



รายงานวิจัยฉบับสมบูรณ์

โครงการ: เทคนิคการเข้ารหัสสำหรับการเข้าถึงระบบเครือข่าย
broadcasting ที่มีสองระดับ

โดย
อารงรัตน์ อมรรักษา

โครงการเสร็จสิ้น เดือนมิถุนายน พ.ศ. 2545



Final Report

**Project: Encoding Technique for Dual Level Access
to Broadcasting Networks**

**By
Thumrongrat Amornraksa**

Completed by June 2002

Final Report

Project: Encoding Technique for Dual Level Access to Broadcasting Networks

Researcher

Dr. Thumrongrat Amornraksa King Mongkut's University of Technology Thonburi

Mentor

Dr. Peter Sweeney University of Surrey, UK.

This research work is supported by the Thailand Research Fund (TRF)

Acknowledgments

Firstly, I would like to express my sincere gratitude to Dr. Peter Sweeney for his fruitful discussion and guidance he has given me for the last two years. Secondly, I would like to express my gratitude to members in the department of computer engineering, KMUTT, without whom this report might never have happened. I would also like to thank members of my family for their constant love, care and endless support they have provided me. Finally but not least, I would like to express my thankfulness to the Thailand Research Fund (TRF) for supporting this research work.

รหัสโครงการ : PDF/27/2543

ชื่อโครงการ : เทคนิคการเข้ารหัสสำหรับการเข้าถึงระบบเครือข่าย broadcasting ที่มีสองระดับ

ชื่อนักวิจัย : ดร. ชำรงรัตน์ อมรรักษา

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

อีเมลติดต่อ : t.amornraksa@cpe.eng.kmutt.ac.th

ระยะเวลาโครงการ : กรกฎาคม 2543 – มิถุนายน 2545

บทคัดย่อ

งานวิจัยชิ้นนี้นำเสนอกระบวนการเข้ารหัสสำหรับการเข้าถึงเครือข่ายแพร่กระจายสัญญาณวีดิทัศน์ดิจิทัลที่มีสองระดับ สัญญาณที่แพร่กระจายได้สร้างจากการเข้ารหัสข้อมูลขนาดเล็กลงในตัวข้อมูลหลักในลักษณะที่ว่า สัญญาณผลลัพธ์ที่ได้สามารถส่งผ่านช่องสัญญาณแพร่กระจายโดยไม่ต้องการความกว้างของการส่งสัญญาณเพิ่มเติม ที่ฝั่งผู้รับ สัญญาณที่รับเข้ามาจะถูกถอดรหัสออกเป็นสัญญาณดั้งเดิมทั้งสองสัญญาณได้ ซึ่งจะให้คุณสมบัติของการเข้าถึงแบบสองระดับ ความเป็นไปได้ในการใช้งานกระบวนการเข้ารหัสในทางปฏิบัติถูกตรวจสอบโดยใช้วิธีการทางแบบจำลอง โดยการประยุกต์ใช้กระบวนการเข้ารหัสในระบบแพร่กระจายวีดิทัศน์ดิจิทัล DVB ในช่องทางสื่อสารแบบ AWGN ประสิทธิภาพของกระบวนการเข้ารหัสสามารถวัดได้จากการทำงานของระบบในสถานะที่มีสัญญาณรบกวน ความเป็นไปได้ในการใช้งานกระบวนการดังกล่าวถูกอภิปรายโดยใช้ข้อมูลที่ได้รับจากการทดลองเป็นหลัก

คำหลัก กระบวนการเข้าถึงที่มีสองระดับ การกระจายแถบความถี่แบบอันดับโดยตรง การประมวลผลสัญญาณดิจิทัล

Project Code : PDF/27/2543

Project Title : Encoding Technique for Dual Level Access to Broadcasting Networks

Investigator : Dr. Thumrongrat Amornraksa

King Mongkut's University of Technology Thonburi

Email Address : t.amornraksa@cpe.eng.kmutt.ac.th

Project Period : July 2000 – June 2002

Abstract

This research work presents an encoding scheme for dual level access to digital video broadcasting network. The broadcast signal is generated by encoding small information into the main data in such a way that the resultant signal can be broadcast via a broadcast channel without requiring extra bandwidth. At the receiver end, the received signal can be decoded into two different sets of data, giving two different levels of access. The possibility of employing the scheme for practice use is investigated by a simulation method. The encoding scheme is implemented with a DVB system, while the resultant signal is broadcast through an AWGN channel. The scheme's performance is examined to determine how well it performs in the presence of noise. Based on the simulation results, the possibility of using the scheme in practice is discussed.

Keywords: Dual level access scheme, Direct sequence spread spectrum, Digital signal processing

Problem Statement

Presently, some multimedia data, which is broadcast through a communication network, are not effectively used, for example, in digital video broadcasting services. Since some programs are classified as pay-TV services, such programs will be protected by scrambling techniques before broadcasting to every user in the network. Only the authorized users who pay an extra fee can get access to those programs, while the rest cannot see anything from those channels. This method does not give any value at all to other users who have not paid for that particular program, since the allocated bandwidth is only used for broadcasting the protected signal to the authorized users, which may be a small group compared to all users in the network. It will therefore be more efficient if we can devise an encoding scheme in which the authorized users can access the protected signal and, at the same time, the others can receive something on the same channel, such as an advertisement. However, both information to be broadcast should not extend the existing allocated bandwidth.

Research Objective

In this research, we aim at developing such encoding scheme, which gives two different levels of access to broadcast networks. The improved performance of the scheme is achieved by the use of the error control codes. Furthermore, the possibility of implementing the scheme in practice, with digital video broadcasting (DVB) systems in AWGN channel is investigated.

Research Metrology

1. The encoding scheme based on direct sequence spread spectrum technique was designed and constructed. The performance of the scheme was measured. All parameters that effect the scheme's performance were identified e.g. the chip-rate required in the encoding scheme and the Bit Error Rate (BER) in the decoded data. All data used in the simulation were generated by a pseudo-random bit generator.
2. Various errors control codes were studied and applied in the encoding scheme, in order to determine the one that gave the highest improvement to the scheme i.e. the lowest BER and chip-rate required in the encoding process.
3. The encoding scheme was examined for practical use by operating it in a presence of noise. That is, the output signal from the scheme was broadcast through an AWGN channel. The plots between E_b/N_0 and the BER was used to represent the performance of the scheme.
4. The encoding scheme was implemented with the digital video broadcasting (DVB) systems by simulation method. The output signal from the scheme was also broadcast through an AWGN channel. The performance in term of E_b/N_0 and BER obtained after applying the encoding scheme was presented and compared with the performance obtained from the DVB systems alone.

Research Scope

In this research, we designed an encoding technique, based on direct sequence spread spectrum, for dual level access to broadcast networks. The encoding scheme was constructed and its operation was observed by simulation method on Personal Computer (PC). A suitable error control code was determined for improving the performance of the scheme. The possibility of implementing the encoding scheme with the DVB systems was investigated.

Equipment

This research was conducted at the Multimedia Communications Laboratory, which is a part of the Department of Computer Engineering at KMUTT. Most of the results were obtained by running a number of simulations on Personal Computer (PC). The programming language used in the simulation were Turbo C and MATLAB.

Description of the Scheme

Spread Spectrum (SS) systems have been developed since about the mid-1950, and its characteristic can be described as follows [1]: "SS is a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the band spread is accomplished by means of a code which is independent of the data, and a synchronized reception with the code at the receiver is used for despreading and subsequent data recovery". According to its characteristic, a band spread signal can be easily hidden within the same spectrum of another band spread signal, where each signal appears to be noise to the other. The core component of these SS systems is a key-based Pseudo-Random Sequence (PRS) generator. Since the signal is spread and modulated by a PRS to produce a band spread signal, only those receivers equipped with correct PRS can despread and recover the original signal. Although, at the receiver, the band spread signal may be accompanied by noise, which is assumed to be completely random and uncorrelated, by using a suitable demodulator with the correct PRS, this spread signal can be squeezed back into the original signal.

The technique used to construct an encoding scheme for dual level access is based on the concept described above. That is, the first band spread signal is obtained by spreading the extra bits and then modulated with the PRS, while the second band spread signal is merely the main data itself. To implement the encoding scheme in digital video broadcasting applications, MPEG (Moving Picture Expert Group) based compressed video signal is considered as the main data. The extra bits after being spread and modulated are added to the main data. The result is then input to the channel encoder and modulator to obtain the broadcast data. Given a key to reproduce the same PRS at the receiver end, the

extra bits can be recovered from the data, which is already demodulated and channel decoded. Note that all errors occurring at this stage i.e. communication channel errors are assumed to be detected and corrected by the channel decoder. This data will later be subtracted by the recovered extra bits, which are already spread and modulated to obtain the original like MPEG bit-stream. The block diagram of the encoding scheme is illustrated in

Figure 1.

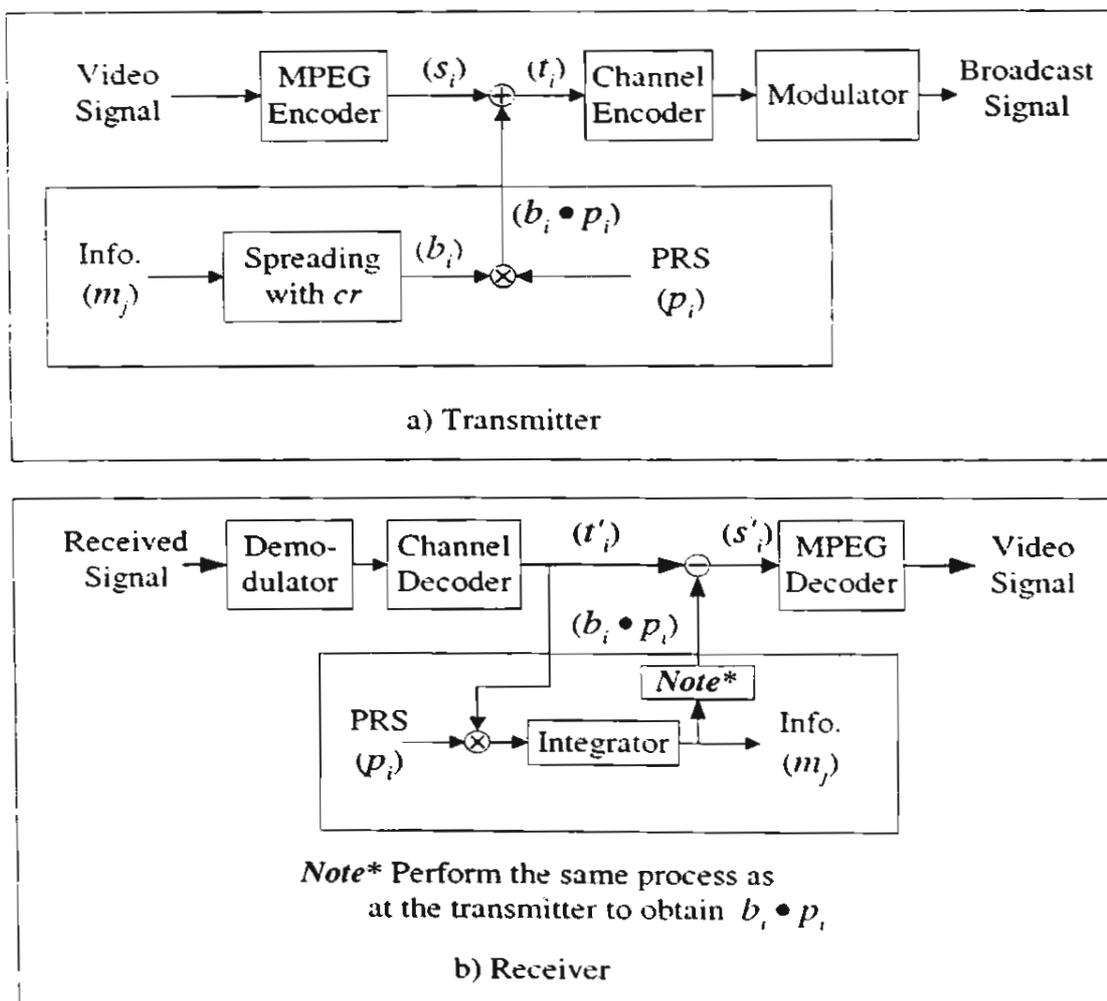


Figure 1. The block diagram of the encoding scheme

In the MPEG based compression scheme, the video sequence is compressed frame-by-frame using block based coding techniques, which apply some sort of transform coding such as Discrete Cosine Transform (DCT). Basically, the encoder calculates the DCT of each 8×8 block, transforming that block into its frequency-based representation, and then applies the processes of thresholding, quantization, zigzag scan, run-level-coding and entropy coding. Note that, in the 8×8 transformed block, the coefficient in the top left corner (coordinate: 0,0) which represents null horizontal and vertical frequencies is called the DC coefficient, whereas the rest of the coefficients are called AC coefficients. According to the MPEG compression standard, the incoming MPEG bit-stream will be split into header, side information, motion vectors (for motion compensation) and DCT encoded data blocks [2]. Only AC coefficients are altered and used in our encoding scheme to carry the extra bits, in order to minimize the visible changes in the image frames. We now describe the basic steps of adding the extra bits to the main data [3]. Firstly, the extra bits m_i , which consists of a set of -1 and 1, are spread by a large factor, called chip-rate, to obtain the spread sequence b_i . This sequence is then modulated via a multiply operation with a PRS (p_i), which consists of a set of -1 and 1, and directly added to the AC coefficients extracted from the MPEG bit-stream s_i , one bit per one coefficient (or block), where each s_i block contains k bits. The encoded AC coefficients t_i , $t_i = s_i + p_i \bullet b_i$, is finally grouped with the header, side information, motion vectors and DC coefficient to reconstruct the MPEG bit-stream.

At the receiver end, the recovery of the extra bits can be accomplished by correlating the encoded AC coefficients t'_i , extracted from the received MPEG bit-stream, with the same PRS that was used in the encoding process. Correlation here is performed via a multiply operation, followed by summation over the width of the chip-rate. If the peak of the

If correlation is positive (or respectively, negative), the recovered bit is a +1 (or -1) [4]. These extra bits are then spread and modulated again with the same parameters used in the encoding scheme, before subtracting them from the encoded AC coefficients. Finally, the resultant AC coefficients s'_i are used to reconstruct the original like MPEG bit-stream.

As an example, let the main data be an image with the size of 512X512 pixels and the chip-rate be 100. With this image size, there will be 258048 AC coefficients to carry the extra bits. Then, the number at which extra bits can be added is 2580 bits. To increase the number of extra bits, the chip-rate and the block size must be reduced. However, a smaller block size implies a greater likelihood that subtracting the extra bits from the encoded AC coefficients will not give the original AC coefficients i.e. some errors are introduced to the reconstructed image. In fact, it is shown in [1] that a smaller chip-rate implies a greater likelihood of error occurred in the recovered extra bits.

Simulation Model

In this research work, two sets of experiment were conducted. The first one, which consists of experiment 1, 2 and 3, focused on the worst case that the encoding scheme may encounter. The experiment was carried out using simulation method, where both extra bits and main data were generated by a pseudo-random bit generator. It should be noted that 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. In contrary, the second one, which consists of the experiment 4 and 5, focused on realistic data, where the gray-scale images with the size of 256×256 pixels were considered as an image frame from a video sequence, and used as the main data in the experiment. In the first two experiments, however, the encoding scheme will be performed in an error-free communication channel. That is, the errors that occurred in the main data come solely from the need to remain within the bandwidth of the broadcast channel. The objective for doing this is to focus on only the errors that occur in the recovered extra bits, which are mainly related to the performance of the scheme.

Experiment 1: Performance of the encoding scheme

According to the encoding process, the addition between s_i and $\rho_i \bullet b_i$ can be carried out by five different methods, yielding five operations, as follows:

- i) $t_i = s_i + \rho_i \bullet b_i$
- ii) $t_i = s_i$, if $s_i = 0$ and $\rho_i \bullet b_i = -1$,
or $s_i = (2^k - 1)$ and $\rho_i \bullet b_i = 1$,
Otherwise $t_i = s_i + \rho_i \bullet b_i$
- iii) $t_i = s_i$, if $s_i = 0$ and $\rho_i \bullet b_i = -1$,
otherwise $t_i = (s_i + \rho_i \bullet b_i) \bmod 2^k$
- iv) $t_i = s_i$, if $s_i = (2^k - 1)$ and $\rho_i \bullet b_i = 1$,
otherwise $t_i = (s_i + \rho_i \bullet b_i) \bmod 2^k$
- v) $t_i = (s_i + \rho_i \bullet b_i) \bmod 2^k$

Table 1 shows five possibilities of t_i resulted from five different addition methods, which can be used in the encoding process.

Table 1. Possible values resulted from five addition methods at block size $k = 2$

Method	$s_i + p_i \bullet b_i$							
s_i	-0	0	1	1	2	2	3	3
$p_i \bullet b_i$	-1	1	-1	1	-1	1	-1	1
i) t_i	-1	1	0	2	1	3	2	4
ii) t_i	0	1	0	2	1	3	2	3
iii) t_i	0	1	0	2	1	3	2	0
iv) t_i	3	1	0	2	1	3	2	3
v) t_i	3	1	0	2	1	3	2	0

As shown in Table 1, the method i produces some results that are out of the range of the values that the original main data t_i can represent. For instance, the value of 4 cannot be represented by 2-bit number. Thus this method will not be considered and used in practice since the increased bandwidth for broadcasting is required. For the remaining methods, the different values of t_i exist when performing the addition between $s_i = 0$ and $p_i \bullet b_i = -1$, or $s_i = (2^k - 1)$ and $p_i \bullet b_i = 1$. Hence, in this research part, the methods $ii-v$ were used in the simulation, with the aim of demonstrating how an encoding scheme may be constructed as well as how well it performs. The differences when applying each addition method were then analyzed, based on the simulation results obtained.

Results and Discussions

From the simulation results, the smallest value of chip-rate, which gives no errors in the recovered extra bits, using four different addition methods (ii-v) are shown in Table 2. Note that the block size k was varied from 2-7 bits to represent up to 128 values.

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

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

According to Table 1, it can be seen that the addition method v provides reversible decoding, i.e. by using the knowledge of $p_i \bullet b_i$, every value of t_i can be referred back to s_i in the same way as one-to-one mapping. However, as shown in Table 2, no matter how large the chip-rate is, when method v was used in the encoding scheme, the extra bits could not be recovered. Since the inaccurate results obtained from the summation over the correlation window in the decoding process, the decoder will give a wrong sign of correlation result, and translate to a wrong value of m_j . This incident can be noticed by observing whether the original main data is random or not. If so, it is most likely that the correlation results from the summation process will lead to a wrong value of m_j .

A good example that verifies this notification can be seen from Table 1, where the original main data s_i is equally distributed, i.e. each value of s_i has the same probability of occurrence. It is obvious that the summation term of all possible values of $t_i \times (p_i \bullet b_i)$ in each addition method is 8, 6, 3, 3 and 0, respectively. For example, in method *ii*, the summation term can be calculated as follows: $(0 \times -1) + (1 \times 1) + (0 \times -1) + (2 \times 1) + (1 \times -1) + (3 \times 1) + (2 \times -1) + (3 \times 1) = 6$. It can then be noticed that the larger the value of summation term, the smaller the chip-rate required to recover the extra bits correctly. This analytical observation is verified by the simulation results shown in Table 2. Therefore, the method *v* is not used for the encoding process.

Now let's consider other addition methods i.e. methods *ii-iv*. A problem occurs when performing the subtraction between $t'_i = 0$ and $p_i \bullet b_i = -1$, or $t'_i = (2^k - 1)$ and $p_i \bullet b_i = 1$, since the decoder will not be able to determine whether s'_i is 0 or 1. This event gives the possibility of making a wrong decision at the decoder up to $1/(2^k)$ %, where k is the block size of main data used in the encoding process. However, when the addition methods *iii* was used, the chip-rate required in the decoding process must be increased in order to prevent any error occurred in the recovered extra bits. The results shown in Table 2 already verified this fact. The same explanation can also be applied to the addition method *iv*.

From Table 2, it can be concluded that the addition method *ii* gave the best performance, i.e. it required the smallest value of chip-rate for correctly recovering the extra bits, especially in the larger block sizes, compared to others. Therefore, from this point, the addition method *ii* will be used in our simulations to measure the performance of the encoding scheme. Since the values of chip-rate shown in Table 2 were the smallest ones that provide the recovered extra bits with no error, other values of chip-rate considered

resulted in different values of bit error rate, left in the recovered extra bits. These values and the underlying line between the chip-rate and the BER are illustrated in Figure 2.

