An Enhanced Cost Effective Line Shuffle Scrambling System with Secure Conditional Access Authorization

Andy Trott, BSC
Research and Development Manager
DCE Satellite Communications

Michael E. Long
Director of Engineering
Zenith Cable Products

ABSTRACT

Many systems currently exist to scramble a composite video television picture, but most suffer from both technical and commercial flaws which can seriously compromise performance and security. Most recently, digital compression and transmission of NTSC, PAL or SECAM video has been envisioned as the ultimate answer to video scrambling and protection from theft of service, but the cost of implementation is high. This paper presents an alternative analog scrambling system with the benefits of video digital processing for hard security and concealment without the high costs associated with full digital compression.

A line-shuffling analog video scrambling system, compatible with NTSC, PAL or SECAM systems, is presented which randomly displaces video lines in a video field to render the resulting television image unviewable but capable of being transmitted and received with standard equipment. By utilizing special memory addressing techniques, only one block of memory is required to continuously scramble successive blocks of video, thereby reducing system components and cost. A fully encrypted conditional access system is also included to provide protection against piracy without the need for "smart-cards" to renew security.

INTRODUCTION

Analog video scrambling systems, and their associated conditional access methods have been in existence since the concept of "Pay TV" began in the 1940's. The popularity and longevity of analog scrambling is due primarily to the low cost of implementation. Methods currently used to render TV video unviewable (until authorized) largely rely on suppression or elimination of video scan synchronization to encode the video (sync suppression). Sync suppression is effective but can be prone to theft of service if the timing-location of the suppressed syncs can be easily determined by analysis of video, audio, or in-band decoding data. Sync suppression additionally proves difficult to implement in CATV head-end modulation, terrestrial broadcast re-transmission equipment or subscriber reception equipment due to the lack of synchronization pulses with which to clamp video. Other analog methods, such as rapid video inversion, random line delay or line-cut-and-rotate systems, can be prone to serious residual artifacts due to the scrambling techniques or non-linearity of transmission systems. Such artifacts may themselves become a weakness in the security of the scrambled signal, allowing for simple signal piracy techniques.

There is a need for a high performance, secure, opaque scrambling method that is cost effective and requires no modification to video transmission or reception equipment. A video line shuffling system with secure encrypted descrambling sequence information and conditional access control data has been developed to satisfy those needs. The system, co-developed by DCE and Zenith, is called "DigiCrypt."

LINE SHUFFLE VIDEO SCRAMBLING

Line Shuffle scrambling is a technique whereby video lines are interchanged within a field of video so as to destroy the entertainment value and information content of a television program to an un-authorized viewer. There are basically two types of line shuffling.

The first type, which can be called "Field Line Shuffling", is where a number of video lines are displaced within a whole video field. Field Line Shuffling suffers with the problem that if the number of lines actually shuffled is less than the number of active video lines within the video field (288 for a PAL system), some of the video lines will remain unshuffled during that field. This can cause the scrambling effect to be less opaque than other scrambling methods even though any line chosen to be displaced can in fact be displaced anywhere within the video field.

The second type of line shuffling, which can be called "Block Line Shuffling", is where the video field is split up into "Blocks" of N video lines and every line of video within each block is displaced from it's original position, rather like shuffling a pack of cards (see Fig. 11). However, this method suffers with the problem that any line can only be displaced from it's original position by up to N lines, i.e. the block size. However, in a system where N is variable, this can be a strength, as the density of the scrambling can be selected by the program provider.

Block Line Shuffling

The DigiCrypt system is based upon Block Line Shuffling as it was considered to be the more flexible system and able to offer a higher degree of opacity. However, one drawback of splitting the video field into blocks of N lines is that the PAL, SECAM and NTSC standards do not offer a 'friendly' number of lines per field, making it difficult to select a number for N to comfortably fit whole blocks within the video field, especially considering that N should be a power of two to maximize the efficiency and utilization of the video memory.

For example, if we select N to be 32 and allow 288 lines of active video per field for a PAL system, we get 9 whole blocks per field. However, if we then transpose this format to an NTSC system and allow 240 lines of active video per field we get 7.5 blocks and therefore we find that the last block will wrap-around into the next video field.

This then leaves us with one of two possible solutions. Either we allow a pre-defined number of video lines at the top and/or bottom of the video to be clear (unscrambled) thereby forcing a multiple of whole blocks, or we design the architecture of the system to allow blocks of N lines to wrap-around into the next video field, only pausing to let the field blanking interval through. Although the former is attractive due to the simplicity of design required, it was considered that the latter would render the most flexibility from the system. This would therefore allow any size of blocks and any number of lines per video field, the only consideration then being how many video fields it would take for the shuffling of blocks to return to a field boundary for re-synchronization of encoder and decoder.

In the DigiCrypt system, N is chosen to be either 32 or 128. This enables the system to give a very high degree of opacity, via the 128 line option, or more visibility, via the 32 line option for 'teaser' viewing. It also means that a very cost effective decoder may be produced using a smaller amount of memory, which will decode 32 line block transmissions only, or an "all-singing, all-dancing" decoder which can decode both 32 line and 128 line transmissions using the full amount of memory.

System Overview

The line shuffling system can be used for terrestrial broadcast, satellite, microwave or cable TV applications. Fig 1 shows a typical system configuration transmitting over a satellite link. The transmission side of the system, consists of an encoder and conditional access computer containing a data-base of subscriber authorizations. The video feed, which could typically come from a television studio, is fed into the encoder, digitized by an analog-to-digital converter and then scrambled by Block Line Shuffling. Data coming in from the conditional access computer is packetized, a scrambling seed added, and then the whole message is encrypted and inserted onto four unused lines in the field blanking interval before being converted back into the analog domain by a digital-to-analog converter and transmitted over the satellite link. The scrambling seed is generated from a random number generator within the encoder (possibly a noise source) and is used by the line shuffling algorithm each time a re-synchronization occurs. The "re-synchronization" decision, generated in the encoder, is sent as a signal along with the in-band data to inform all decoders that a new seed should be used to initialize the block line shuffling algorithm.

