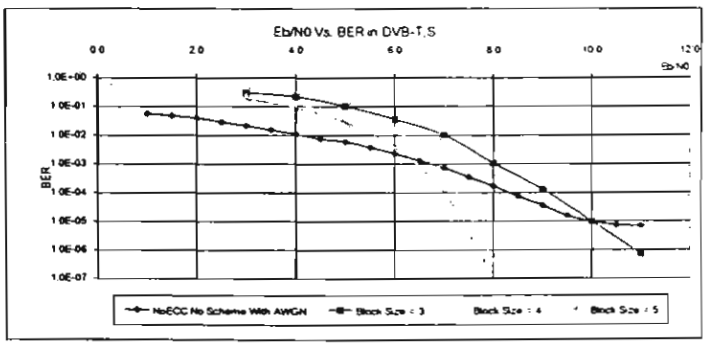

Figure 5. Performance of the scheme in DVB systems at different block sizes

From the figure, it can be seen that errors occurred at the output of the encrypted signal were quite high, compared to the DVB system alone. This is because of the unrecoverable errors remained in the data after decoding process, when the encoding scheme was applied. However, when the value of E_b/N_0 was increased the BER became smaller, especially with the larger block size. It can be noticed that the scheme operated with a larger block size gave better performance. Nevertheless, the value of the chip-rate required for large block size is enormous and this choice should be carefully considered.

5. CONCLUSIONS

In this paper we have shown a method of constructing an encoding scheme for dual level access to broadcasting network, based on the direct sequence spread spectrum technique. We have also shown experimentally and analytically that the scheme's performance was improved by applying the error control codes to the information bits before the encoding process. Furthermore, the encoding scheme was performed in DVB systems by simulation method, and the results have shown the possibility of implementing it in practice. At the end, our approach has showed an idea of how to utilize the existing allocated bandwidth in a more efficient way.

6. ACKNOWLEDGMENT

The authors would like to thank the Thailand Research Fund (TRF) for financial support throughout this work (Funding Code: PDF/27/2543).

REFERENCES

[1] T. Amornraksa, D. R. B. Burgess and P. Sweeney: "An Encoding Scheme for Dual Level Access to Broadcasting Networks", Proceedings of the Seventh IMA International Conference on Cryptography and Coding, Cirencester, UK, December 1999, LNCS 1746, pp. 114-118.

[2] R. Pickholtz, D. Schilling and L. Millstein: "Theory of Spread Spectrum Communications A Tutorial", IEEE Transaction on Communication, Vol. COMM-30, pp. 855-884, 1982.

[3] I. Cox, J. Kilian, T. Leighton and T. Shamoon: "Secure Spread Spectrum Watermarking for Multimedia", IEEE transactions on Image Processing, Vol. 6, No. 12, pp. 1673-1687, December, 1997.

[4] F. Hartung and B. Girod: "Watermarking of Uncompressed and Compressed Video", Signal Processing, Vol. 66, no. 3 (Special issue on Watermarking), pp. 283-301, May 1998.

[5] H. Stark and J. W. Woods: "Probability, Random Variables and Estimation Theory for Engineers", Prentice Hall, Englewood Cliffs, N. J. 1986.

[6] P. Sweeney: "Error Control Coding: An Introduction", Prentice Hall, London, 1991.

[7] G. M. Drury: "DVB Channel Coding Standards for Broadcasting Compressed Video Services", Electronics & Communication Engineering Journal, February 1997, pp. 11-20.