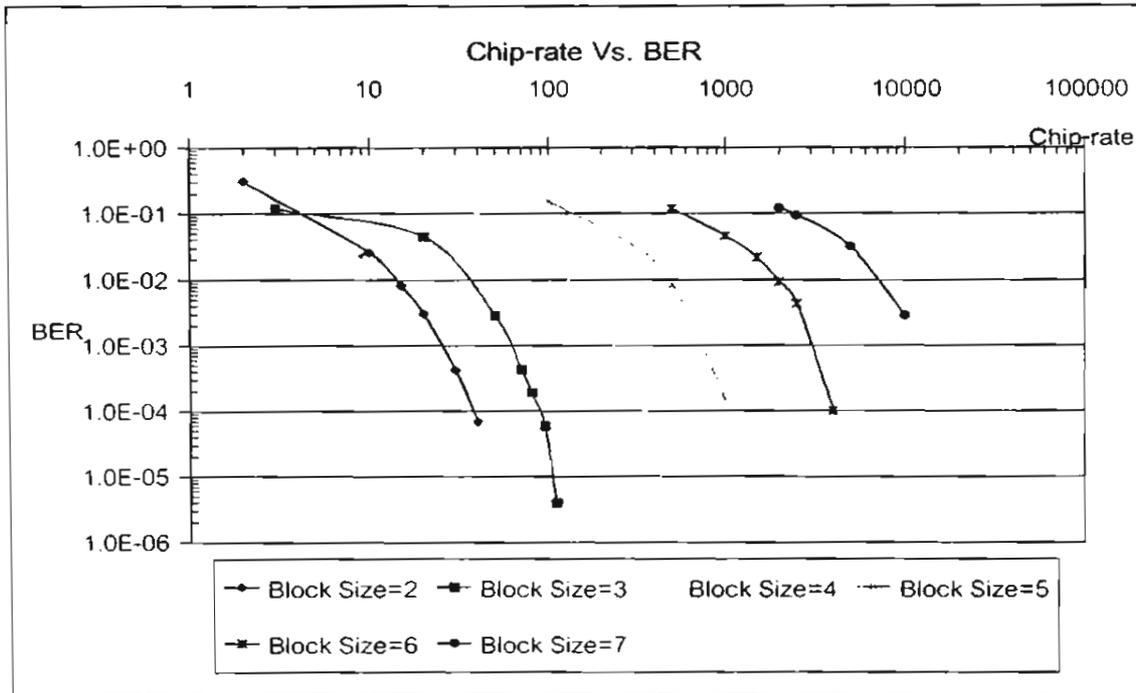


Figure 2. BER of the recovered extra bits at different block sizes

From the figure 2, 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 recovered extra bits causes error propagation in the original signal, any value other than a large chip-rate will result in a large BER. Therefore, according to the simulation results obtained, we conclude that the encoding scheme should be operated with a small block size, in order to achieve the optimal performance. In addition, the smallest value of chip-rate required in the encoding process was 46, 110, 455, 1100, 4150 and 12000 for a block size of 2, 3, 4, 5, 6 and 7 respectively.

ศูนย์บริการข้อมูลสนับสนุนการวิจัย (สทว.)
 ชั้น 14 อาคาร 103 เอ็ม ทาวเวอร์
 เลขที่ 979 17-21 ถนนพหลโยธิน แขวงสามยุคใหม่
 เขตจตุจักร กรุงเทพมหานคร 10400
 โทร 298-0455 โทรสาร 298-0476
 Home page : <http://www.trf.or.th>
 E-mail : trf-info@trf.or.th



Experiment 2: Improved performance from the use of error control codes

When the error control codes (ECC) are applied to the extra bits, before the spreading process, it will of course reduce the main throughput by a constant value [5], which one may think that this may be difficult to compensate by a smaller chip-rate in the decoding process. To demonstrate that the error control codes can improve the performance of the encoding scheme, various codes e.g. Reed Solomon, Binary BCH, Golay and Convolutional codes are directly applied to the extra bits, and their performance are then compared to the original encoding scheme's.

Results and Discussions

The benefit of using the error control codes was demonstrated, where the Reed Solomon (15, 9) code was selected and used in the simulation. The plot between the smaller value of chip-rate required when the RS (15, 9) code was used and the BER of the recovered extra bits is illustrated in Figure 3.

To observe the improved performance of the encoding scheme, the plot between the chip-rate and the BER of the scheme with and without the RS (15, 9) code at the block sizes of 2 and 4 is illustrated in Figure 4. Moreover, the summary of efficiency improvement when the RS (15, 9) code was used is given in Table 3.

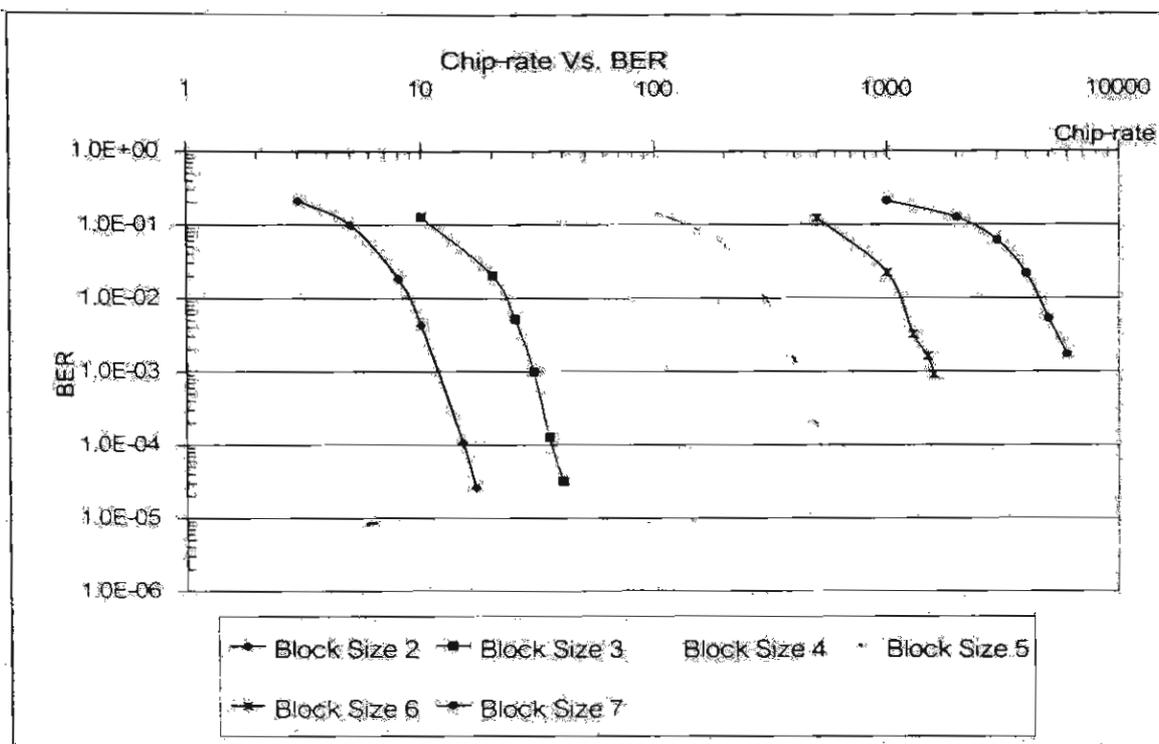


Figure 3. BER of the recovered extra bits when the RS (15, 9) code was used at different block sizes

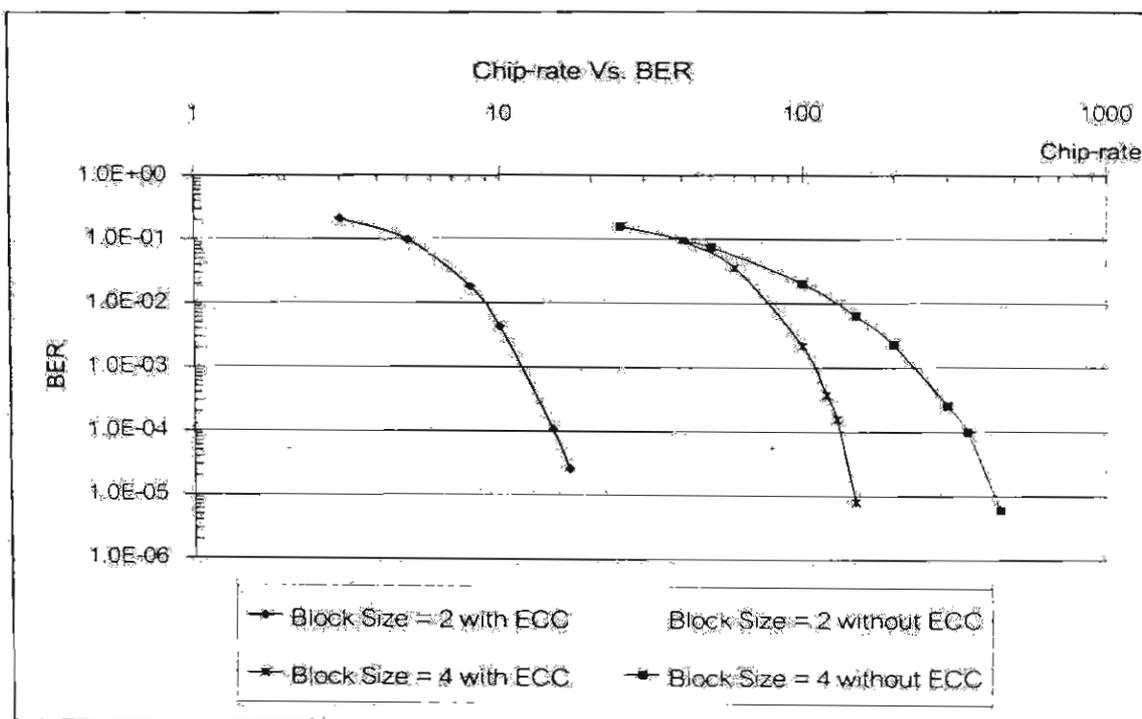


Figure 4. Performance comparison between the scheme with and without the RS (15, 9) code at the block sizes of 2 and 4

Table 3. Efficiency improvement when the RS (15, 9) code was used at different block sizes

Block Size k	2	3	4	5	6	7
Efficiency (%)	33.33	33.33	39.07	38.89	46.08	44.17

It can be seen from Figure 3 and 4 that the error control code used in the encoding scheme provides a smaller value of chip-rate. Although the RS (15, 9) code reduces the amount of extra bits to be added to the main data by 40 %, the improvement in total can still be obtained. As Table 3 shows, up to 46.08 % improvement was achieved when the encoding scheme was operated with the RS (15, 9) code at the block size of 6. To determine the type of error control code that give the best performance, the performance comparison of the encoding scheme with various types of error control code, i.e. RS (255, 239), RS (15, 9), BCH (31, 21), Golay (23, 12) and $\frac{1}{2}$ rate Convolutional codes, at the block size of 4 is illustrated in Figure 5. Note that the performance curves were plotted in such a way that the amount of main data required to convey the fixed number of extra bits was varied upon the type of error control code used in the encoding scheme.

It can be seen from Figure 5 the $\frac{1}{2}$ rate convolutional code gave the best performance, compared to the others, and hence will be selected and used in the encoding scheme from now on. The summary of efficiency improvement in term of number of extra bits that can be added to the main data, when the $\frac{1}{2}$ rate convolutional code was used, is given as an example in Table 4.

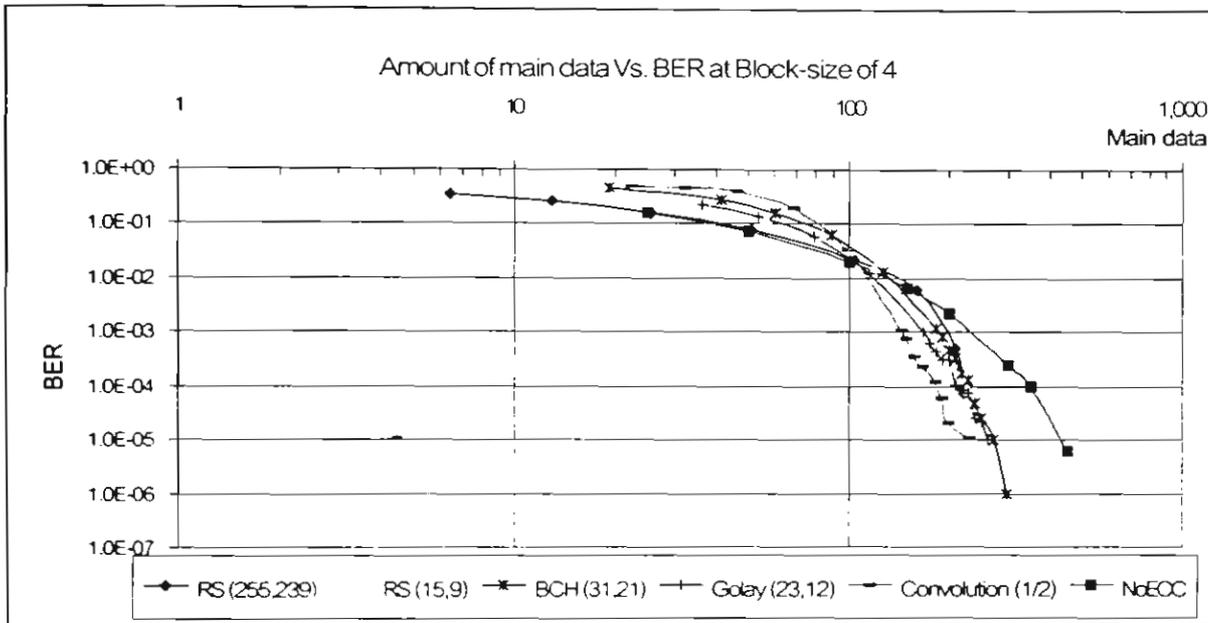


Figure 5. Performance comparison of the scheme with different error control codes at the block sizes of 4

Table 4. Number of extra bits to be added with and without the $\frac{1}{2}$ rate convolutional code at different block sizes

Block size	If main data = 10Mbit/s, the number of extra bits we can add	
	With ECC	Without ECC
3	51 kb/s	28 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

According to the simulation results obtained, it can be concluded that with the use of error control codes the encoding scheme give a better performance when operating with a larger block size. In other words, the error correcting codes would be very useful when the encoding scheme is operated with a large block size. However, the value of the chip-rate required for large block size is enormous and this choice should be carefully considered. In addition, the $\frac{1}{2}$ rate convolutional code is the most suitable one since it gave superior performance compared to the others.

Experiment 3: Implementing the scheme in AWGN channel

After an appropriate error control code that gives the best performance was found, the encoding scheme will then be simulated in a broadcast channel. At this step, the resultant signal from the encoding scheme is encoded again by the channel coder before broadcasting through an Additive White Gaussian Noise (AWGN) channel. We chose the AWGN channel since it is a type of noise that most communication systems encounter [6]. The block diagram of the model used in the simulation is illustrated in Figure 6.

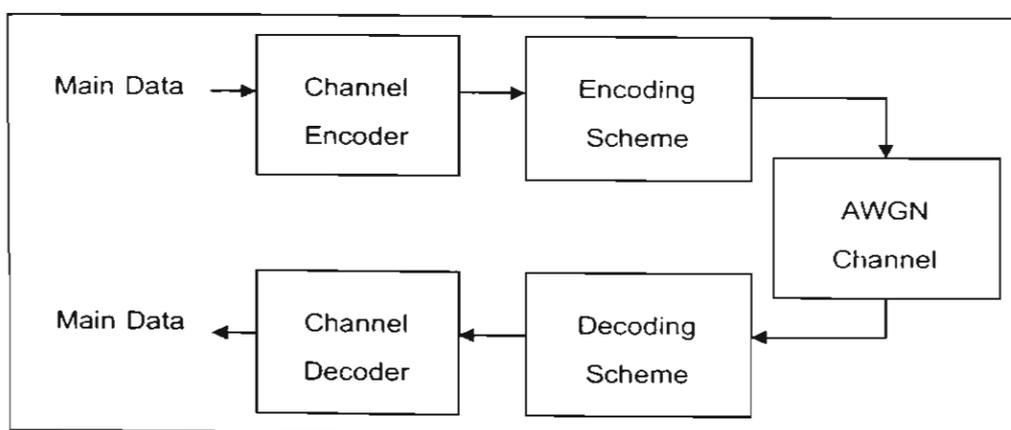


Figure 6. Block diagram of the simulation model in an AWGN channel

In this part, the performance of a transmission system implementing our encoding scheme is evaluated in order to observe whether the channel decoder can detect and correct the errors which occur at the receiver's side, whether from the communication channel or the decoding process or not. The channel coders used in the simulation are as follows: RS (255, 239), BCH (31, 21) and $\frac{1}{2}$ rate Convolutional codes.

Results and Discussions

When the extra bits, which already encoded by the $\frac{1}{2}$ rate convolutional code, was added to the main data, the resultant signal was then encoded by the RS (255,239,8), BCH (31,21) and $\frac{1}{2}$ rate Convolutional code before sending it to the AWGN channel, and the plots between the BER and the E_b/N_0 at various block sizes for each type channel coder are illustrated in Figure 7, 8 and 9, respectively.