The reception side of the system in this example, consists merely of a satellite receiver and a line shuffle decoder. The decoder constantly monitors all field blanking interval lines for in-band data, and upon recognition, determines whether it is authorized for the current programming. If it is authorized, decoding starts automatically.

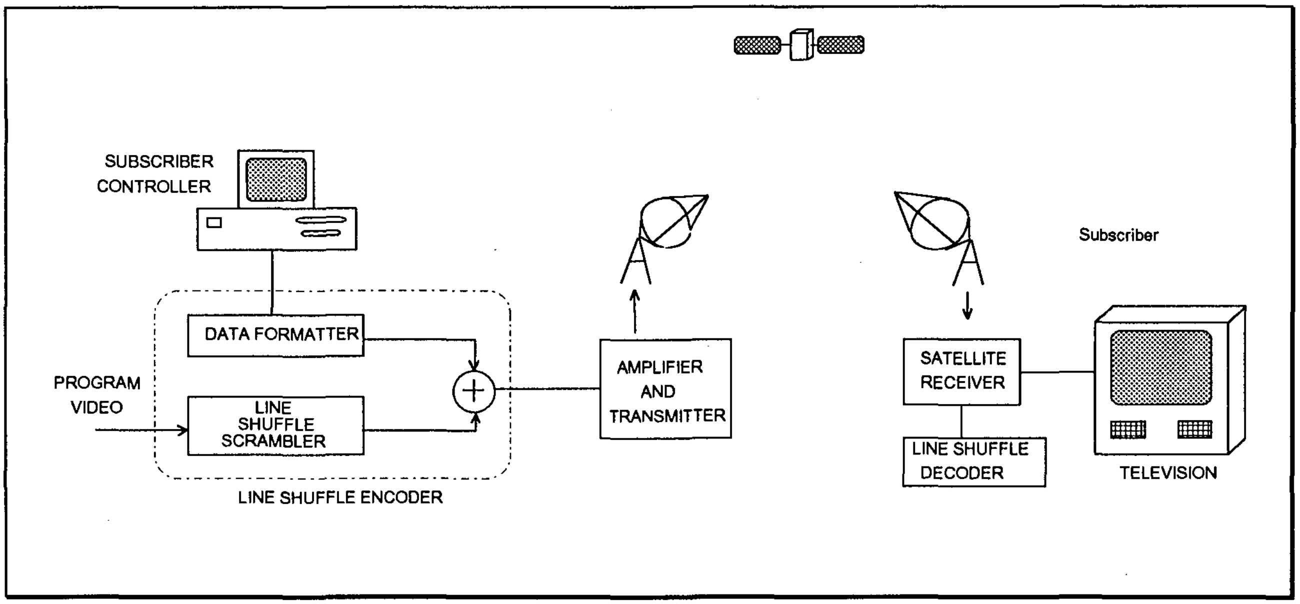


Fig 1: Block diagram of a typical system configuration using Block Line Shuffle Scrambling

Block Shuffling and System Re-Synchronization

Each block of N video lines is shuffled using a Pseudo Random Number sequence which is "locked" between both the encoder and authorized decoder. The "locking" of the encoder and decoder is vital in order for the decoder to reposition the displaced video lines into their original and correct positions. This "locking" is performed by the synchronization pulse every F_n fields, where F_n is a multiple of the number of fields required for the blocks to return to a field boundary. The repetition rate of the re-synchronization pulse will determine how quickly a decoder will lock to an authorized program once it is tuned to the correct channel. From tests, it was found that a repetition rate of between 0.5 and 1 second gave the best result.

Because the re-synchronization is such a vital part of the system, much effort has gone into designing it to be very robust and reliable. Hand-in-hand with the re-synchronization is the transmission of the scrambling seed from the encoder to all authorized decoders - it is all very well for a decoder to recognize the re-synchronization, but if the scrambling seed was corrupted prior to reception, the decoder block shuffling algorithm will be exercising a different sequence than the encoder, and hence the decoder will "drop-out" until the next successful re-synchronization occurs. This then necessitates that the scrambling seed is sent more than once between re-syncronizations to ensure that it has been received properly by the decoder. In fact, the more times the better, although an acceptable balance must be found on the number of times the scrambling seed is sent, as it follows that it will occupy valuable in-band data bandwidth which could be used for other purposes.

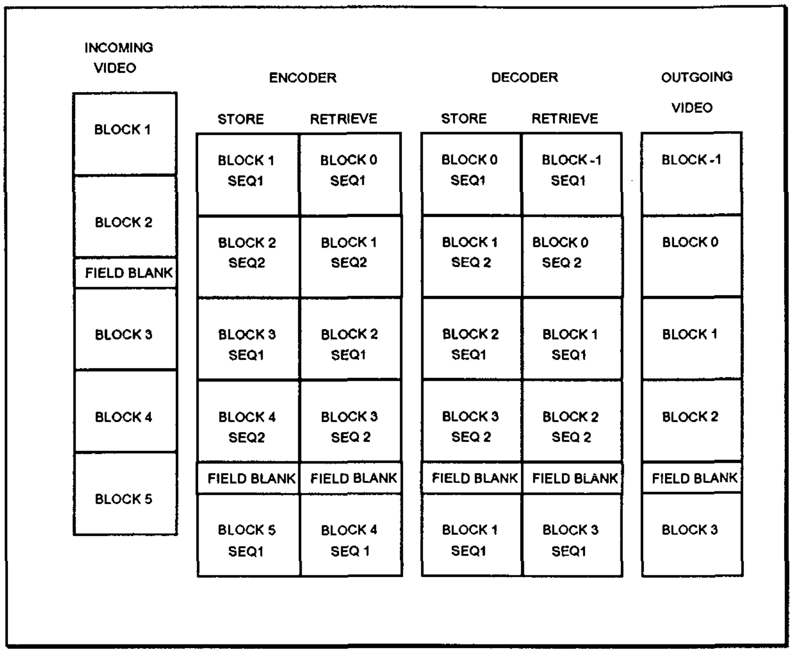
From tests, the balance that has been found acceptable is that the scrambling seed is sent 5 times in between re-synchronization pulses, and that the decoder will only accept a scrambling seed once it has received the same seed twice in succession.

System Timing and Field-Blanking Interval

The video to be scrambled is pre-formatted in the encoder in such a way as to make the decoding as simple and cost-effective as possible.

The Field Blanking Interval must not be scrambled due to the in-band data and teletext it contains. Due to delays through the system while scrambling, the Field Blanking Interval has to be stored and then retrieved in the encoder at a time when it is required by the decoder. This is to ensure that the decoder only has to "pass" the field blanking interval without processing it or using any of it's memory.

Fig 2 shows the overall system timing and subsequent delays introduced by Block Line Shuffling. It should be noted that everything revolves around the correct reconstruction of output video from the decoder.



ENCODER		DECODER		OUTGOING VIDEO
STORE	RETRIEVE	STORE	RETRIEVE	
BLOCK 1 SEQ1	BLOCK 0 SEQ1	BLOCK 0 SEQ1	BLOCK -1 SEQ1	BLOCK -1
BLOCK 2 SEQ2	BLOCK 1 SEQ2	BLOCK 1 SEQ 2	BLOCK 0 SEQ 2	BLOCK 0
BLOCK 3 SEQ1	BLOCK 2 SEQ1	BLOCK 2 SEQ1	BLOCK 1 SEQ1	BLOCK 1
BLOCK 4 SEQ2	BLOCK 3 SEQ 2	BLOCK 3 SEQ 2	BLOCK 2 SEQ 2	BLOCK 2
FIELD BLANK	FIELD BLANK	FIELD BLANK	FIELD BLANK	FIELD BLANK
BLOCK 5 SEQ1	BLOCK 4 SEQ 1	BLOCK 1 SEQ1	BLOCK 3 SEQ1	BLOCK 3

Fig 2: System timing chart

From Fig 2, it can be seen that Block Line Shuffling introduces a delay of:

$$D=(2N+FB_n)\times Lt \quad \text{(seconds)}$$

where: N = Block Size (in lines)

FB_n = Field Blank Size (in lines)

Lt = Line time (in Sec)

Therefore for a 32 Line Shuffle PAL system, where N=32, FB_n=25 and Lt=64 x 10^{-6}:

$$D=(2\times 32+25)\times 64\times 10^{-6}\ \text{sec}$$

$$=5.696\ \text{mSec}$$

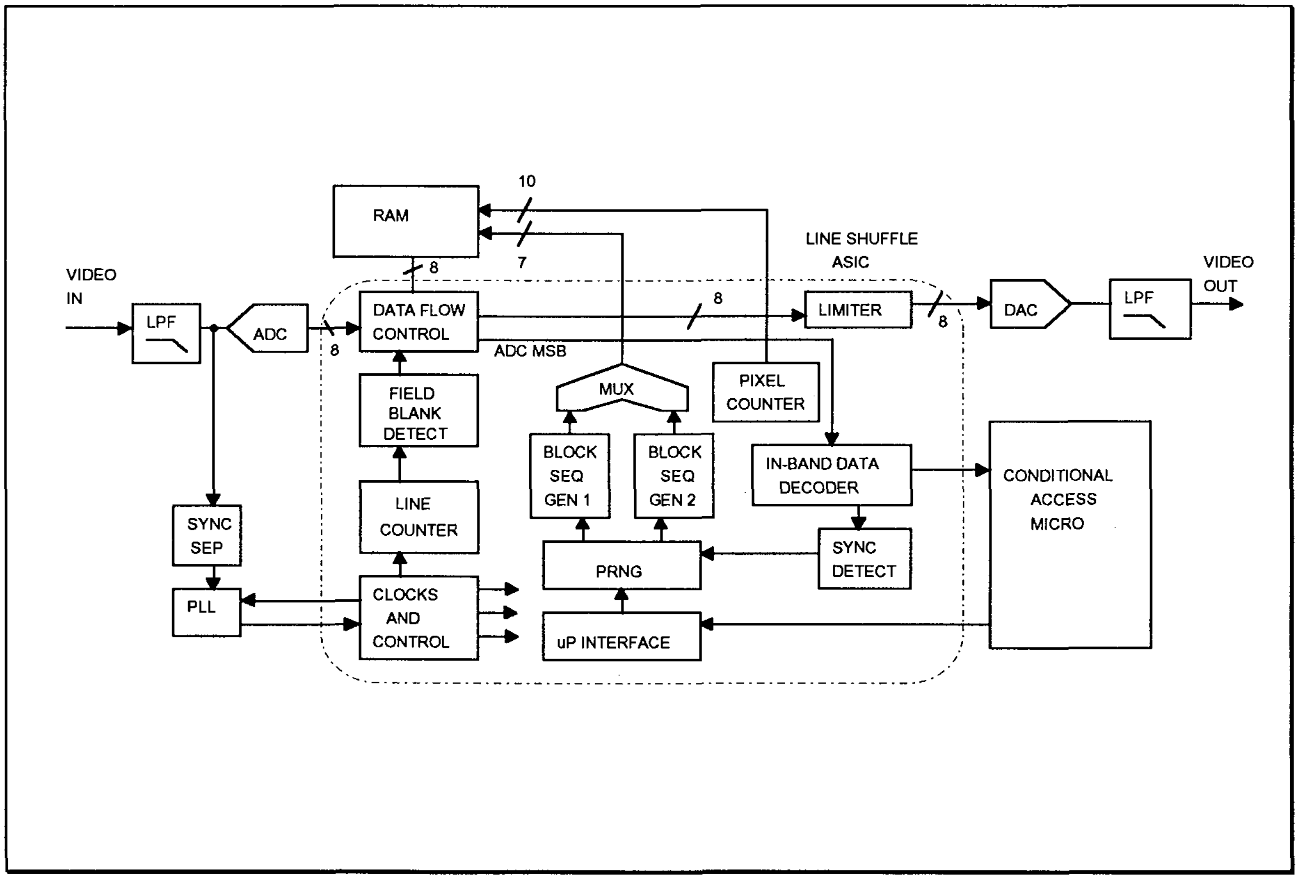


Fig 3: Block diagram of the block line shuffling decoder

The Decoder

The decoder is essentially the heart of the system, as the demands placed on it from the commercial world are high. It should cause no visual picture degradation, be secure from piracy, reliable while decoding, robust in a noisy environment and, most importantly, very cost effective.

Fig 3 shows a block diagram of the decoder, which is built around two fundamental components. The first is the CONDITIONAL ACCESS MICRO which is a full custom integrated circuit containing a Microprocessor, ROM, RAM and secure EEPROM, while the second is an Application Specific Integrated Circuit (ASIC), which is shown as a dotted outline. All blocks within the outline exist as part of the ASIC.

The video input is digitized by an 8 bit analog to digital converter (ADC) and fed into the DATA FLOW CONTROL block which is used to control the flow of digital video. While decoding a line shuffle block, the RAM is first read at it's current address and the resultant byte fed to the LIMITER. The input from the ADC is then written into the RAM for later retrieval. The LIMITER is invoked only if the in-band data lines occupy the first four active video lines and only then to ensure that the in-band data lines do not appear visible on a television set for aesthetic purposes. The output from the LIMITER is then fed to the digital to analog converter, DAC, for conversion back into the analog domain.

The most significant bit of the ADC, ADC MSB, is used as a data slicer and fed into the IN-BAND DATA DECODER, which permanently interrogates the incoming video lines in the field blanking interval to determine whether or not the line contains in-band data. This data decoder can reliably distinguish it's own data from any other known data type, including teletext. Once all four lines of in-band data have been collected, the re-synchronization pulse is removed and the remaining encrypted data sent to the CONDITIONAL ACCESS MICRO. This micro decrypts the data and decodes the packets for configuration information, scrambling seed and user authorizations. If it detects a scrambling seed and determines that the decoder is authorized for the current programming, the micro will write the scrambling seed to the uP INTERFACE for storage and subsequently to the PRNG when a re-synchronization is detected by SYNC DETECT.

The Phase Locked Loop, PLL, locks the whole system to the incoming video line and frame synchronization signals, which are in turn extracted from the video by the sync separator, SYNC SEP.

Decoder Memory Utilization

Because every effort has been made to ensure that the decoder is as low cost as possible, the whole scrambling system has been developed around the decoding process which employs a novel memory management algorithm.

It is usual to have two blocks of memory when changing the order of an incoming data stream, of any sort. The first block is used to write data into, while reading data from the second block in a different order. Once the first block is full, the writing then switches to the second block which by now has been completely read, and reading commences from the first block which is now full of new data.

For example, lets assume that N = 4 (i.e. shuffling in blocks of 4 video lines):

Assuming input video lines of 1,2,3,4......16, suppose we wish to shuffle these lines such that each successive block of four video lines are rearranged according to a new order, i.e.: 1,3,4,2; which we will call sequence S1. The resultant video line output order would therefore be:

1,3,4,2,5,7,8,6,9,11,12,10,13,15,16,14

The simplest way of achieving this result would be to use two blocks of four element memory as in Fig 4:

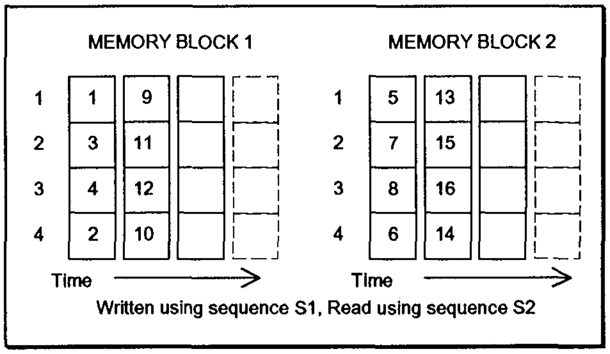


Fig 4

As each new incoming video line is written into the next available memory position of one memory block with a previously known sequence (in this case sequence S1), a video line is read from the memory element of the other memory block using a previously known sequence, different from the first sequence, say

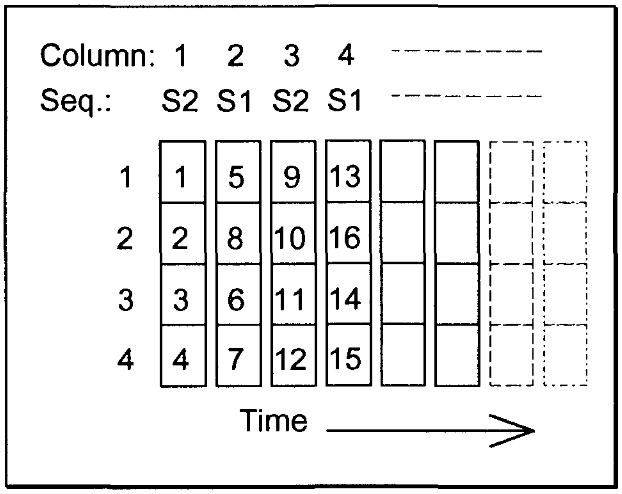
1,2,3,4, which we shall call sequence S2. Once each of the four memory elements of the block being written into is full, the writing and reading exchange blocks. The resultant video line output order is therefore:

x,x,x,x,1,3,4,2,5,7,8,6,9,11,12,10,13,15,16,14

where x is undefined due to the fact that initially, nothing has been written into the memory block where reading commences. From this, we can see that there will always be an inherent system delay.