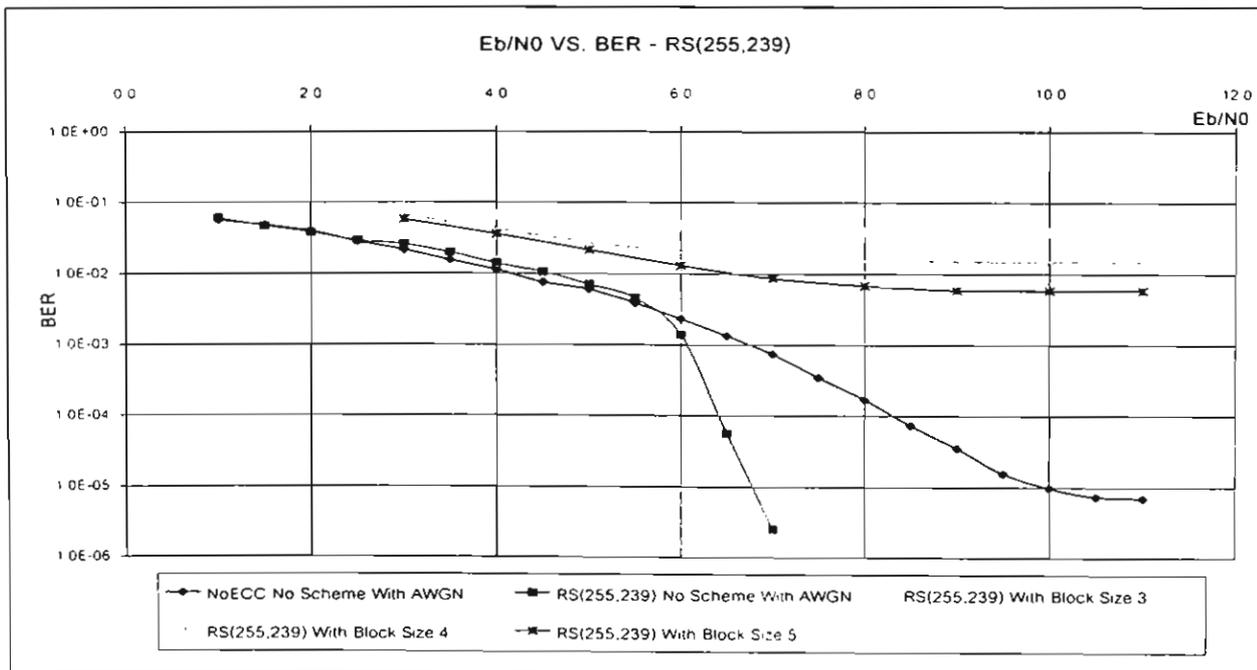


Figure 7. Performance comparison of a system implementing the scheme with the RS (255, 239) channel coding at different block sizes

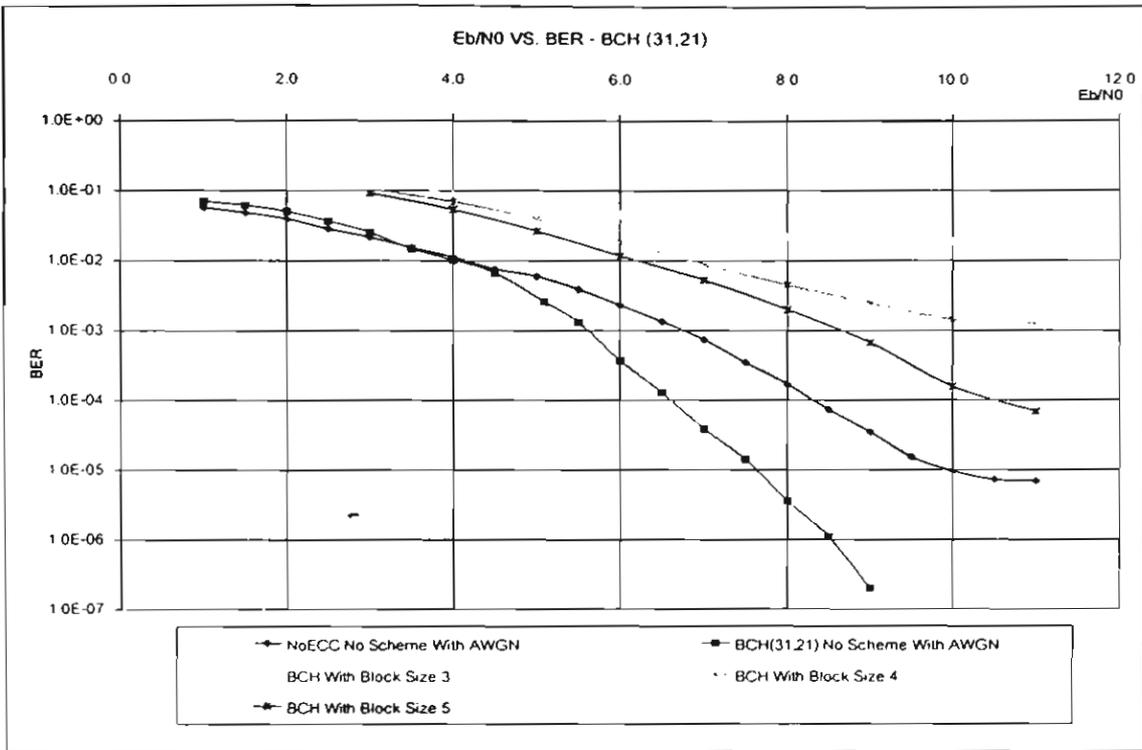


Figure 8. Performance comparison of a system implementing the scheme with the BCH (31, 21) channel coding at different block sizes

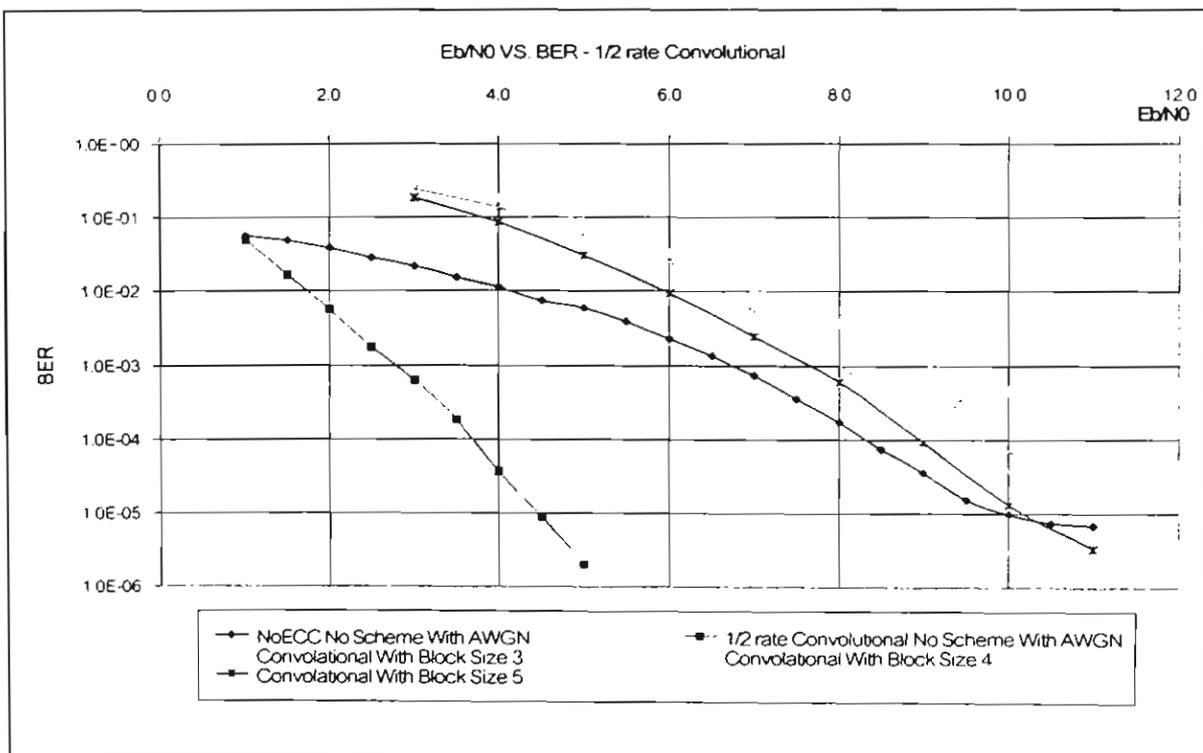


Figure 9. Performance comparison of a system implementing the scheme with the 1/2 rate Convolutional channel coding at different block sizes

From the above Figures, it can be seen that the error rate occurred in the output signal from the system implementing the encoding scheme was larger than the one from the system without the scheme. This is because of the combination between the errors from the communication channel and the decoding process was too large to be detected and corrected by the channel decoders. However, when the value of E_b/N_0 was increased the BER became smaller, especially with the larger block size. According to the results, it can be concluded that, when implementing the encoding scheme with an application in a broadcast channel, the scheme should be operated with a large block size. Furthermore, the performance is getting worse since the numbers of errors that the channel decoders cannot correct is greater than that of the application without implementing the scheme.

Experiment 4: Implementing the scheme with digital video broadcasting system

In this part, the performance of the encoding scheme is evaluated for practical use by implementing the scheme with the DVB system, where a powerful error control scheme, or the concatenated error control scheme, is used. Commonly, the error control scheme in the DVB comprises three stages of forward error correction (FEC) coding, namely, outer coding, interleaving and inner coding, followed by a modulation stage where Quadrature Phase Shift Keying (QPSK) is chosen. Normally, the RS (204,188) is used as an outer code, while the $\frac{1}{2}$ rate Convolutional code is used as an inner code [7]. Figure 10 illustrates the concatenated error control scheme.

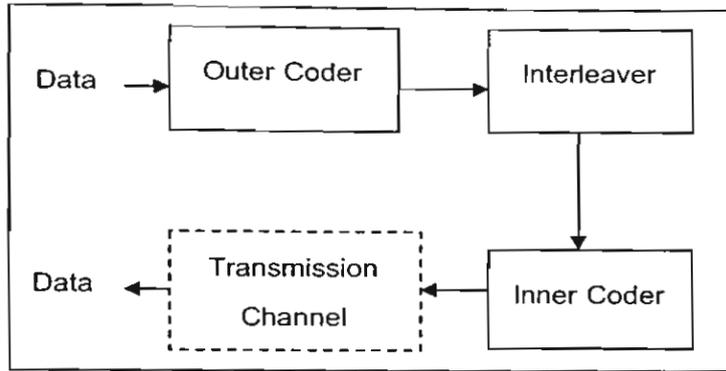


Figure 10. Concatenated error control scheme

In the simulation, the output signal from the DVB system is again broadcast through an AWGN channel. The performance of the DVB system implementing the encoding scheme is evaluated in order to observe whether the concatenated error control scheme can detect and correct all errors which occur at the receiver's side or not.

Results and Discussions

A plot between the BER and the E_b/N_0 when implementing the scheme at different block sizes in the DVB system is illustrated in Figure 11. Note that in this plot, to observe the worst case that the encoding scheme may encounter, both extra bits and main data were generated by a pseudo-random bit generator. The next step, we used the MPEG bit-stream as the main data, and its plot is illustrated in Figure 12.

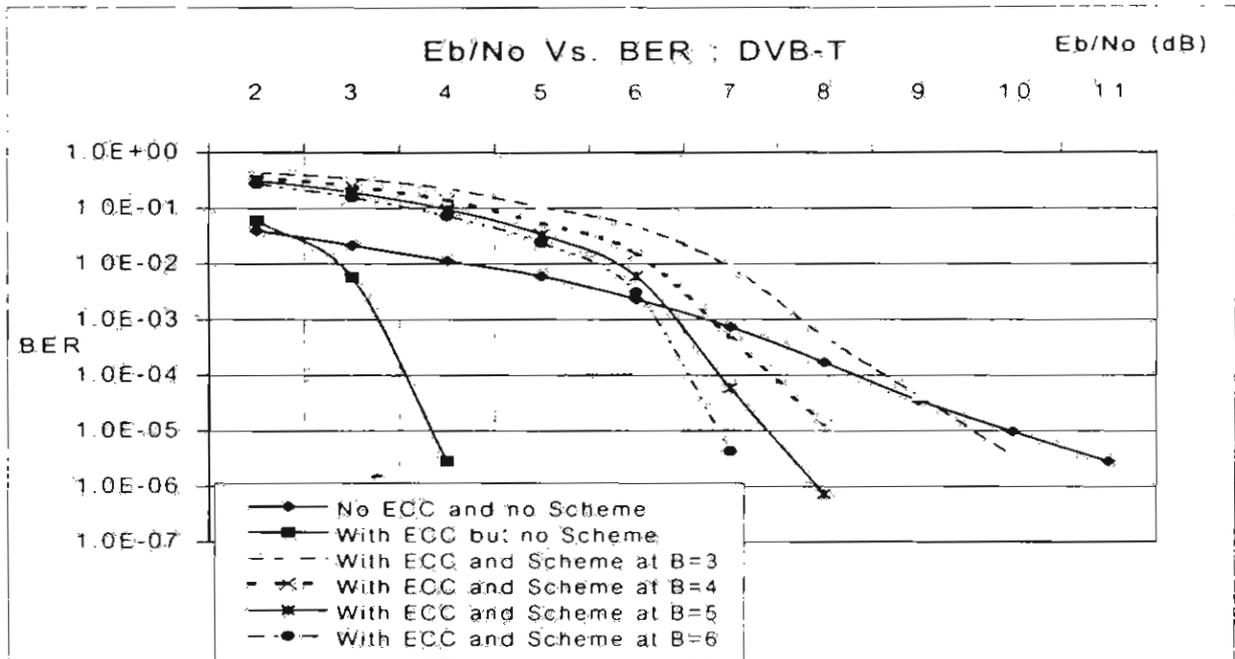


Figure 11. Performance of the encoding scheme in the DVB systems at different block sizes, using pseudo-random bits as main data

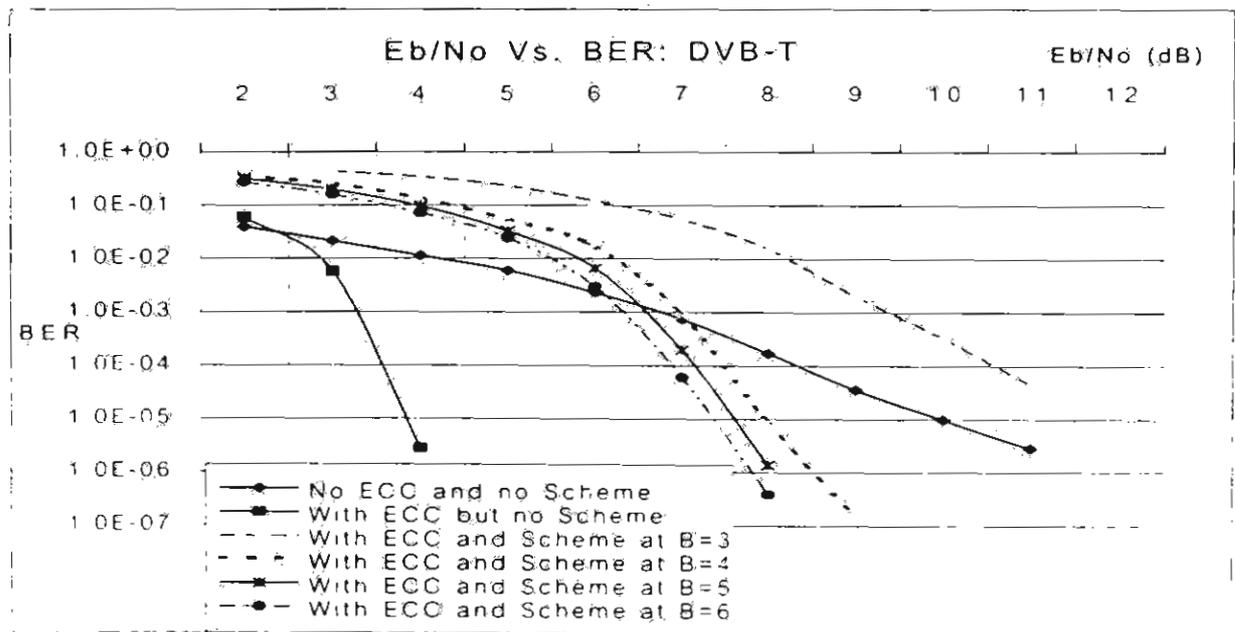


Figure 12. Performance of the encoding scheme in the DVB systems at different block sizes, using MPEG bit-stream as main data

From both Figures, it can be seen that errors occurred in the resultant signal were quite high, compared to the DVB system alone. For example, in Figure 11, at BER of 10^{-4} and the block size of 6, the encoding scheme required more Eb/No, approximately 3.5 dB, to achieve the transmission of extra bits without requiring extra bandwidth. Although the scheme used the MPEG bit-stream, which contained some redundancies as the main data, it still required at least 3 dB of Eb/No to achieve such goal. However, it should be noticed from the simulation results that, when implementing the encoding scheme with the DVB system, the scheme operating with a larger block size gave better performance.

Experiment 5: Quality of the resultant images

Since we cannot correct all errors occurred in the main data, caused by the symbol-mapping problem in the decoding process, such errors will be left in the main data with an assumption that the errors in the improperly compressed data would be unobservable by Human Visual System (HVS). In the MPEG based compression scheme, the video sequence is compressed frame-by-frame. To observe the quality degradation, a number of images having different various textures, such as *Airplane*, *Tiffany*, *Splash*, etc. were used. The MPEG based compression method for Intra-frames (reference frames) of a video sequence was used to prepare the testing images in compressed form. The quality of the decoded image was evaluated by the value of PSNR obtained from each image frame at various block sizes. Basically, the PSNR can be determined by the following equation.

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2}{\frac{1}{N} \sum_{i=1}^N \left(P_{\text{org}}(x,y) - P_{\text{mod}}(x,y) \right)^2} \right) \quad (1)$$

where N is total number of pixels within an image frame, $P_{org}(x, y)$ and $P_{mod}(x, y)$ are the values of pixel (x, y) in the original and modified images, respectively. According to the equation, high value of PSNR implies that the quality of the modified image is as close as that of the original one.

Results and Discussions

Table 5 and 6 show the smallest value of chip-rate required and the maximum number of extra bits that can be added, at various block sizes for each type of image, respectively.

Table 5. The smallest value of chip-rate required at various block sizes

Chip-rate required to recover the added bits correctly					
Image Frame	Block size of 3	Block size of 4	Block size of 5	Block size of 6	Block size of 7
Airplane	20	65	235	320	1890
Tiffany	20	75	220	620	2300
Splash	22	55	168	580	2000
Boats	20	95	270	480	1200
Barbara	18	68	245	560	1300
Average	20	72	228	512	1738

Table 6. The maximum number of added extra bits at various block sizes

Image Frame	Number of extra bits that can be added (bits)				
	Block size of 3	Block size of 4	Block size of 5	Block size of 6	Block size of 7
Airplane	3267	1008	278	204	16
Tiffany	3267	873	297	105	16
Splash	2978	1191	390	112	20
Boats	3276	698	242	136	22
Barbara	3640	693	267	117	26
Average	3285	892	294	134	20

In this simulation, the case of 2-bit block size was omitted because the remaining errors from the use of 2-bit block size was too high to be used in practice i.e. nearly one fourth of the main data's size. Also we omitted the case of 8-bit block size because we could not get the considerably improved performance with more than 7-bit. From the results, we can see that the performance of the encoding scheme was significantly improved when implemented with the realistic data. For instance, the smallest value of chip-rate required to recover the extra bits correctly, at the block size of 4, was reduced from 455 to 95 (at the worst case). This resulted in the increasing number of extra bits that can be added in the main data by up to 4.8 times.

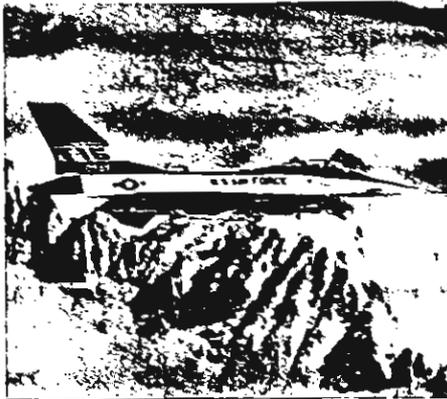
As stated earlier, there will be some remaining errors around $1/(2^k)$ % in the main data. This error rate can be improved by increasing the block size used in the decoding

process. To observe how much the remaining errors effect the quality of the main data i.e. the image frame, the values of PSNR obtained from the resultant images are measured and the results are shown in Table 7.

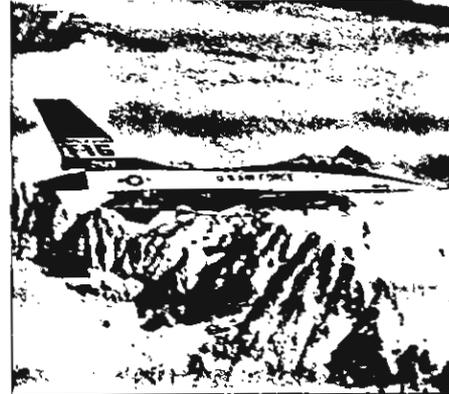
Table 7. The values of PSNR obtained from the decoded image at various block sizes

Image Frame	Peak Signal to Noise Ratio (dB)				
	Block size of 3	Block size of 4	Block size of 5	Block size of 6	Block size of 7
Airplane	51.0898	51.0502	51.1261	51.0571	54.1345
Tiffany	50.0815	50.9584	51.1394	51.2789	54.0508
Splash	51.0580	51.0352	51.1021	50.9992	54.0674
Boats	51.0922	51.0688	51.1027	51.5036	54.6165
Barbara	51.0905	51.0746	51.0528	51.2181	54.0342
Average	50.8824	51.0374	51.1046	51.2114	54.1807

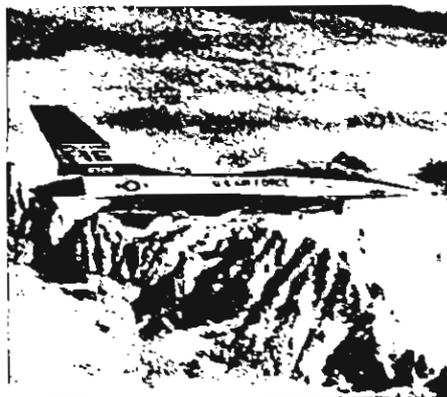
From Table 7, when increasing the block size from 3 to 7-bit, the PSNR of the resultant image frame was improved by 3.3 dB, on average. However, in the worst case at the block size of 3, we could not notice any visual differences between the original and the resultant images. From the experimental results, the average value of PSNR obtained at the block size of 3 was above 50 dB, which is considered as a very good quality image. Examples of the images with the remaining errors resulted from the decoding process at various block sizes are illustrated in Figure 13.



(a)



(b)



(c)



(d)

Figure 13. The resultant image 'Airplane' at various block sizes

(a) original image (b) block of 3 (c) block of 5 (d) block of 7

It can be seen that although the encoding scheme operated with a larger block size gave lower error rate in the resultant images, the errors occurred from a smaller block size do not have any visual effect to the human eye. Therefore, operating the scheme with a small block size is the best choice, and should be selected for practical use e.g. in the digital video broadcasting networks.