There is however a more efficient way of producing a very similar result using half the amount of memory than the example just given. Suppose each element, or video line, is read and then immediately over-written using the same sequence, but that now the sequence changes each time all four elements have been completed.

Lets again use the same sequences as the last example, i.e. sequence S1 = 1,3,4,2 and sequence S2 = 1,2,3,4:



Column:	1	2	3	4	--------
Seq.:	S2	S1	S2	S1	--------
1	1	5	9	13	
2	2	8	10	16	
3	3	6	11	14	
4	4	7	12	15	

Fig. 5

As the video lines are read from and then written into the memory, it first uses sequence S2, shown as column 1 in Fig 5. After all four memory elements have been stored, the sequence changes to S1, shown as column 2. As each block in the sequence is read, it is immediately written to with the next video line. This process continues, giving the following result:

x,x,x,x,1,3,4,2,5,8,6,7,9,11,12,10,13,16,14,15

As can be seen, this result is slightly different to the two memory element solution, but still very usable. Indeed, this is the process which has been adopted in the decoder, where each data word represents one entire video line and N = 32 or 128.

This single memory solution however, requires a slightly more complex algorithm in the encoder, along with twice the amount of memory, but this is considered to be a justifiable trade-off, as the savings in the decoder are considerable.

Block Shuffling Sequence Generators

In the example given in the previous section, the two shuffling sequences were fixed, i.e. they didn't change from block to block. In a practical design, this would leave the system open to piracy, as a mere logic analyzer would reveal the shuffling order and allow a pirate to descramble the signal.

It is vital therefore that the sequence generators are continually changing in an apparently random and unpredictable way. Pseudo Random Number Generators (PRNG) are therefore utilized to provide this effect. Since it is common knowledge that Pseudo Random Number Generators on their own are completely logical and easily predicted, the sequence generator must be designed in such a way as to utilize Pseudo Random Number generators to form a highly complex, variable and apparently unpredictable algorithm, the basic structure of which is shown in Fig. 6.

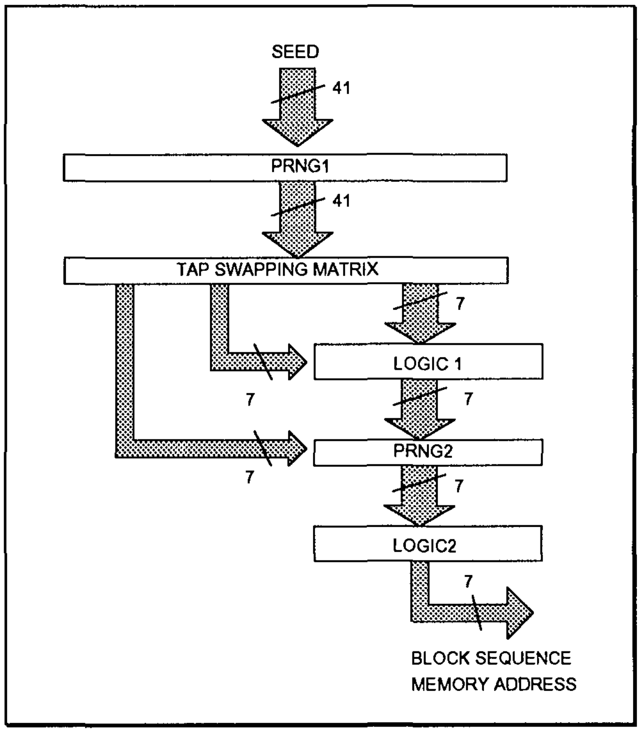


Fig 6: Basic structure of the Block Shuffling Sequence Generator

Whenever a re-synchronization occurs, the next SCRAMBLING SEED is clocked into the generator to provide the initial conditions. The SCRAMBLING SEED is used by PRNG1 which generates a set of completely unique sequences for all subsequent blocks between re-synchronization pulses. The output from

PRNG1 then feeds the TAP SWAPPING block which is used to further "randomize" the result and various taps are then taken and fed into the two LOGIC blocks around PRNG2, the function of which is to produce the actual line ordering sequence, such that for a 32 line block system, the resultant Pseudo Random Number numbers are 0-31 inclusive in a pseudo random order. The entire sequence generator is clocked once per block except for PRNG2 which is clocked every video line.

From Fig 6, it can be seen that each block line sequence is a function of 14 binary inputs, thereby giving 2^{14}, or 16,384 different line shuffle sequences. However, because these inputs change after every block between re-syncronizations, according to a 41 bit PRNG, the number of possible block sequences is:

$$2^{41} = 2.2 \text{ x } 10^{+12}.$$

Therefore, the number of block and line sequences that are possible from this sequence generator is:

$$2^{14} \text{ x } 2^{41} = 3.6 \text{ x } 10^{+16}$$

Also, if a new and unique scrambling seed was used every re-synchronization period of 1 sec, all possible sequences would be exhausted after:

$$2^{41} / 86{,}400 = 25.5 \text{ million days}$$

$$= 69{,}730 \text{ years}$$

In the DigiCrypt system, the SCRAMBLING SEED is assembled from two smaller seeds which together are used to provide the initial conditions for the Sequence Generators. These two smaller seeds are called the "Dynamic Seed" and the "Static Seed".

The "Dynamic Seed" changes every re-synchronization period and is transmitted from the encoder continually to ensure successful reception by the decoders. It's bit size is chosen to render "guess-work" completely ineffective by a potential pirate and yet still maintain a healthy size for the "Static Seed".

The "Static Seed" changes very infrequently and is transmitted from the encoder independently from the "Dynamic Seed". It occupies all other bits that are not assigned to the "Dynamic Seed".

Active Line Only Shuffling

In early developments of this project, it was envisaged that the whole video line should be scrambled including synchronization pulse and color burst, cutting each line just before the falling edge of the sync pulse (Fig 7a). However after much consulting and testing of the system with the BBC, it was felt that shuffling the entire line of a PAL signal could leave a "back-door" open to piracy. This was due to the fact that the color burst of a PAL signal continually rotates through 135° per video field. It was therefore considered theoretically possible, although unlikely, to determine the correct position of a line within a scrambled block by the absolute phase of it's color burst, even though the change in phase of each line with respect to the previous line is only 0.43°.

In view of this, an option to scramble only the active part of the video line was included in the DigiCrypt system (see Fig 7b) such that for systems scrambling a PAL signal, this option could be invoked rendering the possibility of piracy by this means ineffective.

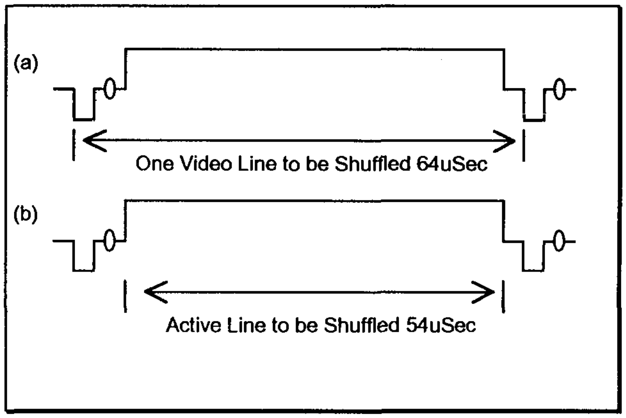

(a)
One Video Line to be Shuffled 64uSec
(b)
Active Line to be Shuffled 54uSec

Fig 7: Splicing points for scrambling the video line via Block Line Shuffling (a) The entire video line 64µSec; (b) The active portion of the video only 54µSec

Active Line Jitter

Active Line Jitter is a process whereby the active portion of a video line, approximately 54 µSec for a PAL signal, is displaced within the same line by a predefined amount, the direction of displacement changing randomly from line to line. Fig 8 shows this effect.

It should be stressed that this type of process is in no way a satisfactory scrambling method in it's own right, but by adding it to Block Line Shuffling, it adds another option which a system operator can invoke at anytime to frustrate potential pirates or to enhance the scrambled video's opacity.

The DigiCrypt system adds Active Line Jitter to it's repertoire with no extra system cost, as once the signal is in the digital domain, processes like this become very simple, using very little logic to realize.

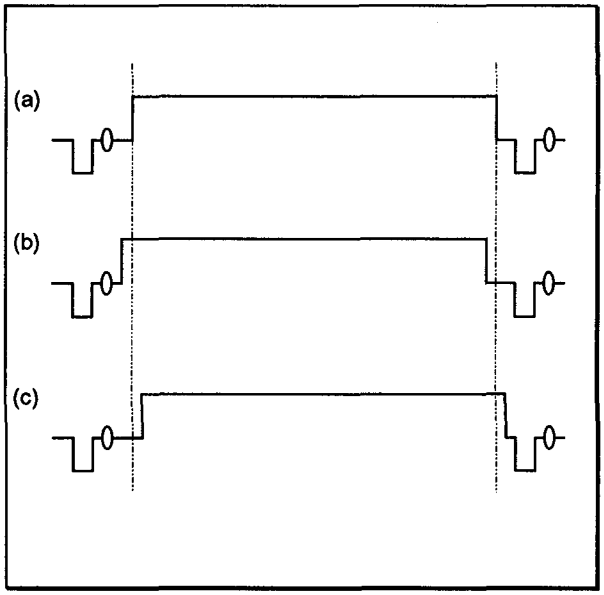


Fig 8: Active line jitter
(a) No horizontal displacement (b) Displacement 1µSec to left (c) Displacement 1µSec to right

The displacement chosen for the system is ±1µSec which is annoying enough to the viewers, and can indeed be used to "tease" subscribers into purchasing more services. However, in order to prevent the video line encroaching into the color burst and/or back porch and consequently impeding the PAL, NTSC or SECAM specifications, truncation of the front and back of the active video line is necessary at the encoder.

The Encoder

The encoder has been developed using Field Programmable Gate Array's (FPGA), for maximum product flexibility and size reduction, allowing the Video Encoder to occupy minimal rack space. 10 bit A/D and D/A converters have been used for optimum linearity, noise and distortion performance, giving very high quality broadcast video.

CONDITIONAL ACCESS

The conditional access system utilized in DigiCrypt is based on an error-protected, encrypted in-band data technique compatible with terrestrial, satellite and cable transmission methods. Each decoder has its own unique identification number and other security passwords and session keys buried within the proprietary Conditional Access Micro in secure non-volatile memory. New session keys may be issued at will by the encryption control system to any decoder by encrypting, with other encryption keys, the in-band data message addressed to the decoder's unique identification number (address). A "scrambling seed", changed approximately every second, and encrypted with one of the variable session keys, is used by the line-shuffling system to determine the correct line order sequence for decoding. Failure to receive the correct "scrambling seed" due to de-authorization or not having correct decryption keys results in a scrambled video display. A second secret "static seed" to be used with the dynamic "scrambling seed" can be changed in the unlikely event that the system is compromised, thus allowing for "renewable" security.

Conditional Access and System Security

The main method of restricting un-authorized subscriber access is via the absence of scrambling seeds being sent from the CONDITIONAL ACCESS MICRO to the ASIC. This means that the security of the system primarily lies with the security of the encrypted in-band data as opposed to the predictability of the block sequence generators. This is due to the fact that without knowledge of the next scrambling seed, a prospective pirate has a chance of 1 in 2^n of predicting the correct seed, where n is the number of bits allocated to the seed, which even for a seed as small as 8 bits would be 0.39%.