Conclusions and Future Work

A method of constructing an encoding scheme for dual level access to broadcasting network, based on the direct sequence SS technique has been presented in this research work. It was also shown experimentally and analytically that our encoding scheme could be used for transmitting extra bits, together with the broadcast data in compressed form. The number of extra bits to be added in the main data depends on the block size k , the types of error control code used in the encoding scheme and the characteristics of the main data i.e. texture. The experimental results showed that, at the block size of 3 with (15,9) RS code, up to 0.627 % of the main data can be transmitted without requiring extra bandwidth. Since there were some errors up to $1/(2^k)$ % remaining in the main data, the quality of the main data in term of PSNR was degraded. The experiments showed that the PSNR values obtained from the resultant images, which were used as the main data, were approximately 50-54 dB, depending on the block size used. However, the human visual system i.e. human eye could not notice any visual degradation. In summary, we have shown the possibility of implementing in practice the concept of dual level access in digital video broadcasting networks. 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 applications with employ an efficient channel encoding technique, and observe its performance.

References

- [1] 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.
- [2] J. Mitchell, W. Pennebaker, C. Fogg and D. Legall, "*MPEG Compression Standard*", Chapman & Hall, New York, 1997.
- [3] T. Amornraksa and P. Sweeney, "*Dual Level Access Scheme for Broadcasting Networks*", *Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001)*, Orlando, USA, July 22-25, 2001, volume V, pp.7-12.
- [4] H. Stark and J. W. Woods, "*Probability, Random Variables and Estimation Theory for Engineers*", Prentice Hall, Englewood Cliffs, N.J. 1986.
- [5] P. Sweeney, "*Error Control Coding: An Introduction*", Prentice Hall, London, 1991.
- [6] S. Lin and D. J. Jr. Costello, "*Error Control Coding: Fundamentals and Applications*", Prentice-Hall, Englewood Cliffs, NJ, 1983.
- [7] U. Reimers, "*Digital Video Broadcasting (DVB): The International Standard for Digital Television*", Imprint Berlin, Springer, 2001.

Output จากโครงการวิจัยที่ได้รับทุนจาก สกว.

1. ผลงานตีพิมพ์ในวารสารวิชาการนานาชาติ

ผลงานวิจัยทั้งหมดได้รวบรวมเขียนเป็นบทความวิจัยแล้ว ในชื่อ 'Encoding Technique for Dual Level Access to Broadcasting Networks' ขณะนี้กำลังรอให้ที่ปรึกษา (Mentor) ตรวจสอบ แก้ไขเป็นครั้งสุดท้าย ก่อนที่จะส่งไปให้ทางวารสารวิชาการระดับนานาชาติ เพื่อพิจารณาลงตีพิมพ์ใน IEE Proceeding - Communications

2. การนำผลงานวิจัยไปใช้ประโยชน์

ผลงานวิจัยบางส่วนได้ถูกนำไปใช้ประโยชน์ในเชิงวิชาการ เช่น ในการทำวิจัยของนักศึกษาในระดับปริญญาโท นอกจากนี้ความรู้ที่ได้รับจากงานวิจัยชิ้นนี้ยังได้นำไปใช้ในประโยชน์ในการทำงานวิจัยทางด้านเทคโนโลยีการทำภาพพื้นพลาายนํ้าดิจิตอล (Digital Watermarking) อีกด้วย

3. การเสนอผลงานในที่ประชุมวิชาการ

- 3.1 T. Amomraksa and P. Sweeney, 'Improved Encoding Scheme for Transmitting Extra Bits in Pay TV Services', Proceedings of the 2001 International Conference on Information Technology IConIT2001, Bangkok, Thailand, May 28-30, 2001, pp. 153-162.
- 3.2 T. Amomraksa and P. Sweeney, 'A Fast Encryption Technique for Asymmetric Algorithms', Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001), Orlando, USA, July 22-25, 2001, volume VII, pp.177-182.
- 3.3 T. Amomraksa and P. Sweeney, 'Dual Level Access Scheme for Broadcasting Networks', Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001), Orlando, USA, July 22-25, 2001, volume V, pp.7-12.
- 3.4 W. Kanjanarin, P. Supasirsun and T. Amomraksa, 'Access Limited Coding for Digital Video Streams', Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001), Orlando, USA, July 22-25, 2001, volume VI, pp.117-122.

- 3.5 W. Kanjanarin and T. Amornraksa 'Scrambling and Key Distribution Scheme for Digital Television', Proceedings of the 9th IEEE International Conference on Networks ICON2001, Bangkok, Thailand, October 10-12, 2001, pp.140-145.
- 3.6 T. Amornraksa and P. Sweeney, 'Applying Spread Spectrum Technique for Transmitting Extra Bits over AWGN Channel', Proceedings of the 9th IEEE International Conference on Networks ICON2001, Bangkok, Thailand, October 10-12, 2001, pp.390-395.
- 3.7 T. Amornraksa and P. Sweeney, 'Transmitting Extra Bits over DVB Systems', Proceedings of the International Symposium on Communications and Information Technology (ISCIT2001) , ChiangMai, Thailand, November 14-16, 2001, pp.389-392.

4. ผลงานที่ส่งเพื่อพิจารณาลงตีพิมพ์ในวารสารวิชาการในประเทศ

- 4.1 T. Amornraksa and P. Sweeney, 'A Study of Dual Level Access to Digital Video Broadcasting Network', Submitted to Engineering Transactions, Mahanakorn University of Technology.

ภาคผนวก

1. บทความสำหรับการเผยแพร่
2. บทความที่ได้รับการตีพิมพ์ในการประชุมวิชาการระดับนานาชาติ
3. รายงานการเงิน (แนบมาพร้อมกับรายงานฉบับนี้)

Title : Encoding Technique for Dual Level Access to Broadcasting Networks

Authors : Thumrongrat Amornraksa¹ and Peter Sweeney²

Address : ¹Computer Engineering Department, KMUTT, Pracha-Utid, Bangkok 10140, Thailand

²Centre for Communication Systems Research, University of Surrey, Guildford, GU2 5XH, UK.

Objective: To develop an encoding scheme that gives two levels of access to the received signal in the broadcasting networks. Moreover, the implementation of the scheme with the digital video broadcasting (DVB) systems in AWGN channel is considered and investigated.

Method: An approach based on direct sequence spread spectrum technique was used to construct an encoding scheme for adding the extra information bits into the main data, without requiring additional bandwidth for broadcasting the resultant signal. In the decoding process, the extra bits are first recovered from the received signal, and later used to recover the main data. Hence, the received signal can be decoded into two different sets of data, giving two different levels of access.

Result: From the simulation results, after applying the developed encoding scheme with the DVB-T system in AWGN channel, its performance was measured and illustrated below.

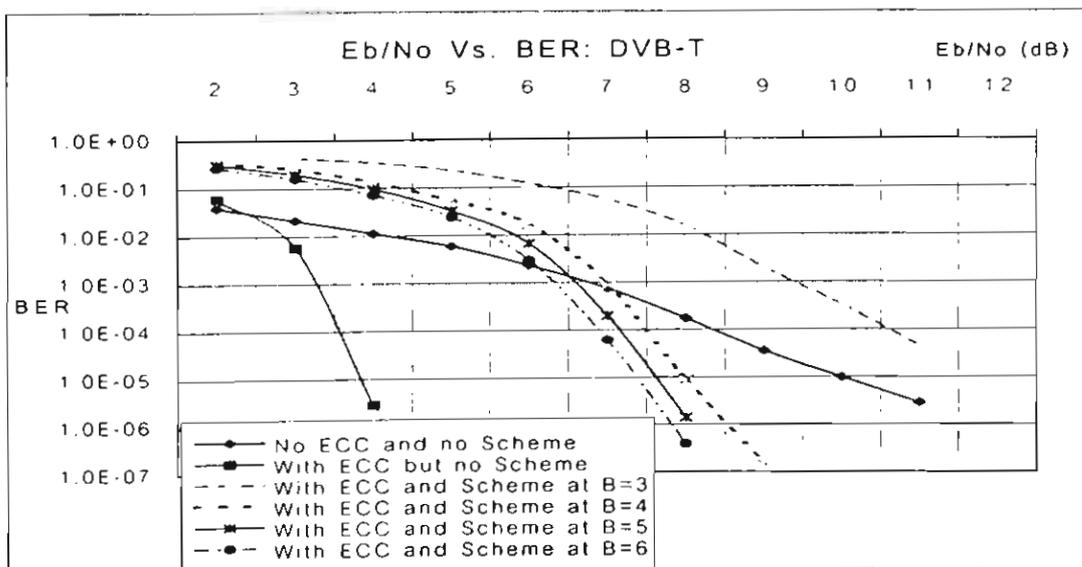


Figure 1. Performance of the encoding scheme in the DVB-T systems at different block sizes

Conclusion: Up to 0.627 % of the main data can be transmitted without requiring additional bandwidth. Since there were some errors up to $1/(2^k)$ % remaining in the main data, its quality was degraded. However, the results showed that the PSNR obtained from the resultant images, which were used as main data, were approximately 50-54 dB, depending on the block size used. Nevertheless, the human visual system i.e. human eye could not notice any visual degradation.

Keyword: Dual level access scheme, Direct sequence spread spectrum, Digital signal processing

บทความที่ได้รับการตีพิมพ์ในการประชุมวิชาการระดับนานาชาติ

1. T. Amornraksa and P. Sweeney, 'Improved Encoding Scheme for Transmitting Extra Bits in Pay TV Services', Proceedings of the 2001 International Conference on Information Technology IConIT'2001, Bangkok, Thailand, May 28-30, 2001, pp. 153-162.
2. T. Amornraksa and P. Sweeney, 'A Fast Encryption Technique for Asymmetric Algorithms', Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001), Orlando, USA, July 22-25, 2001, volume VII, pp.177-182.
3. T. Amornraksa and P. Sweeney, 'Dual Level Access Scheme for Broadcasting Networks', Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001), Orlando, USA, July 22-25, 2001, volume V, pp.7-12.
4. W. Kanjanarin, P. Supasirisun and T. Amornraksa, 'Access Limited Coding for Digital Video Streams' Proceedings of the 5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001), Orlando, USA, July 22-25, 2001, volume VI, pp.117-122.
5. W. Kanjanarin and T. Amornraksa 'Scrambling and Key Distribution Scheme for Digital Television', Proceedings of the 9th IEEE International Conference on Networks ICON2001, Bangkok, Thailand, October 10-12, 2001, pp.140-145.
6. T. Amornraksa and P. Sweeney, 'Applying Spread Spectrum Technique for Transmitting Extra Bits over AWGN Channel', Proceedings of the 9th IEEE International Conference on Networks ICON2001, Bangkok, Thailand, October 10-12, 2001, pp.390-395.
7. T. Amornraksa and P. Sweeney, 'Transmitting Extra Bits over DVB Systems', Proceedings of the International Symposium on Communications and Information Technology (ISCIT2001), ChiangMai, Thailand, November 14-16, 2001, pp.389-392.

IMPROVED ENCODING SCHEME FOR TRANSMITTING EXTRA BITS IN PAY TV SERVICES

Thumrongrat AMORNRAKSA
Multimedia Communications Laboratory, Department of Computer Engineering,
Faculty of Engineering, King Mongkut's University of Technology Thonburi,
Bangkok 10140, Thailand. Email: 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
Email: p.sweeney@eim.surrey.ac.uk

ABSTRACT

An encoding scheme based on direct sequence spread spectrum technique, for conveying some extra bits in pay-TV services, with no increase in bandwidth is described in this paper. By adding small information into the encrypted video signal to generate the output signal for transmission, any subscriber is allowed to view this information, while only the subscribers with the decryption key can view the encrypted content. The scheme's efficiency was improved by applying error control codes to the information bits before the encoding process. The scheme was examined by simulation method and its performances with various error control codes were measured. The improved scheme was tested for the practical use purpose by transmitting the encoded MPEG stream through an AWGN channel to observe its performance.

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 many 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, as referred to dual level access scheme (Amornraksa et al. 1999). A technique based on direct-sequence spread spectrum communications 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. With this scheme, the allocated bandwidth for broadcasting is utilised more efficiently, and then 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 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 and types of data used to evaluate the performance of the scheme. The results from simulations and discussions are then given in Section 4. Finally, the last Section provides some concluding remarks.

BACKGROUND

In spread spectrum communications (Pickholtz et al. 1982, Lee 1989), 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 spread spectrum systems is a pseudo-random binary sequence (PRBS). For these direct sequence spread spectrum systems, the original baseband signal is multiplied by the PRBS to produce a new bit stream. Only those receivers equipped with the correct PRBS can decode the original baseband signal. 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 (Dixon 1994).

Apart from applications in wireless communications, the direct sequence spread spectrum technique is widely used in digital watermarking applications such as in (Cox et al. 1997, Hartung et al. 1998, George et al. 1999). By spreading the information bits and modulating them with a PRBS, the spread signal can be obtained. This signal is then embedded in the video signal, below the threshold of perception. The recovery of the embedded signal can be accomplished by correlating the modified video signal with the same PRBS that was used in the process of constructing the spread spectrum signal. Correlation here is demodulation followed by summation over the width of the chip-rate, that is, the number of blocks over which each information bit is spread.

Based on this concept, an encoding scheme for transmitting extra bits via a broadcast channel can be constructed. That is, some extra bits will be added into the encrypted video signal before the transmission process takes place, so that at the receiver end, they will be first extracted from the received signal and then used to recover the encrypted signal. These extra bits can be utilized in some ways such as transmitting some advertisements. Beside the benefit to the other subscribers, 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.

DESCRIPTION OF THE SCHEME

In digital communication systems, channel coding is normally applied to the modulated signal before being transmitted, and this signal is considered as the encrypted video signal in our encoding scheme. Using a similar technique to that proposed in (Hartung et al. 1998), the extra bits will be added to the encrypted signal after the channel coding process to obtain the resultant signal for transmission. Given a key to reproduce the same PRBS at the receiver end, the extra bits can be recovered. The

encrypted 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 may be detected and corrected by the channel decoder. The operation of the encoding scheme is shown in figure below.

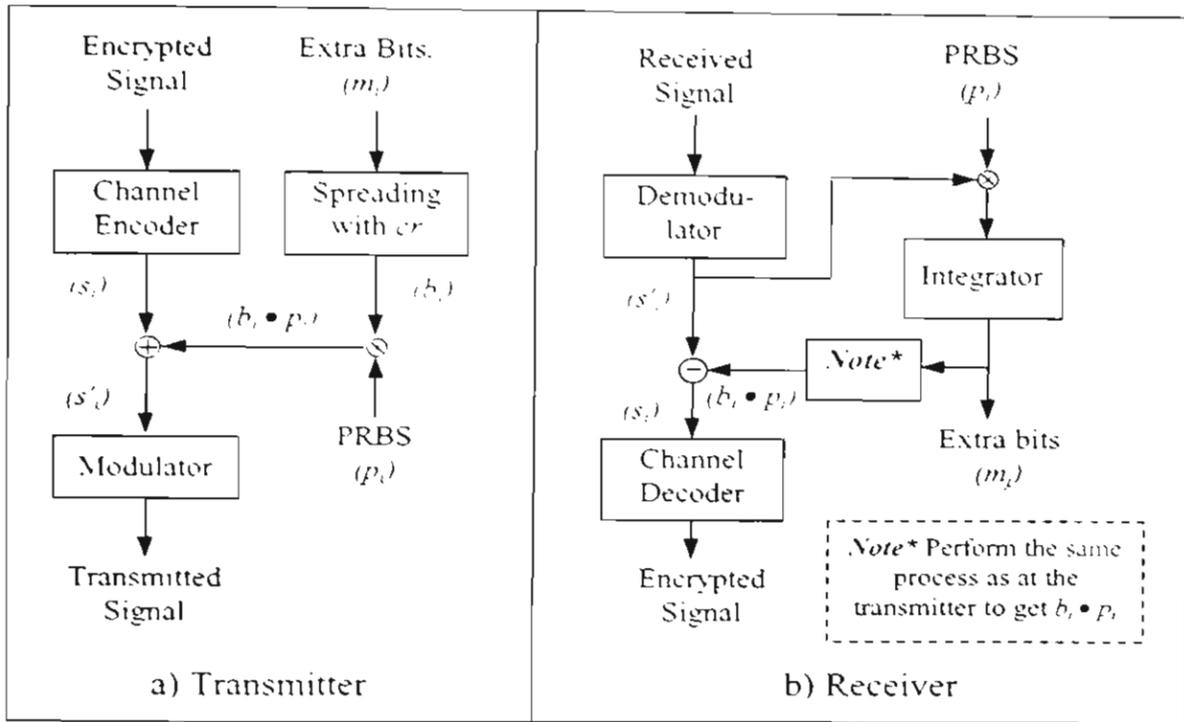


Figure 1. The operation of the Encoding Scheme

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) , $b_i = m_j$, $j \cdot cr \leq i < (j+1) \cdot cr$. 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) , $s'_i = s_i + p_i \cdot b_i$.

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:

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

The first term on the right-hand side of (1) vanishes if p_i and s_i are uncorrelated, and then $\sum_{i=j \cdot cr}^{(j+1) \cdot cr - 1} p_i = 0$ (Stark et al. 1986). Since $p_i^2 = 1$, r_j ideally becomes

$$r'_j \approx cr \cdot m_j \quad (2)$$

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

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.

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:

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

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 their 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 (Sweeney 1991, Lin et al. 1983), 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 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, the data used as extra bits in the experiments was pseudo-randomly generated by a function $rand()$ in C language, while the MPEG encoded stream was used to carry the extra bits. Both types of data will then be input into the constructed encoding scheme as described in the Figure 1.

After the proper code that gives the best performance is found, the scheme will be simulated in a communication channel. The generated Additive White Gaussian Noise (AWGN) channel will be added to the transmitted signal since it is a type of noise that most communication systems encounter (Michelson et al. 1985). This time, the channel encoder is also applied to the encrypted signal in order to observe the performance of

the entire system. The results from the simulation will be shown by a plot of the BER versus the E_b/N_0 . The simulation model used in the experiments is shown in figure below.

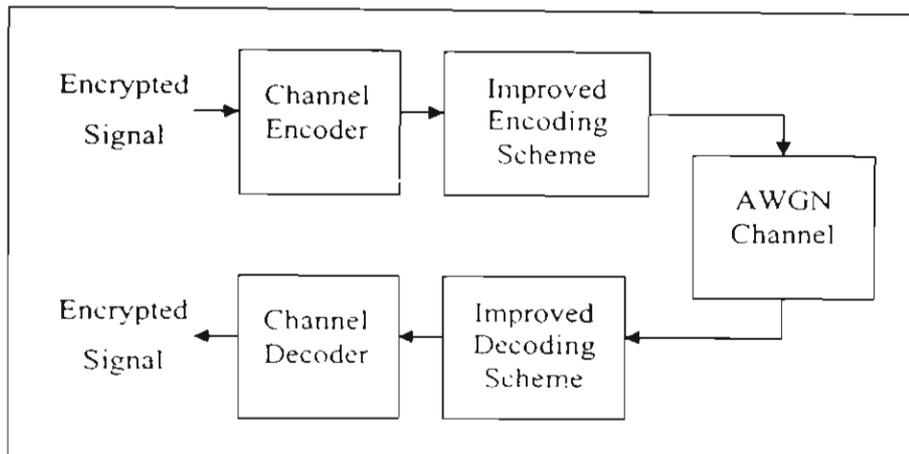


Figure 2. Simulation model in an AWGN channel

SIMULATION 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 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.

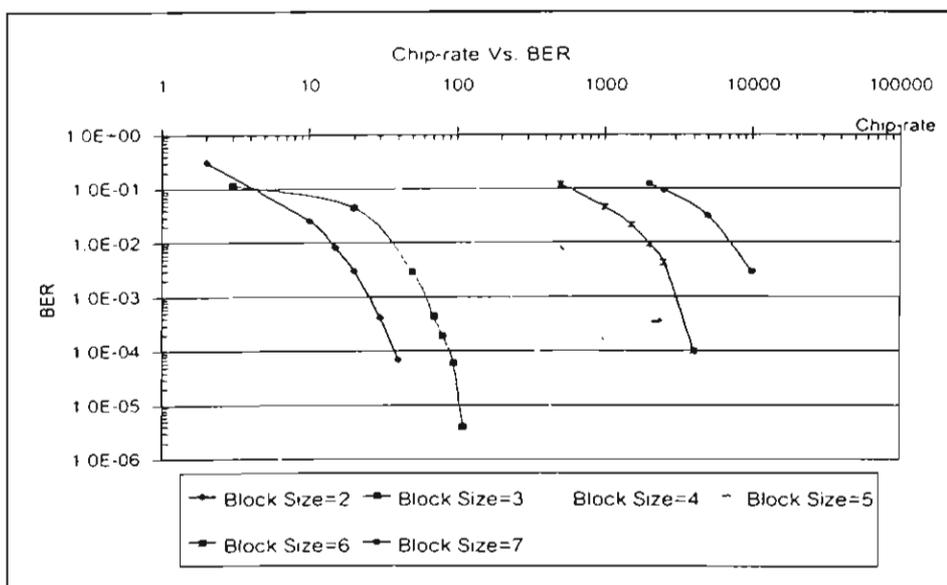


Figure 3. Bit error rate of extracted extra bits at different block sizes

From the figure, 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. 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 figure 4, 5, 6 and 7 for the block sizes of 3, 4, 5, 6 respectively.

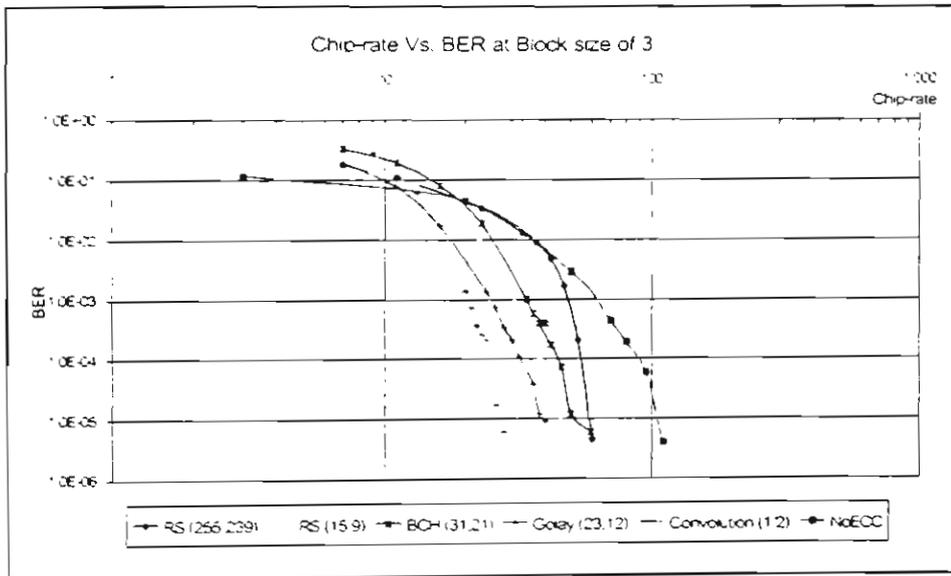


Figure 4. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 3

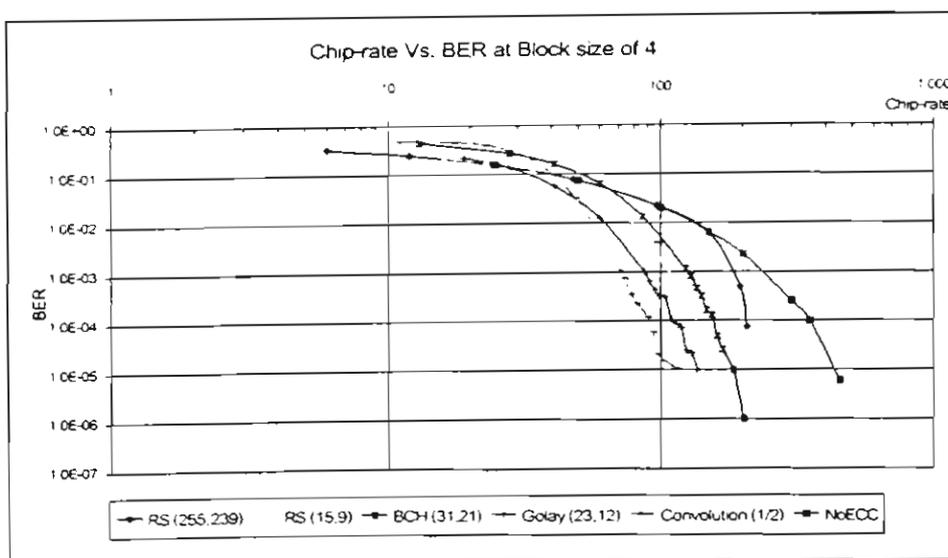


Figure 5. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 4

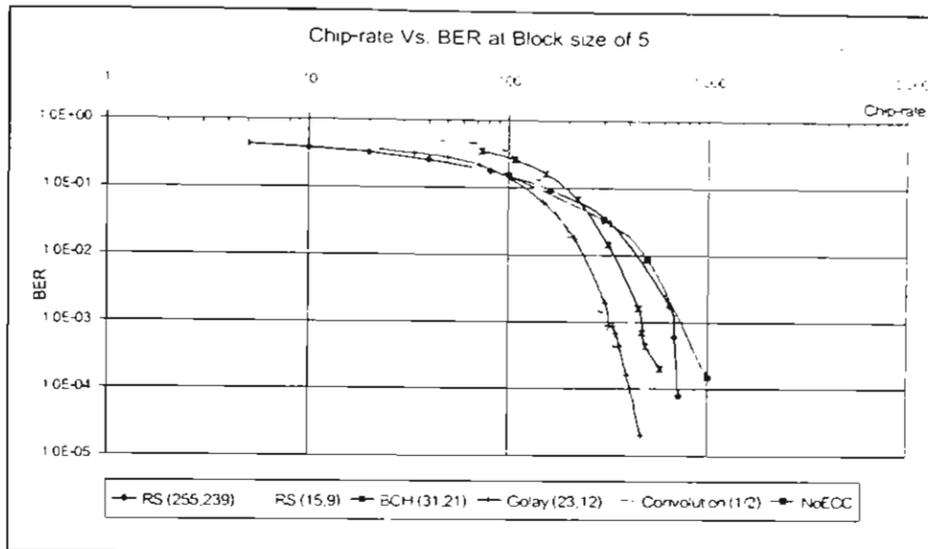


Figure 6. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 5

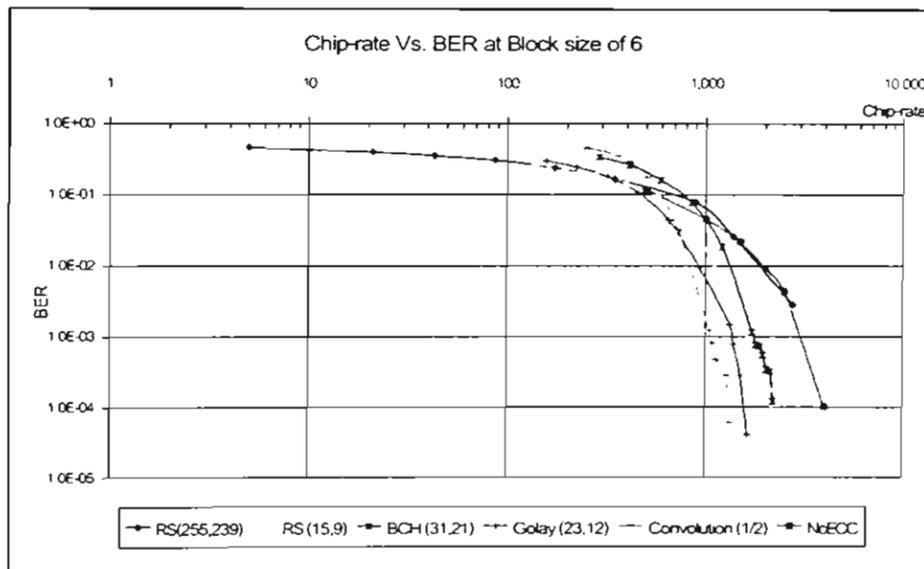


Figure 7. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 6

It obviously shows, from the figures above, 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 should be the first choice to be used in practice. To illustrate the benefits of using error control codes, the number of encrypted data (symbol) which is used to convey the extra bits is plotted against the BER. Figure 8 shows the performance comparison of the encoding scheme with and without the error control codes at the block size of 4.

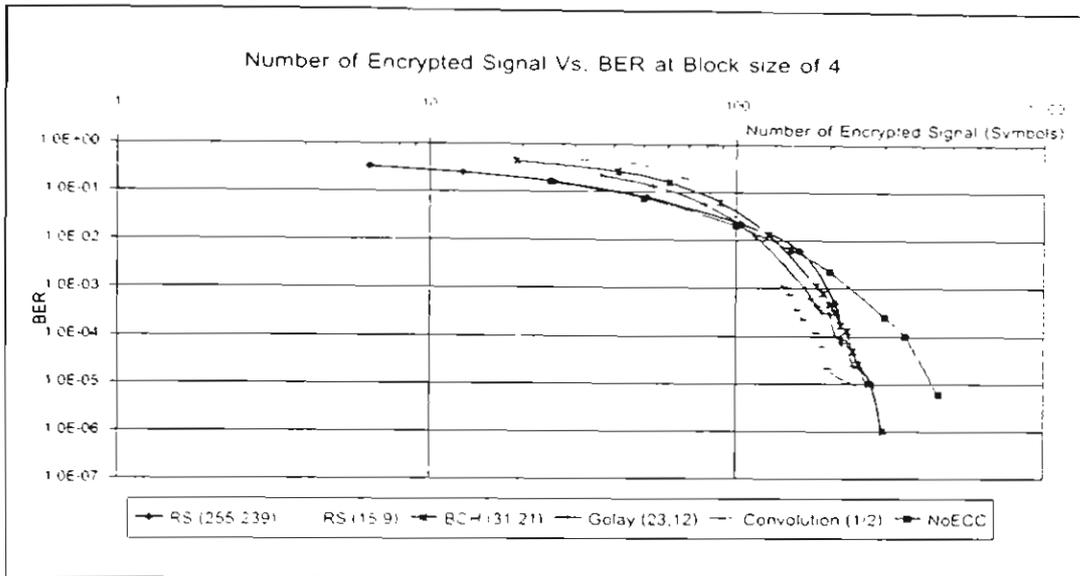


Figure 8. Performance comparison between the scheme with and without the error control codes at the block sizes of 4

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 figure 9 and 10, the plots of the BER versus the E_b/N_0 for the scheme using binary BCH (31, 21) code and the convolutional code with rate 1/2 are given.

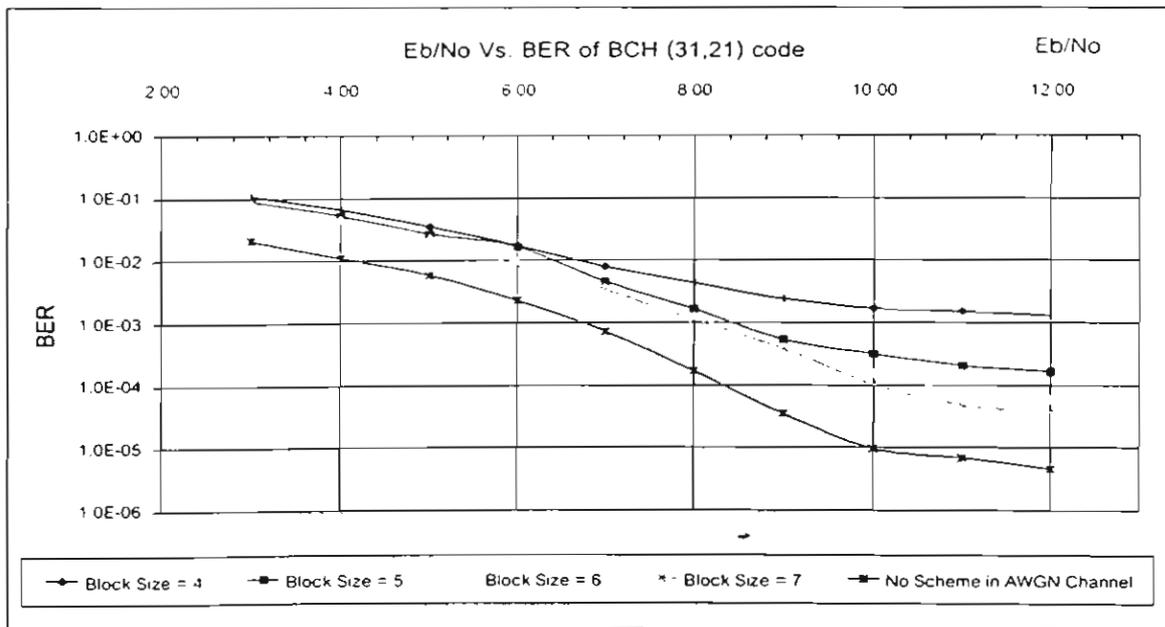


Figure 9. Performance comparison when the scheme is implemented with the binary BCH (31, 21) code in the AWGN channel at various block sizes

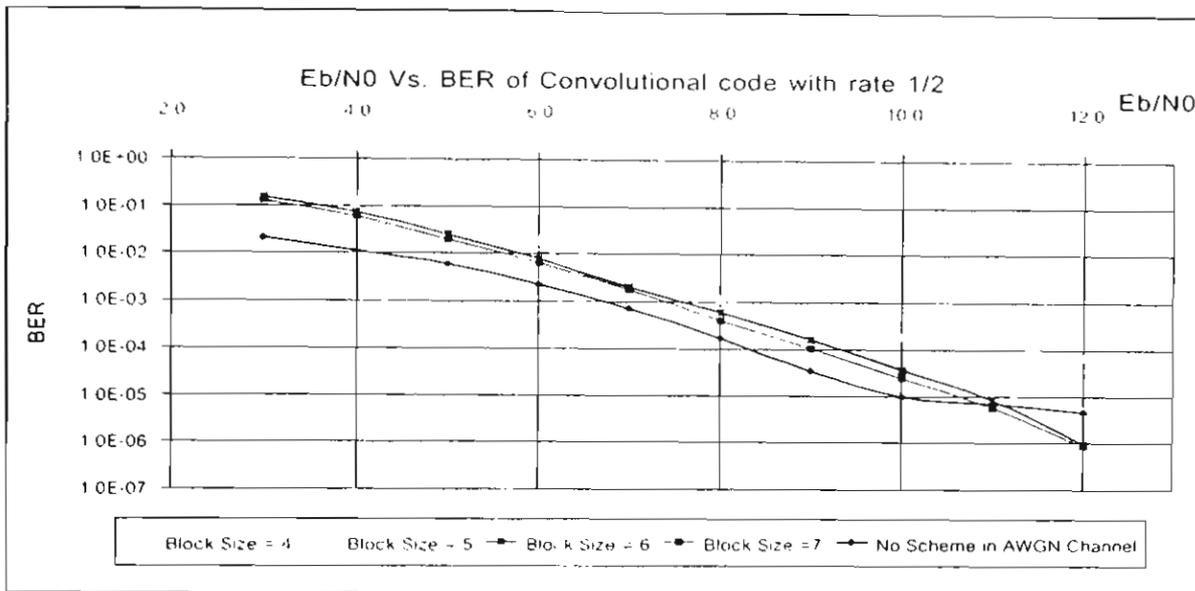


Figure 10. Performance comparison when the scheme is implemented with the convolutional code with rate $\frac{1}{2}$ in the AWGN channel at various block sizes

From both figures, it can be seen that errors occurred at the output of the original signal were larger than the ones occurred 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.

CONCLUSIONS

In this paper we have shown a method of constructing an encoding scheme based on the direct sequence spread spectrum technique for transmitting the extra bits in pay TV services. We have also shown experimentally that the scheme's performance was improved by applying the error control codes to the information bits before the encoding process. Although its performance was not quite impressed when implementing in the AWGN channel, our approach has showed an idea of how to utilize the existing allocated bandwidth in a more efficient way. Moreover, in practice the concatenated codes will be used as a channel encoder in broadcasting service such as Digital Video Broadcasting (DVB), which has much better performance than the stand-alone one used in our experiments. Finally, further work can be carried out by focusing on DVB applications and different kinds of noise occurred in communication systems.

ACKNOWLEDGMENT

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

REFERENCES

- Amornraksa T, Burgess D, Sweeney P (1999) An encoding scheme for dual level access to broadcasting networks, Proceeding of the Seventh IMA International Conference on Cryptography and Coding, Cirencester, UK, December, pp. 114-118.
- Cox I, Kilian J, Leighton T, Shamoon T (1997) Secure spread spectrum watermarking for multimedia, IEEE Transactions on Image Processing, vol. 6, No. 12, December, pp. 1673-1687.
- Dixon R C (1994) Spread Spectrum Systems with Commercial Applications 3rd Edition", John Wiley & Son Inc., New York.
- George M, Chouinard J V, Georganas N (1999) 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, pp. 116-121.
- Hartung F, Girod B (1998) Watermarking of uncompressed and compressed video, Signal Processing, vol. 66, no. 3 (Special issue on Watermarking), pp. 283-301.
- Lee W C Y (1989) Spectrum Efficiency in Cellular, IEEE Transactions on Vehicular Technology, vol. 38, No. 2, May, pp. 69-75.
- Lin S, Costello D J Jr (1983) Error Control Coding: Fundamentals and Applications, Prentice-Hall, Englewood Cliffs, N. J.
- Michelson A M, Levesque A H (1985). Error Control Techniques for Digital Communication", New York: John Wiley & Sons.
- Pickholtz R, Schilling D, Millstein L (1982) Theory of spread spectrum communications a tutorial, IEEE Transaction on Communication, vol. COMM-30, pp. 855-884.
- Stark H, Woods J W (1986) Probability, Random Variables and Estimation Theory for Engineers, Prentice Hall, Englewood Cliffs, N. J.
- Sweeney P (1991) Error Control Coding: An Introduction, Prentice Hall, London.

A Fast Encryption Technique for Asymmetric Algorithms

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

In this paper, we consider the encryption of large amounts of data using an asymmetric algorithm. A new encryption method to reduce the complexity in the encryption process that maintains the security level of the entire system is presented. A discrete linear transform is used to prepare the plaintext before being encrypted. Only a small portion of the data is to be encrypted and the encrypted data is then used to protect the remaining data. This method can significantly reduce encryption time and is suited for applications requiring a high encryption rate.

Keywords: Cryptography, Encryption Method, Asymmetric Algorithms, Discrete Linear Transform, Security.

1. INTRODUCTION

The advantage of asymmetric algorithms over symmetric algorithms is that one of the key pair can be made public so that anyone can use that public key to encrypt or to authenticate a message. However, since the security of asymmetric algorithms depends on the difficulty of mathematical problems, most of the algorithms require complex computations, typically involving raising a large number to a large power. The RSA (Rivest-Shamir-Adelman) algorithm [1] is an example. It is generally 100 to 1000 times slower than block symmetric algorithms such as DES (Data Encryption Standard) [2, 3]. This drawback sometimes makes asymmetric algorithms impractical for encrypting large data sets such as a database, or applications with a high bit rate such as digital TV services.

Conventionally, asymmetric algorithms are used to encrypt the secret key of a symmetric algorithm and that symmetric algorithm with the secret key is then used to encrypt the data. This is known as the principle of hybrid cryptosystems. Besides hybrid method, there are many encryption schemes for speeding up the encryption process, such as those proposed in [4, 5, 6]. In [4] the authors proposed a new scheme for block encryption, allowing the en/decryption of arbitrarily long messages,

but performing en/decryption on only a single block, where the rest of the message is only processed by a good scrambling function, while [5, 6] suggested a design where scrambling via hashing is done prior to partial encryption as in [4]. Anyhow, our approach is different from them since we employ asymmetric algorithms directly in data encryption, in a more efficient way. Instead of encrypting the entire data, we pseudo-randomly extract a small portion of that data to be encrypted. The encrypted portion of the data is then used to protect the remaining data. To obtain the original data, one needs to decrypt this encrypted part. That is, the security of the system depends only on the encrypted part.