In-Band-Data

The in-band data occupies four video lines which can be positioned anywhere within the field blanking interval, or the first four active video lines, which must subsequently be blanked by the decoder. The data system incorporated utilizes 256 bits of digital data to be error-detected and decrypted every video field by the decoder. These 256 bits are split into four 64 bit data packets, each packet being placed on its own predetermined video line. Therefore, in order to transmit all data, four video lines are required and can be located anywhere in the VBI or, alternately, in active video if it is desired to keep the VBI intact for teletext, etc.. The decoder blanks data lines after decoding so that if data must occur in active video it will not be visible to viewers. The data is modulated onto a video line as shown in Figure 9:

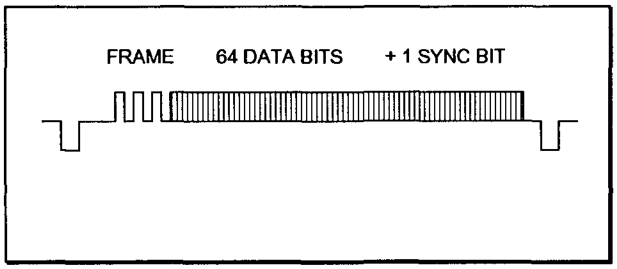


Fig 9: One Data Line

The data consists of a framing code followed by 64 Manchester encoded data bits and one synchronization bit. The instantaneous data rate is approximately 1.2 Mb/sec, significantly lower, and more robust than teletext data. The decoder detects data, over-sampling each data bit 10:1 to correctly identify each bit. Each half of the Manchester encoded bits are tested to be equal and opposite, and an error signal is generated if the result is negative. If eight or more errors are accumulated within a 64 bit packet, it is assumed that the packet under test is not a relevant in-band data packet and consequently "thrown-away". After each packet passes these tests, a 16 bit CRC error check is additionally made on every data packet to insure complete integrity. Finally, after all the error checking has been completed, the data must correspond to one of the valid command structures programmed in the Conditional Access Micro's ROM code in order to be acted upon.

Data Encryption

The four data packets transmitted within each video frame consist of a "global" data packet followed by three "individual" data packets. Each data packet is encrypted independently of the others with different encryption keys. In order to gain access to "scrambling seed" data (which is changed approximately once per second), secret keys must be used to gain access to other variable keys for decryption of the desired data.

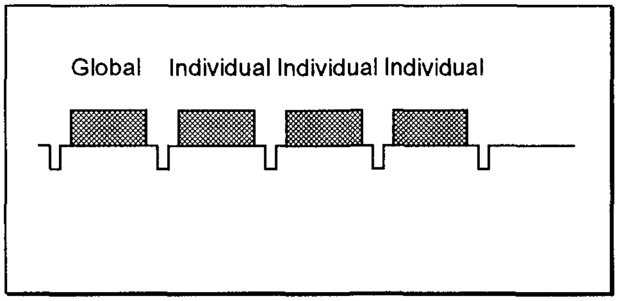


Fig 10: Conditional Access Data Packets

Global Data Packet

The "global" packet contains information of use to all valid decoders, such as local time synchronization, operator customized initialization data, and a variable "program tag" for the video program being received. This packet also periodically contains the "next" dynamic scrambling seed which is necessary for the decoder to use for un-shuffling the encoded video. For security, the "global" packet is encrypted with a 16 bit "session key" which can be changed periodically to a new "session key" previously downloaded to authorized individual decoders via independently encrypted "individual" data packets.

Individual Data Packets:

The "individual" data packets contain information to be addressed to individual valid decoders in the control computer's data-base. With three "individual" packets per field, 9,000 decoders per minute can be addressed in 50 Hz vertical scan rate video systems (i.e. PAL, SECAM) and 10,800 decoders per minute can be addressed in 60 Hz vertical scan rate video (i.e. NTSC) systems. The information in the packet can include commands to authorize (or de-authorize) a decoder for any program tag of any video program to be shown on any scrambled channel. The "individual" packet is also used to download current and future "session keys" to valid decoders to be used to decrypt the global packet.

For security, each "individual" packet is encrypted with a secret 16 bit "address key" unique to each decoder address. Decoder address data in the "individual" packet is also encrypted such that it cannot be identified simply by observing data. There are over 67 million individual addresses available for decoders. Decoder addresses and "address keys" are assigned to each decoder at manufacture and "sealed" into secure non-volatile memory which cannot be altered or read externally thereafter.

Authorization

Decoders are authorized by downloading to each valid decoder, in an encrypted "individual" data packet, a list of "program levels" for which it is authorized. This list is stored by the decoder in secure non-volatile memory for reference. The secure non-volatile memory is integrated on the same IC die with the Conditional Access Microprocessor. When a subscription or "pay-per-view" video program is tuned, the decoder

Conditional Access Microprocessor compares the encrypted "program tag" in the "global" packet with its list of "program levels" stored in memory. If the "program tag" in the encrypted video matches a "program level" previously loaded into the decoder, the decoder passes the scrambling seed (encrypted in the "global" packet) to the descrambling ASIC. This permits decoding of the selected video. If the "program tag" does not match any of the authorized program levels in memory, or if the decoder cannot decrypt the "global" packet (due to not having proper "session keys"), the scrambling seed will not be passed, and decoding of video will not occur. Decoders can store up to 256 unique "program levels" for which they may be authorized at any one time. The decoder also stores two independent encryption session keys which can be changed periodically for security or as a means of de-authorization to invalid decoders.

An example of the use of session keys for de-authorization would be to allow a subscriber to play back and decode a tape of a "scrambled" program through his decoder only for a pre-determined time. At the end of that time period, the decoder will have been loaded with new "session keys" incompatible with the "session key" used to encrypt the data on the recorded tape.

Two-Way Interactivity

The conditional access system also allows for two-way interactivity on cable, fiber or possibly in wireless transmission systems through a PSK RF return data path. The 42 bit return data from the decoder is at a 45Kb/s data rate in a short, pre-defined data packet lasting 1.4 msec. The data is CRC error protected and can be encrypted with a downloaded encryption key. Decoder "status" information, including memory contents, channel tuned, etc., can be transmitted from the decoder to the control computer via polling techniques. This method allows for channel monitoring if required. Additionally, decoders can initiate transmissions for pay-per-view authorization requests, opinion poll responses, requests for information or merchandise, or to request control sequences from a service provider in a video-on-demand system. These operations must be initiated by a subscriber through inputting a proper PIN (Personal Identification Number) which the decoder verifies before transmission of the request. PIN numbers can be changed only through downloading a new number from the control computer at the transmission site.

Such decoder initiated transmissions rely on the short message length and a contention "blind-aloha" methodology to insure that messages do not collide and are received by the control computer. The decoder always expects a response from the transmission center via inband data whenever it initiates a two-way transmission. If the decoder receives no response within a random-per-decoder time period, the decoder assumes that a "data collision" occurred and re-transmits after another random time delay. This assures that all decoders requesting services via two-way interactivity receive responses within a short period of time. Actual average operational response times in large cable TV systems is under 1 second.

CONCLUSION

A new line-shuffling system for PAL, SECAM or NTSC has been introduced which has the benefits of hard renewable digital security and multi-level encryption without the cost and difficulty of full digital compression and transmission. The system is compatible with existing analog transmission techniques for broadcast, microwave, satellite, fiber and cable applications. The system is also compatible with existing subscriber receiving equipment since the encoded signal is essentially standard NTSC, PAL or SECAM. Performance of the system exceeds that of line cut-and rotate, sync suppression or other analog methods and a novel memory management technique is used to reduce memory requirements and cost. Photographs of the encoded video are shown in Fig 11.

REFERENCES

Department Of Trade And Industry - Specification of Television Standards for 625-Line System I Transmissions in the United Kingdom

Rohde & Schwarz - CCIR and FCC TV Standards

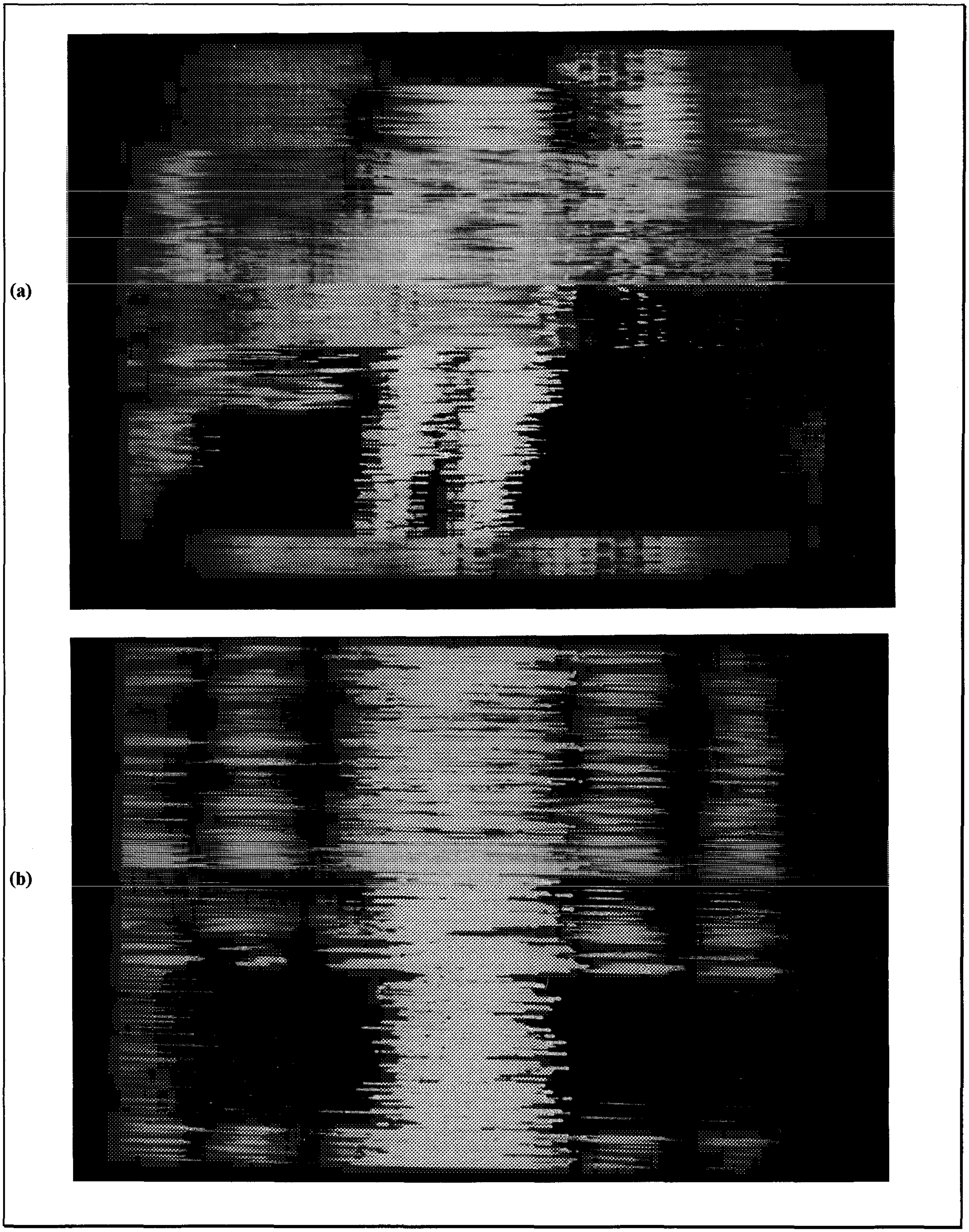


Fig 11 Actual TV Screen Images (a) 32 Line Shuffle (b) 128 Line Shuffle