Another interesting technique is based on the All-Or-Nothing Transform (AONT) [7, 8], where the AONT is applied to the blocks of message before performing the encryption. An adversary now cannot find out any information about any block of the message without decrypting all the blocks of the ciphertext. To make it more efficient, instead of encrypting the whole message block by block, we can apply AONT to it, and encrypt just some of the output blocks. This technique is quite similar to ours, except we use the ciphertext of the transformed extracted information to create the final ciphertext via an adding operation.

In this paper, we consider the case where a large amount of data is required for encryption by an asymmetric algorithm. Based on the approach previously described, an encryption method, in which block encryption algorithms may be applied, is proposed to reduce the complexity in the encryption process, while maintaining the security level of the entire system.

2. BASIC APPROACH

An encryption method is designed for encrypting a digital bit stream consisting of a plurality of data blocks. Instead of encrypting the whole bit stream, in the proposed method, a small portion of data is first pseudorandomly extracted from that bit stream to be encrypted and that encrypted part is then used in

some way to protect the original bit stream. To obtain the original data, one needs to decrypt this encrypted part. By doing this, the security of the system relies on the encrypted part only. The basic approach of the above idea can be described as follows:

The digital bit stream is broken into successive data blocks of the same length. Each data block is then transformed using a discrete linear transform to generate a respective set of transformation coefficients. One of the transformation coefficients in each set is selected pseudorandomly and the selected coefficients are formed into a block of plaintext which is encrypted using a block cipher to create a block of ciphertext. The blocks of plaintext and ciphertext are differenced to form a set of difference values, each corresponding to a respective data block. Each difference value is then inverse-transformed using a corresponding inverse transform to generate a set of inverse-transformation values which are added to the data values in the respective data block to generate a corresponding ciphertext. The block diagram of the basic approach is shown in Figure 1.

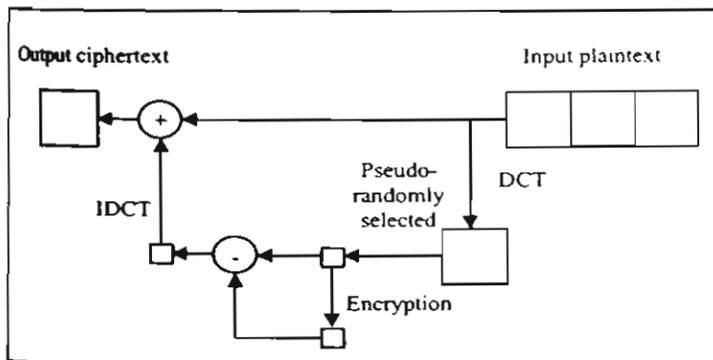


Figure 1. Block Diagram of the Basic Approach

There are many discrete linear transforms currently used in digital signal processing, and one of them is chosen for use as an example in the encryption scheme, namely the discrete cosine transform (DCT). The discrete cosine transform [9] is widely used in image and speech transform coding due to the availability of fast DCT algorithms. We chose DCT for presenting our algorithm for this particular reason.

2.1. Discrete Cosine Transform (DCT)

In general, the DCT is a particular case of the Fourier transform applied to discrete (sampled) signals. It decomposes the signal into only one series of harmonic cosine functions in phase with the signal, which reduces by half the number of coefficients necessary to describe the signal compared to the Fourier transform. Note that the Fourier transform decomposes a periodic signal into a series of sine and cosine harmonic functions.

In the subject of video coding, the image is separated into blocks typically 8×8 or 16×16 pixels, which the DCT transforms into a matrix of 8×8 or 16×16 coefficients respectively. (Each pixel contains 8 bit information which represents the values between 0 to 256). In order to reduce the complexity of the computation and the processing time required in transformation process, the block size chosen is generally 8×8 pixels. This block size is also optimum for trade-off between compression efficiency and computational complexity [10]. Although a larger block size leads to more efficient coding, it requires more computational power.

After arranging the input data into blocks of 8×8 pixels, the two-dimensional (2-D) DCT is then applied to each block of them to obtain the transformation coefficients. In the transformed block, coefficients on the horizontal axis represent increasing horizontal frequencies from left to right, and on the vertical axis they represent increasing vertical frequencies from top to bottom. The first coefficient in the top left corner (co-ordinates: 0, 0) represents null horizontal and vertical frequencies, and the bottom right coefficient represents the highest spatial frequency component in the two directions. The coefficient at the top left corner is called the *DC coefficient*, whereas the rest of the coefficients are called *AC coefficients*. The basis functions of the DCT is illustrated in Figure 2.

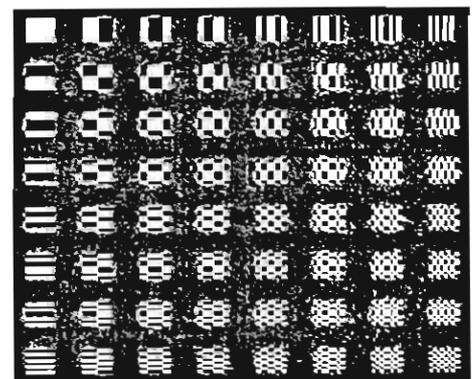


Figure 2. DCT Basis Functions

To reduce the number of calculations required to perform the 2-D transform, a one-dimensional transform can be used instead. That is, the same result can be achieved by applying a 1-D transform along all the rows of the block and then down the columns of the block. The computational complexity can be further reduced by replacing the cosine form of the transforms with a fast algorithm [11] which reduces the operation to a short series of multiplications and additions. The 1-D FDCT and IDCT are described by the following formula.

$$\text{FDCT: } S(u) = \frac{C(u)}{2} \sum_{x=0}^7 s(x) \cos \left[\frac{\pi(2x+1)u}{16} \right] \quad (1)$$

$$\text{IDCT: } s(x) = \sum_{u=0}^7 \frac{C(u)}{2} S(u) \cos \left[\frac{\pi(2x+1)u}{16} \right] \quad (2)$$

with $u, x = 0, 1, 2, \dots, 7$

where x is the spatial co-ordinate in the pixel domain (sample value)

u is the co-ordinate in the transform domain (DCT coefficient)

$$C(u) = \frac{1}{\sqrt{2}} \text{ for } u = 0, \text{ otherwise } 1$$

The 2-D FDCT and IDCT can then be constructed from products of the terms of a horizontal 1-D DCT (using u and x) and a vertical 1-D DCT (using v and y), where v represents vertical frequencies and y represents vertical displacements [9].

3. DESCRIPTION OF THE SCHEME

According to the encryption method previously described, the encryption scheme can be divided into four main steps. That is, transforming the data blocks of the digital bit stream, selecting a part of each transformed block and combining the selected parts to form a block of plaintext, encrypting that block of plaintext, and modifying the data blocks using the difference between blocks of plaintext and ciphertext whereby to encrypt the digital bit stream. However, in practice, steps one and two can be combined into one operation. This is done by calculating only one DCT coefficient at the particular position chosen pseudorandomly.

The encryption scheme is now described by way of example. The asymmetric algorithm used in the encryption process is the 1024-bit RSA algorithm, whereas the 8×8 DCT is used in the transformation process. A pseudorandom number generator is also used in the scheme. It generates a sequence of numbers ranging from 2 to the maximum size of transformation block. Each successive number in the sequence determines the position of the selected DCT coefficient for each successive block in the digital bit stream. It will be appreciated that the first DCT coefficient represents the DC component in the frequency domain and is ignored in the selection process. The block diagram of the encryption scheme is shown in Figure 3.

The flow diagram of the encryption scheme in more detail is shown in Figure 4 and its operation is described as follows.

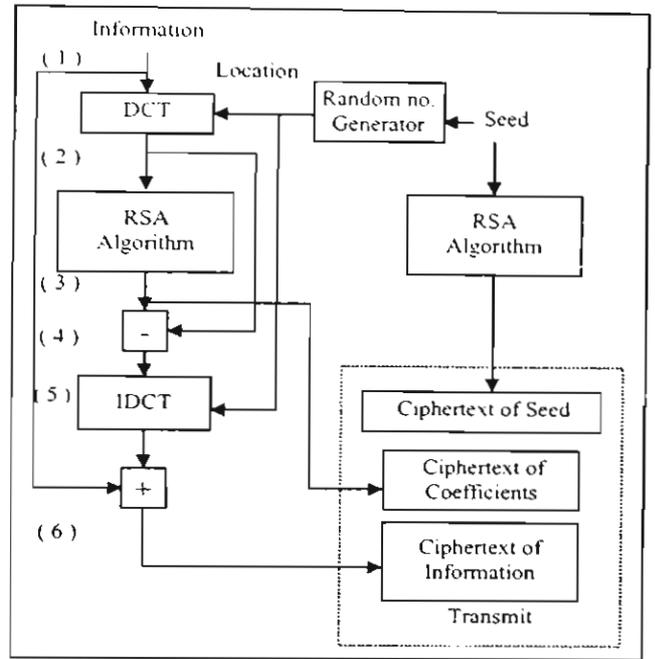


Figure 3. Block Diagram of the Encryption Scheme

- (1) Data block in time domain
- (2) DCT data blocks into frequency domain
- (3) Encryption process using RSA
- (4) Blocks of plaintext and ciphertext are differenced
- (5) IDCT the difference into time domain
- (6) The difference is added to data block to form the final ciphertext

1. The input digital bit stream is broken into successive blocks of the same length. The ending block is padded to the same length. In this example, the scheme processes a digital bit stream consisting of 128 successive data blocks block B_1, B_2, \dots, B_{128} at a time. Each block contains 64 bytes and each byte represents an 8-bit data value ranging from 0 to 255.
2. The DCT transforms the 64 data values $s(x)$ in the respective block from the time domain to the frequency domain thereby generating a corresponding set of 64 DCT coefficients $S(u)$ for the block.
3. Only one of the 64 DCT coefficients derived from each block is selected, and the selected coefficients for all the blocks are grouped together to form a block of plaintext consisting of 128 selected DCT coefficients $S_1(u_1), S_2(u_2), \dots, S_{128}(u_{128})$. (See step 1 in Figure 4.4). The position u of a selected DCT coefficient within the respective transform is determined pseudorandomly by means of a pseudorandom number generator operating in accordance with a pseudorandom seed.
4. The block of plaintext (1024 bits) is encrypted using the RSA algorithm to form a corresponding block of ciphertext (1024 bits) (step 2).

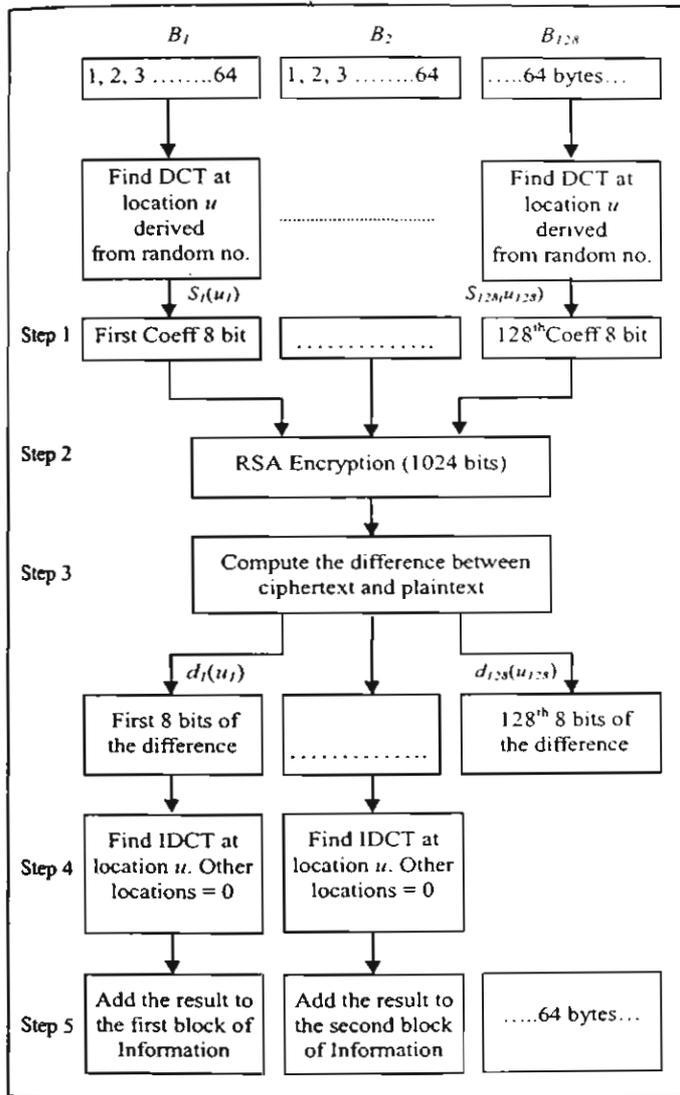


Figure 4. Flow Diagram of the Encryption Scheme in More Detail

- The blocks of plaintext and ciphertext are differenced, and the resultant difference is de-grouped to form a set of 8-bit difference values $d_1(u_1), d_2(u_2), \dots, d_{128}(u_{128})$, each corresponding to a respective data block B_1, B_2, \dots, B_{128} (step 3).
- The IDCT transforms each difference value $d(u)$ from the frequency domain to the time domain (all the other coefficients $S(u)$ in the IDCT are set at zero), and generates a respective set of inverse-transformed value $s(x)$ (step 4). Note that the position of each difference value $d(u)$ in the respective IDCT is the same as the position of the selected DCT coefficient $S(u)$ in the corresponding DCT and, to that end, the IDCT is supplied with the same sequence of numbers generated by the pseudorandom number generator.
- The inverse-transformed values $s(x)$ in each set are added to the respective data values in the corresponding data block using modular arithmetic.

In this manner, the encryption method adds a random value to the original data values to generate a corresponding ciphertext (step 5).

The pseudorandom seed is also encrypted using the RSA algorithm to generate a ciphertext of the seed. The ciphertext of the seed, the ciphertext of the selected DCT coefficients and the ciphertext of the original data values are then combined to create a sent-out message. Figure 5 illustrates the frames structure of the sent-out message.

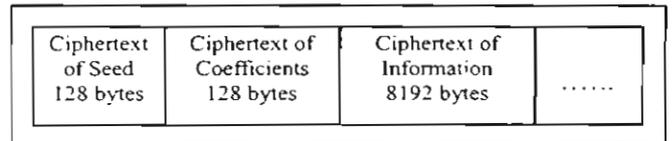


Figure 5. Basic Frame Structure of the Sent-out Message

To recover the original bit stream, the process is simple and straightforward. Figure 6 illustrates the block diagram of the decryption scheme used to decrypt a digital bit stream which has been encrypted using the encryption scheme of Figure 3.

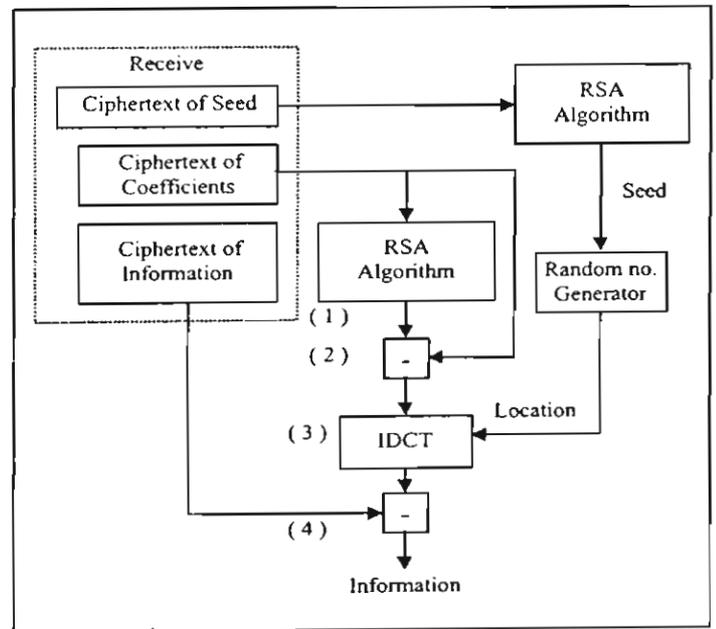


Figure 6. Block Diagram of Decryption Scheme

- Decryption process using RSA
- Blocks of plaintext and ciphertext are differenced
- IDCT the difference into time domain
- The difference is subtracted from data block to recover the original information

The flow diagram of the decryption scheme in more detail is shown in Figure 7 and its operation is described as follows.

- Initially, the ciphertext generated by encrypting the selected DCT coefficients which is part of the sent-out message is decrypted using the RSA algorithm.

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.

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, \dots, 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.

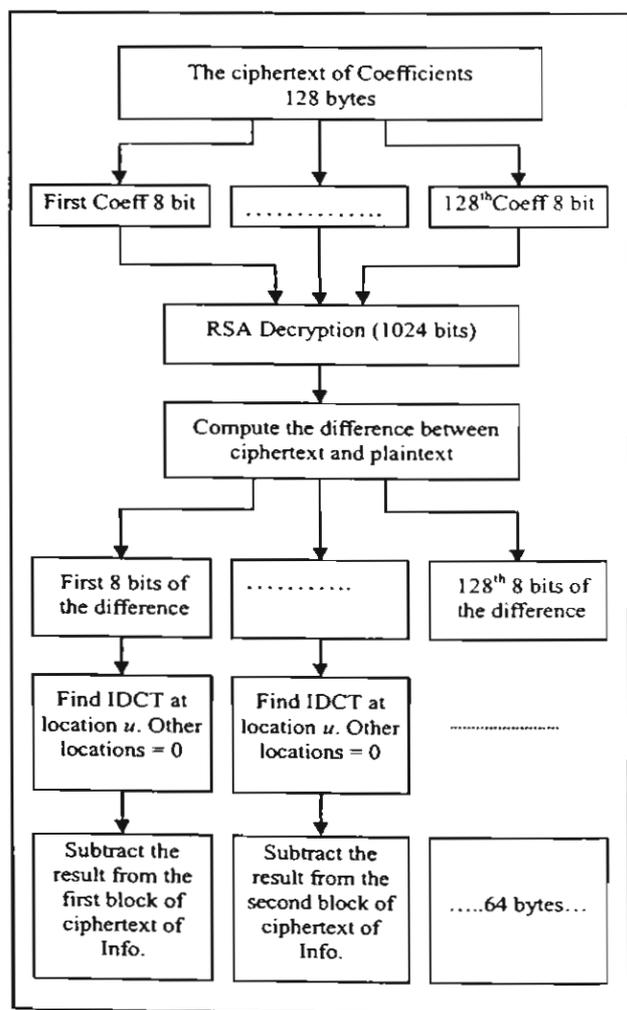


Figure 7. Flow Diagram of the Decryption Scheme in More Detail

5. SECURITY ANALYSIS

In the given example, the RSA algorithm is used in the encryption/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 encryption/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. Rivest, 'All-or-nothing Encryption and the Package Transform', the 4th International Workshop 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. Fraiick, '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 direct sequence spread spectrum technique is widely used in digital watermarking applications such as in [4, 5, 6]. By spreading the information bits and modulating them with a PRBS, the spread signal can be obtained. This signal is then embedded in the video signal, below the threshold of perception. The recovery of the embedded signal can be accomplished by correlating the modified video signal with the same PRBS that was used in the process of constructing the spread spectrum signal. Correlation here is demodulation followed by summation over 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 subscribers, 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) , $b_i = m_j$, $j \cdot cr \leq i < (j+1) \cdot cr$. 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) , $s'_i = s_i + p_i \cdot b_i$.

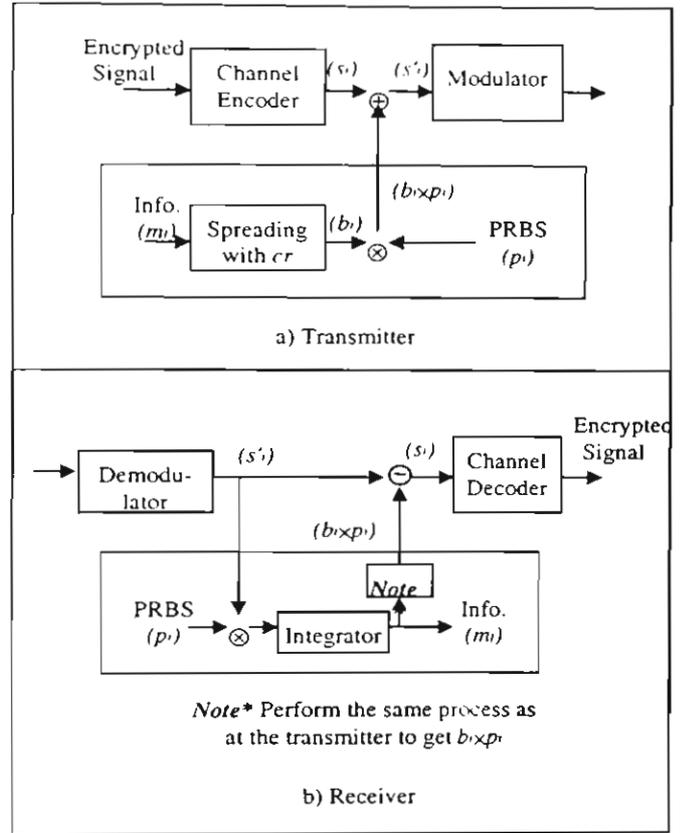


Figure 1. The operation of the Encoding Scheme

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:

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

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

$$r'_j = cr \cdot m_j \quad (2)$$

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

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;

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

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 their 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.

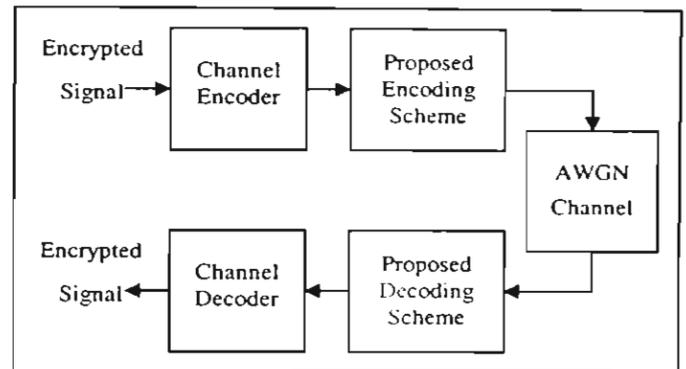


Figure 2. Simulation model in an AWGN channel

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

As Table 1 shows, the smaller the block size, the larger the value of 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.

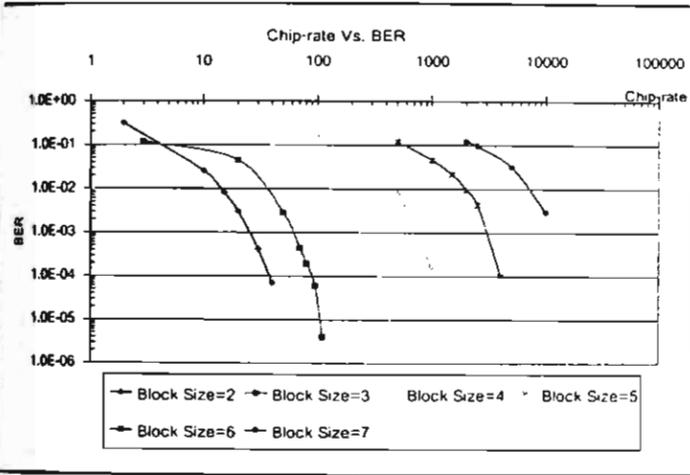


Figure 3. Bit error rate of extracted extra bits at different block sizes

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.

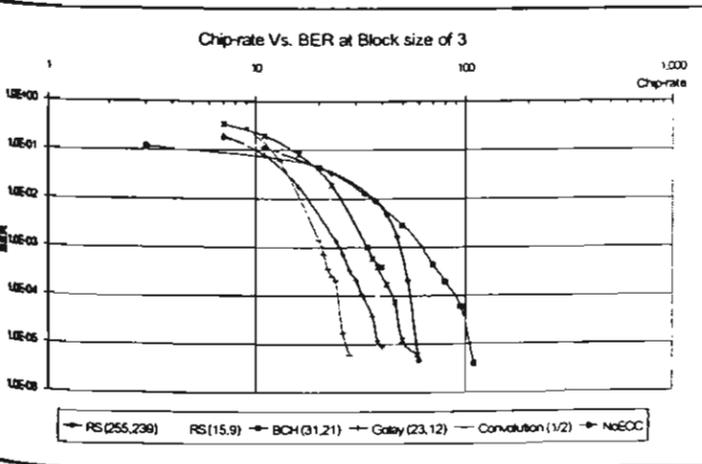


Figure 4. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 3

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.

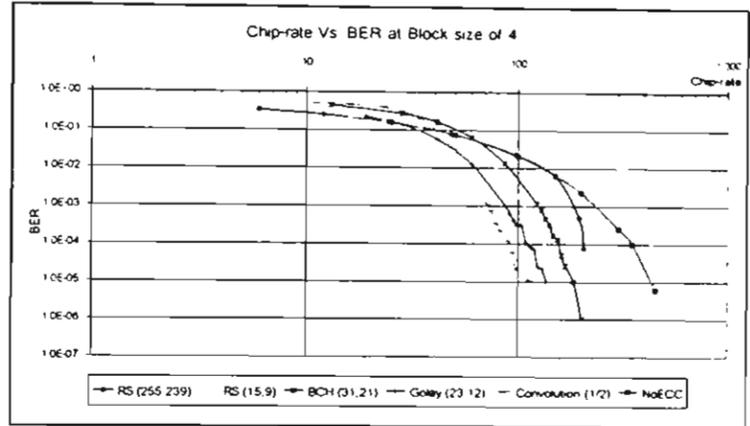


Figure 5. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 4

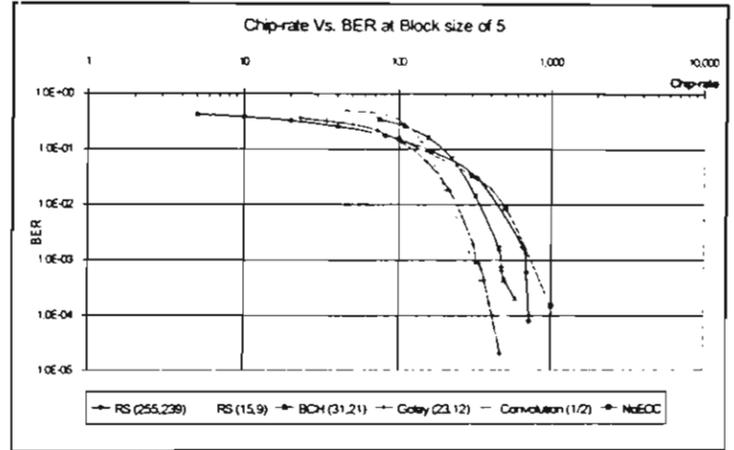


Figure 6. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 5

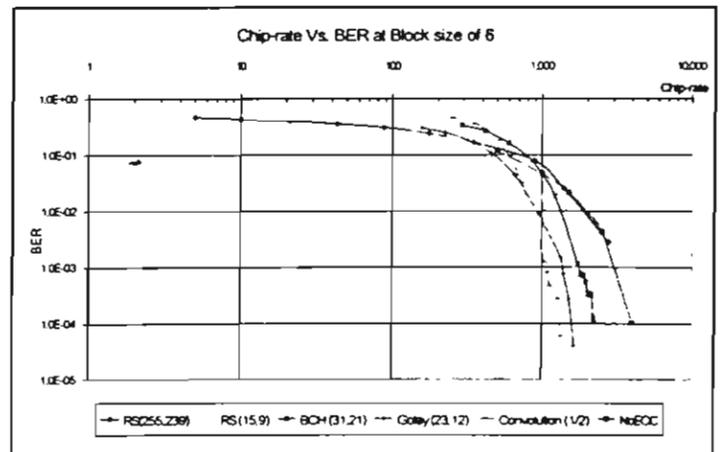


Figure 7. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 6

obviously shows, from the figures above, that the error 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 encrypted data symbol (which is used to convey the extra bits) is plotted against the BER. Figure 8 shows the performance comparison of the encoding scheme with and without the error control codes at the block size of 4.

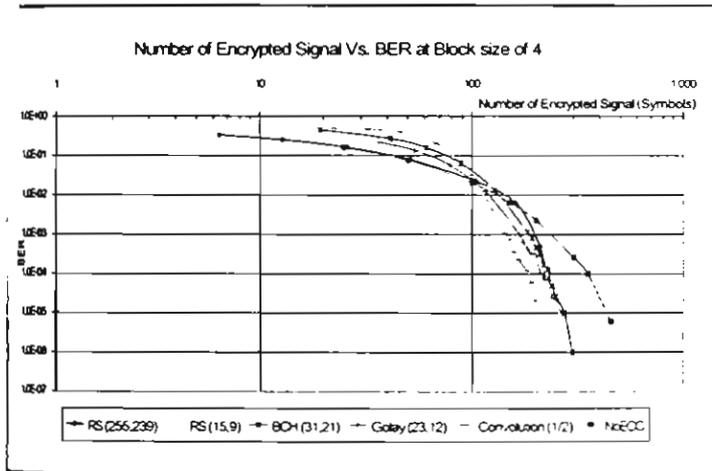


Figure 8. Performance comparison between the scheme with and without the error control codes at the block sizes of 4

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.

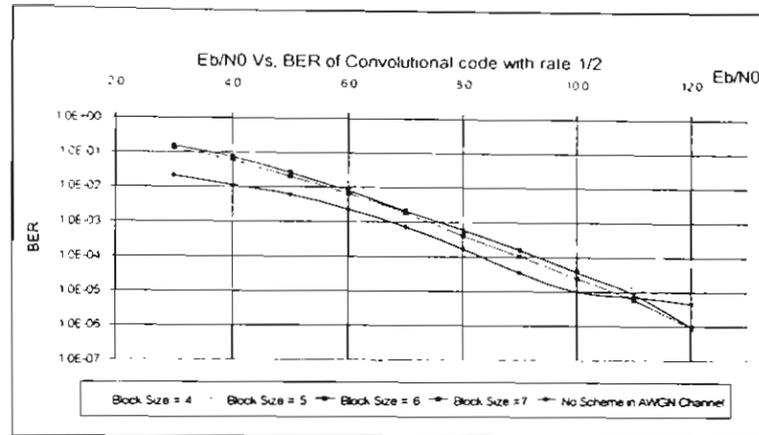


Figure 9. Performance comparison when the scheme is implemented in the AWGN channel at various block sizes

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. Shamoan, "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 ciphertext. This ciphertext is then broadcast via a public channel to all receivers. Only the authorized receivers with a key can decrypt this ciphertext and view the original plaintext. Normally, there are two kinds of digital scrambling technique that are commonly 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 linear 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,

due to the concatenated hash value. This lengthens the calculation time and, in some applications, the increased 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, \dots 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, \dots 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.

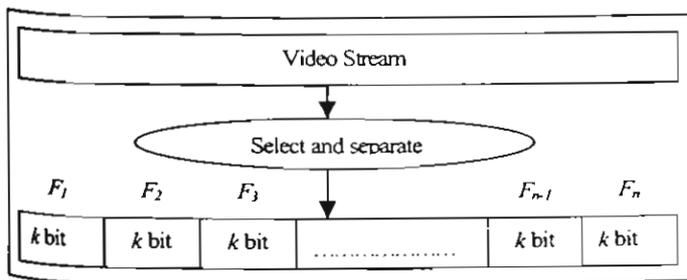


Figure 1: Bit selection and separation process

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 $F'_1, F'_2, F'_3, \dots, F'_n$. Then use this result, together with a pseudo-random number generator (G), initial vector (IV) again, to define $F''_1, F''_2, F''_3, \dots, F''_n$

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

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

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

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

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

5. Concatenate $F''_1, F''_2, \dots, F''_n$ 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 .

$$\begin{aligned} FR'_1 &= G(h(FR)) \\ FR'_i &= G(FR'_{i-1} \oplus i-1) \end{aligned} \quad (\text{Eq.3})$$

where $i = 2, 3, \dots, n/2$

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

$$\begin{aligned} FR' &= FR'_1 \parallel FR'_2 \parallel \dots \parallel FR'_{n/2} \quad (\text{Eq.4}) \\ ML &= FR' \oplus FL \quad (\text{Eq.5}) \end{aligned}$$

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 .

$$\begin{aligned} ML'_1 &= G(h(ML)) \\ ML'_i &= G(ML'_{i-1} \oplus i-1) \end{aligned} \quad (\text{Eq.6})$$

where $i = 2, 3, \dots, n/2$

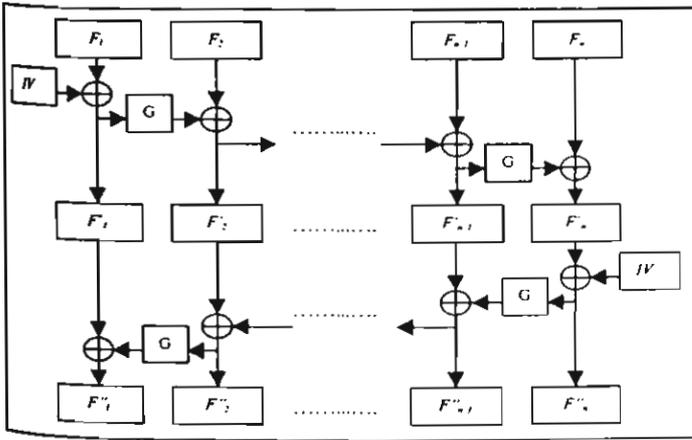


Figure 2: Step 1 of marking process

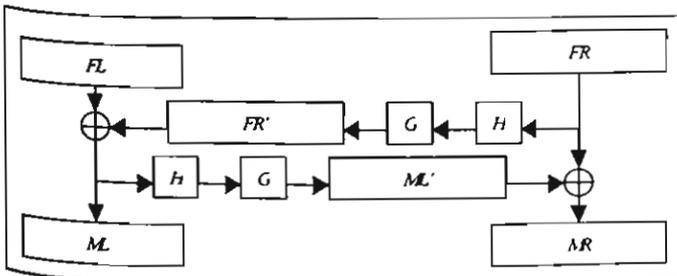


Figure 3: Steps 3 to 6 of marking process

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

$$ML' = ML'_1 \parallel ML'_2 \parallel \dots \parallel ML'_{n/2} \quad (\text{Eq.7})$$

$$MR = ML' \oplus FR \quad (\text{Eq.8})$$

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.

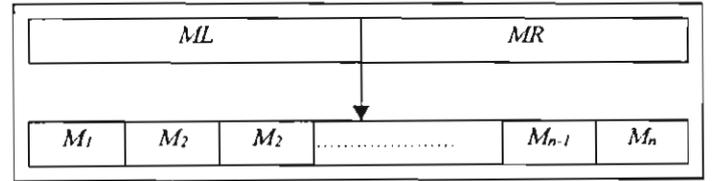


Figure 4: Position of marked blocks

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 .

$$\begin{aligned} \tilde{ML}'_1 &= G(h(ML)) \\ \tilde{ML}'_i &= G(\tilde{ML}'_{i-1} \oplus i-1) \end{aligned} \quad (\text{Eq.9})$$

where $i = 2, 3, \dots, n/2$

$$ML' = \tilde{ML}'_1 \parallel \tilde{ML}'_2 \parallel \dots \parallel \tilde{ML}'_{n/2} \quad (\text{Eq.10})$$

$$FR = ML' \oplus MR \quad (\text{Eq.11})$$

2. Compute FL from FR and ML

$$\begin{aligned} FR'_1 &= G(h(FR)) \\ FR'_i &= G(FR'_{i-1} \oplus i-1) \end{aligned} \quad (\text{Eq.12})$$

where $i = 2, 3, \dots, n/2$

$$FR' = FR'_1 \parallel FR'_2 \parallel \dots \parallel FR'_{n/2} \quad (\text{Eq.13})$$

$$FL = FR' \oplus ML \quad (\text{Eq.14})$$

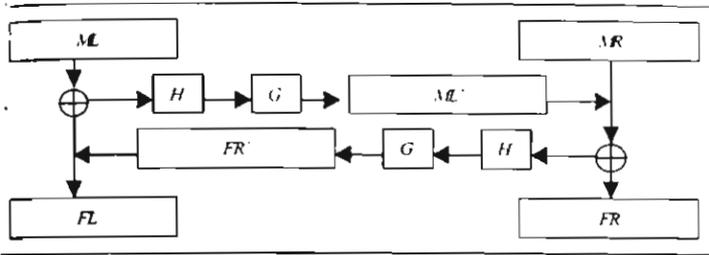


Figure 5: Steps 1 and 2 of the reverse of marking process

- Put FR and FL together and separate the sum into n blocks ($F''_1, F''_2, F''_3, \dots, F''_n$). Then use IV and $F''_1, F''_2, F''_3, \dots, F''_n$ 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 $i = 2, 3, \dots, n$

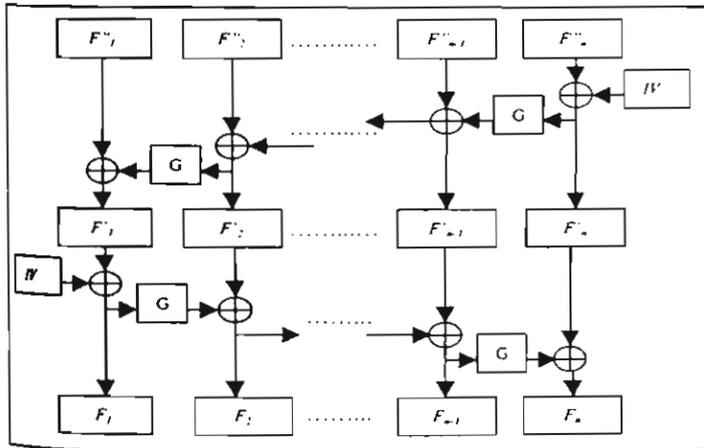


Figure 6: Step 3 of the reverse of marking process

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

There are several major requirements when scrambling the digital video streams: the security level, the computation time required and the complexity of scrambling devices. Our scheme has satisfied all three requirements by using the existing device efficiently, such as the smart card. It also provides sufficiently high security level while reducing the computation time when compared to encryption of the entire video streams. Because we can choose among different algorithms for our system, this makes it flexible and adaptable to technological 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.
- [4] B. Schneier, 'Applied Cryptography', Wiley: New York, 1996.
- [5] R. F. Graf and W. Sheets, 'Video Scrambling & Descrambling for Satellite & Cable TV', the 4th Edition, Howard W. Sams & Company, 1989.
- [6] W. Mooij, 'Advances in Conditional Access Technology', International Broadcasting Convention, No. 447, September, 1997, pp. 461-464.
- [7] S. R. Ely and S. R. Shuttleworth, 'Conditional Access Scrambling Techniques for Terrestrial UHF Television Broadcasts', International Broadcasting Convention, 1988 pp. 318-322.
- [8] M. Bellare and P. Rogaway, 'Optimal Asymmetric Encryption', EUROCRYPT 94, No. 950, Springer-Verlag, 1994, pp. 92-111.
- [9] R. Rivest, 'All-Or-Nothing Encryption and the Package Transform', the 4th International Workshop on Fast Software Encryption, Vol. 1267 of Lecture Notes in Computer Science, Springer-Verlag, 1997, pp. 210-218.
- [10] J. Markus, P. S. Julien and Y. Moti, 'Scramble All, Encrypt Small', Fast Software Encryption, 1999.
- [11] D. Johnson and S. Matyas, 'Asymmetric Encryption : Evolution and Enhancements', CryptoBytes, Vol 2, No.1, Spring, 1996
- [12] F. K. Tu, C. S. Lai, 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 extract 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 $F'_1, F'_2, F'_3, \dots, F'_n$. Then use this result, together with G, IV again, to define $F''_1, F''_2, F''_3, \dots, F''_n$.

$$\begin{aligned} F'_1 &= F_1 \oplus IV \\ F'_i &= F_i \oplus G(F'_{i-1}) \end{aligned} \quad (\text{Eq.1})$$

$$\begin{aligned} F''_n &= F'_n \oplus IV \\ F''_{i-1} &= F'_{i-1} \oplus G(F''_i) \end{aligned} \quad (\text{Eq.2})$$

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

2. Concatenate $F''_1, F''_2, \dots, F''_n$ 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 .

$$\begin{aligned} FR'_1 &= G(h(FR)) \\ FR'_i &= G(FR'_{i-1} \oplus i-1) \end{aligned} \quad (\text{Eq.3})$$

where $i = 2, 3, \dots, n/2$

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

$$\begin{aligned} FR' &= FR'_1 \parallel FR'_2 \parallel \dots \parallel FR'_{n/2} \quad (\text{Eq.4}) \\ ML &= FR' \oplus FL \quad (\text{Eq.5}) \end{aligned}$$

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 .

$$\begin{aligned} ML'_1 &= G(h(ML)) \\ ML'_i &= G(ML'_{i-1} \oplus i-1) \end{aligned} \quad (\text{Eq.6})$$

where $i = 2, 3, \dots, n/2$

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

$$\begin{aligned} ML' &= ML'_1 \parallel ML'_2 \parallel \dots \parallel ML'_{n/2} \quad (\text{Eq.7}) \\ MR &= ML' \oplus FR \quad (\text{Eq.8}) \end{aligned}$$

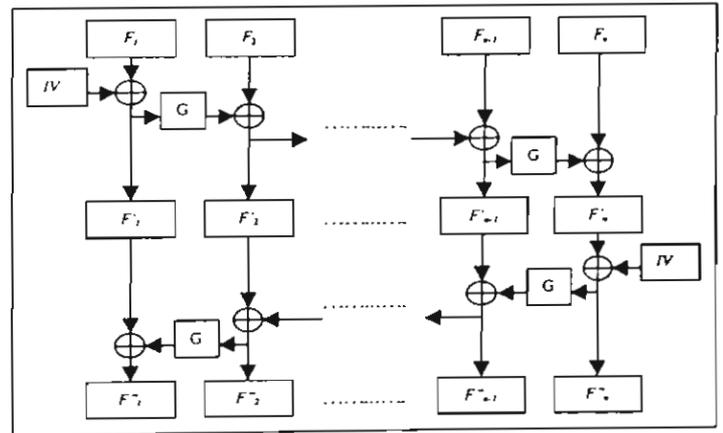


Figure 1. Step 1 of masking process

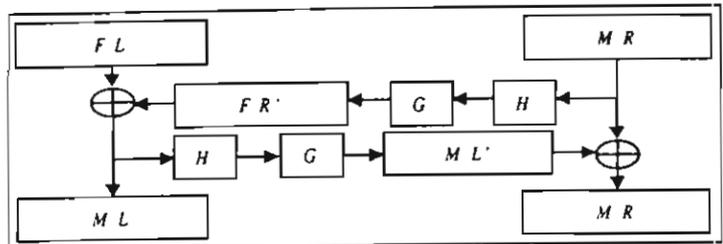


Figure 2. Steps 3 to 6 of masking process

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

1. Masked message ($M \setminus MR$) is divided into blocks of equal size called masked blocks. There are n masked blocks (called $M_1, M_2, M_3, \dots, M_n$) 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.

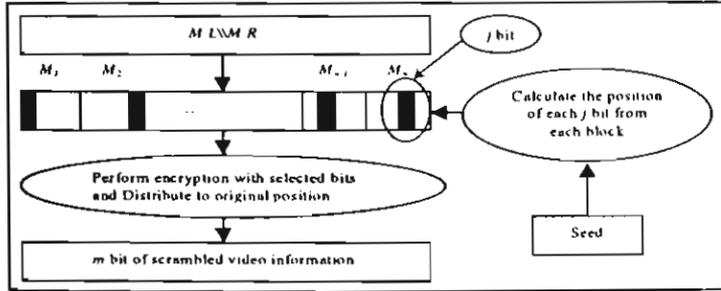


Figure 3. Encryption scheme

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 Remainder 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, \dots, p_t be positive integers that are pair wise relatively prime, and let R_1, R_2, \dots, R_t be positive integers, and let $N = p_1 * p_2 * \dots * p_t$. Then the set of congruous equations

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

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

$$X = (\sum_{i=1}^t (N/p_i) * R_i * f_i) \text{ mod } N \quad (\text{Eq.18})$$

where $1 \equiv f_i * (N/p_i) \text{ mod } p_i$

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:

$$\begin{aligned} X &= C_1 \text{ mod } p_1 \\ X &= C_2 \text{ mod } p_2 \\ X &= C_3 \text{ mod } p_3 \end{aligned}$$

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-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-box}) of each user i (U_i). For the smart card and the decoder box that are paired, IN_{i-card} of each user is the same as IN_{i-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

$$C_1 = E_{K_{card}}(R \setminus S) \quad (\text{Eq.19})$$

$$C_2 = E_{K_{box}}(R \setminus IV) \quad (\text{Eq.20})$$

$$C_3 = E_{K_{box}}(K_s) \quad (\text{Eq.21})$$

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.

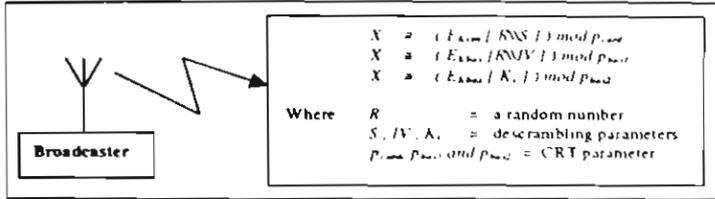


Figure 4. Steps involved in generating X

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} .

$$R \setminus S = D_{K_{card}}(X \bmod p_{card}) \quad (\text{Eq.22})$$

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

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

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_{box1}, p_{box1} and p_{box2} .

$$R \setminus IV = D_{K_{box1}}(X \bmod p_{box1}) \quad (\text{Eq.24})$$

$$K_s = D_{K_{box2}}(X \bmod p_{box2}) \quad (\text{Eq.25})$$

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

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

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

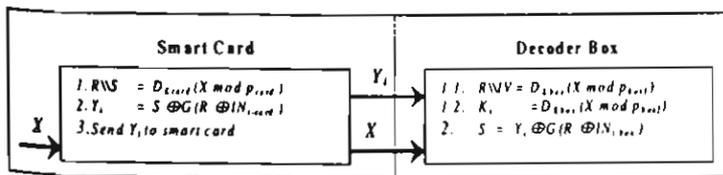


Figure 5. Operations between the decoder box and the smart card

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 ($t_e \gg t_h \gg t_r$). 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 known 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_i required to break the system. This is because if the attackers want to know the ciphertext of IV , S and K_i 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_i 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 calculate 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 $IN_{i,card}$ in the smart card. It is very hard for attackers to know R and $IN_{i,card}$ 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 $IN_{i,card}$ 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 scheme 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. Rivest, 'All-Or-Nothing Encryption and the Package Transform', the 4th International Workshop on Fast Software 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.

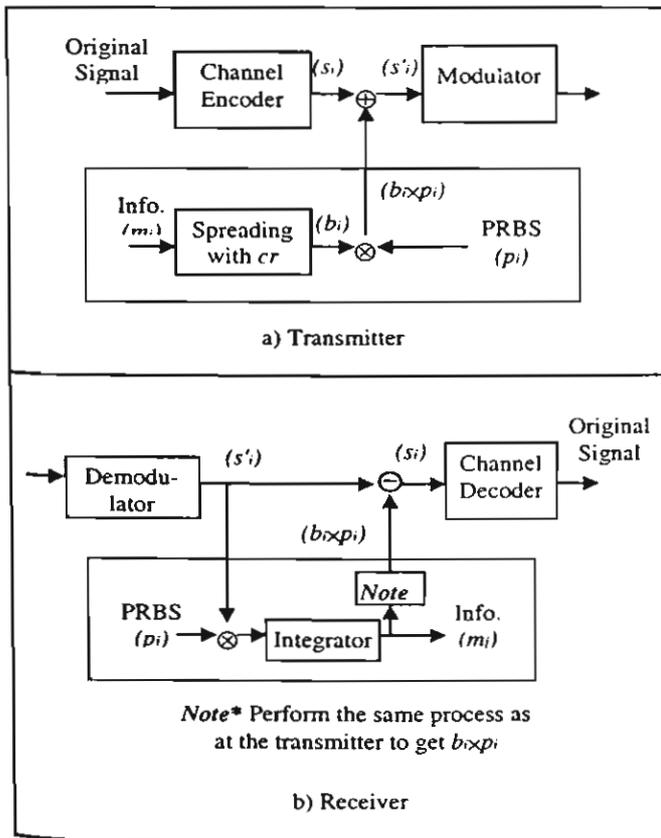


Figure 1. The operation of the encoding scheme

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) , $b_i = m_j$, $j \cdot cr \leq i < (j+1) \cdot cr$. 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) ,

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

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:

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

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

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

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

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 $p_i \cdot b_i$ can be performed in five different methods, yielding five operations, as follows:

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

- 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	$s_i + p_i \bullet b_i$							
	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 their 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.

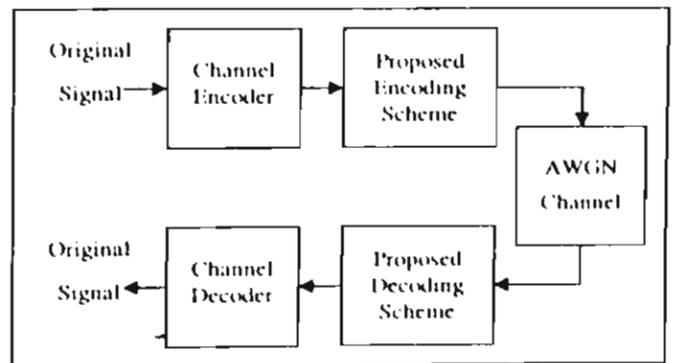


Figure 2. Simulation model in an AWGN channel

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 .

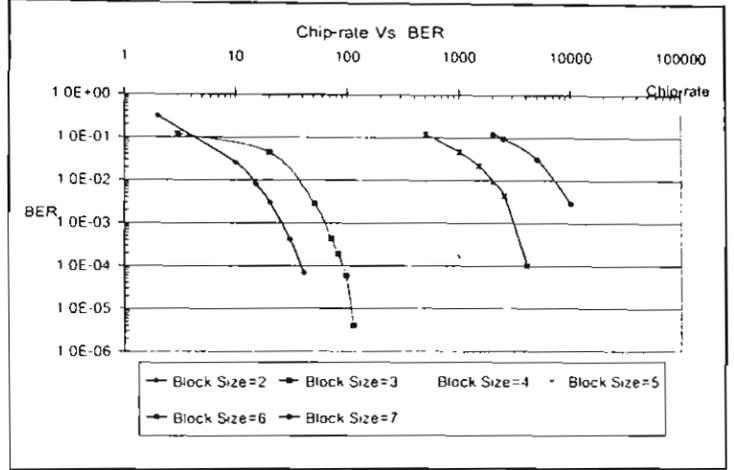


Figure 3. Bit error rate of extracted extra bits at different block sizes using adding method ii

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: $(0 \times -1) + (1 \times 1) + (0 \times -1) + (2 \times 1) + (1 \times -1) + (3 \times 1) + (2 \times -1) + (3 \times 1) = 6$. 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 in the encoding scheme.

As mentioned earlier, although the added bits are correctly obtained when a proper adding method is used, the 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.

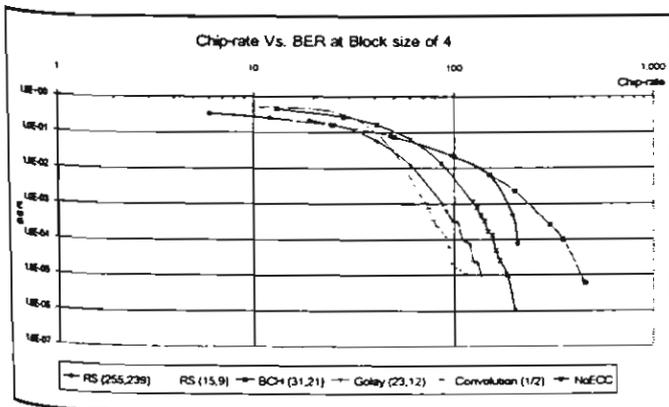


Figure 4. Bit error rate of extracted extra bits when the error control codes are applied, at the block sizes of 4

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.

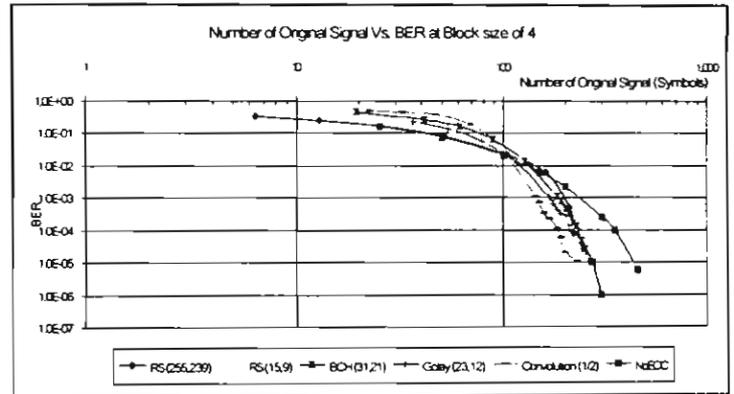


Figure 5. Performance comparison between the scheme with and without the error control codes

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.

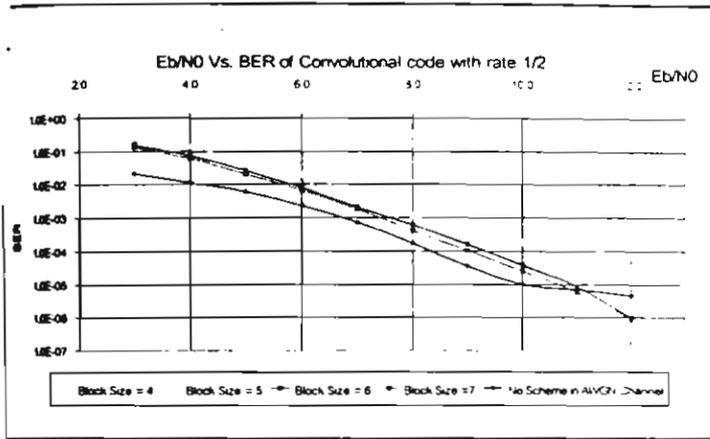


Figure 6. Performance comparison when the scheme is implemented in the AWGN channel at various block sizes

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.

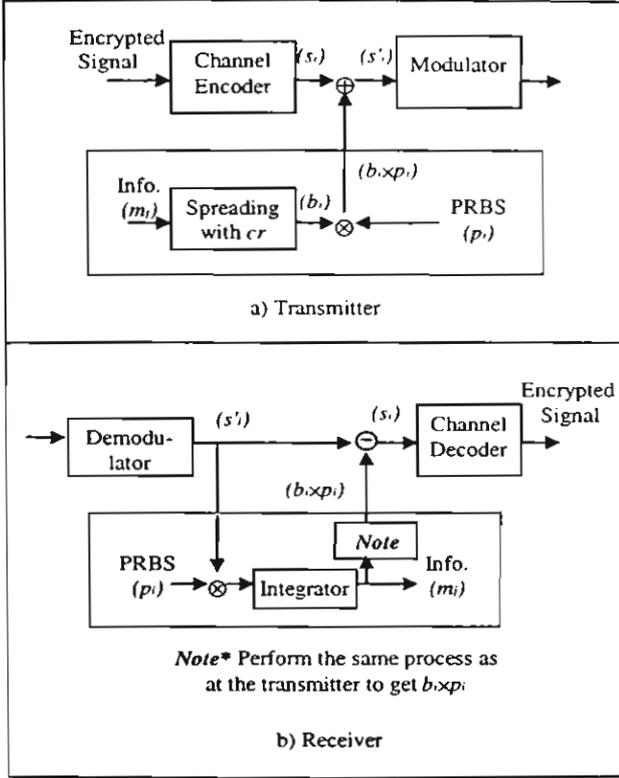


Figure 1. The operation of the encoding scheme

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) , $b_i = m_j$, $j \cdot cr \leq i < (j+1) \cdot cr$. 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) ,

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

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:

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

The first term on the right-hand side of (2) vanishes if p_i and s_i are uncorrelated, and then

$\sum_{i=j \cdot cr}^{(j+1) \cdot cr-1} p_i = 0$ [5]. Since $p_i^2 = 1$, r_j ideally becomes

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

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

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 $p_i \cdot b_i$ 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 $p_i \cdot b_i$ is performed as follows;

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

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 their 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.

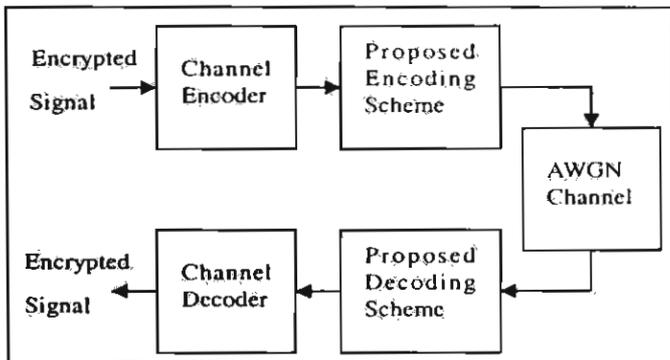


Figure 2. Simulation model in an AWGN channel

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 inner 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.

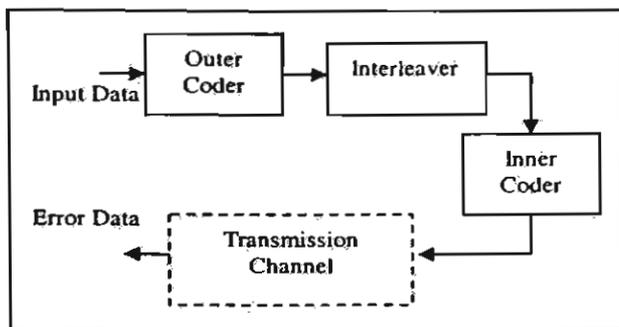


Figure 3. Concatenated error control scheme

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.

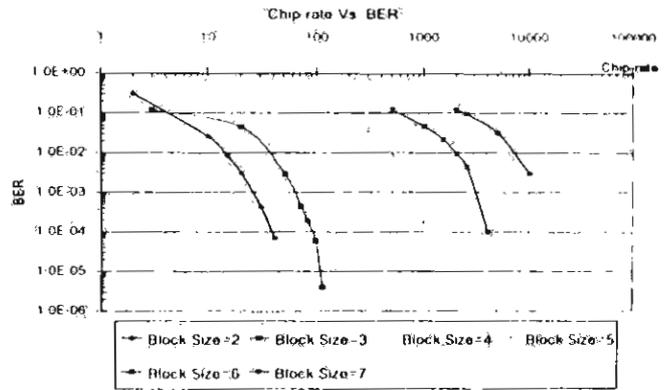


Figure 4. Bit error rate of extracted extra bits at different block sizes

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.

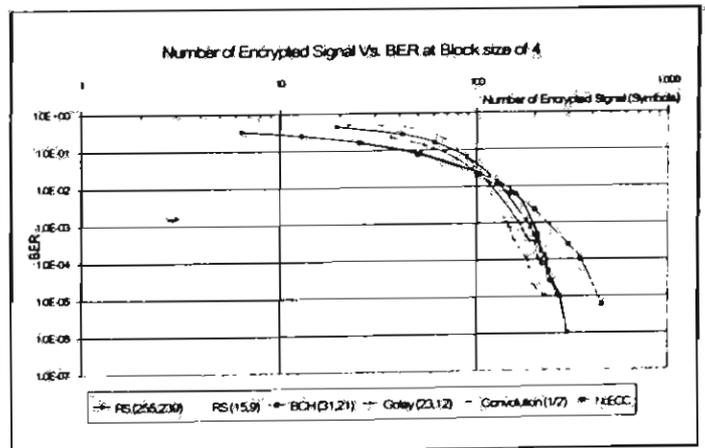


Figure 5. Performance comparison between the scheme with and without the error control codes at the block sizes of 4

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.

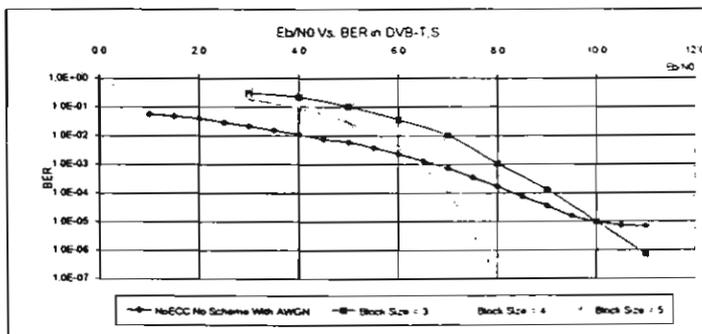


Figure 5. Performance of the scheme in DVB systems at different block sizes

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. Shanon: "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.