Encryption with Statistical Self-Synchronization in Synchronous Broadband Networks

Oliver Jung and Christoph Ruland

University of Siegen, Institute for Data Communications Systems,
Hölderlinstraße 3, D-57068 Siegen, Germany
{jung, ruland}@nue.et-inf.uni-siegen.de
WWW home page: http://www.nue.et-inf.uni-siegen.de

Abstract. Most of the data transmission networks used today are based on the technology of the Synchronous Digital Hierarchy (SDH) or Synchronous Optical Networks (SONET) respectively. However rarely, they support any security services for confidentiality, data integrity, authentication or any protection against unauthorized access to the transmitted information. It is the subscriber's responsibility to apply security measures to the data before the information is passed on to the network. The use of encryption provides data confidentiality. This, however, requires consideration of the underlying network technology. The method described in this paper allows the use of encryption in broadband networks. The advantages of this method are the transparency of the encryption applied to the signal structure and signal format, and the automatic resynchronization after transmission errors. The used mode of operation, is called "statistical self-synchronization", because the synchronization between encryption and decryption is initiated by the presence of a certain bit pattern in the ciphertext, which occurs statistically. An encryption device, designed for SDH/SONET-networks with transmission rates of 622 Mbit/s, is to be presented.

Keywords: Broadband Networks, SDH/SONET, Confidentiality, Cryptography, Encryption, Modes of Operation, Self-Synchronization

1 Introduction

The increased requirement of bandwidth capacity over the last years has prompted for the development of efficient, digital transmission systems. Modern transmission systems provide higher transmission rates with a better bandwidth/cost ratio. SDH or SONET are the technologies that do not only fulfill this demand, but also offer the possibility of enhanced network management, controllable quality of service and simple multiplex structure. They are based on common international standards. One specific property of these types of networks is the supply of one central clock for all of the network components, like multiplexers or cross-connects. These types of networks are thus also called synchronous networks.

Ç.K. Koç and C. Paar (Eds.): CHES'99, LNCS 1717, pp. 340–352, 1999.

The whole range of information transmission, e.g. from phone call to video transmission, is handled via this type of networks. SDH/SONET is also largely used for Corporate Networks, which span long distances that cannot be controlled. The necessity for encryption technology is therefore especially important in this particular instance. There are four main reasons that cause difficulties in realizing an encryption technique for synchronous broadband networks:

- The maximum data transmission rate of 622 Mbit/s, and minimum of 155 Mbit/s respectively, and forecast 2.4 Gigabit/s or 10 Gigabit/s in the future
- Synchronous processing
- Bit slipping
- Complex management information

The synchronous transmission in these networks requires, that the encipherment is done with the same rate as network transmission rate. In Europe, there are no VLSI encryption components available today, supporting such high throughput rates. Therefore, it is necessary to use multiple encryption chips in parallel. The standardized and fixed frame structure does not allow for additional synchronization information of the crypto algorithm. We use the technique of statistical self-synchronization, which allows for the synchronization of the decryption algorithm even in the case of bit slipping. This mode of operation guarantees that the correct plaintext is computed at the receiver's side after an error-propagation has occurred. Bit slipping can occur if bits are deleted or additional bits are added by transmission components due to small differences of transmission lines data rates (jitter).

Bit slipping up to a certain extent is however covered by the synchronous network components. Even up to three bytes can positively or negatively be stuffed into one transmission frame. Therefore, bit- or byte slipping happens only if these thresholds are exceeded. This error needs to be taken into consideration even if the bit- or byte slipping probability is extremely small as also automatic switching of routing due to line drops can cause this.

The management information contained in the transmission frame requires, that specific parts of the frame are not to be encrypted as information which is important for network management and processed by the network components has to stay in plaintext. It is therefore required that management information bypasses encryption.

Chapter 2 shortly describes the structure of the STM-1-frame, whose payload is to be encrypted. Chapter 3 focuses on the modes of operation of block ciphers and shows that the standardized CFB- and OFB-mode are not sufficient for use in synchronous broadband networks. A new mode of operation, which we call statistical self-synchronization, is presented in chapter 4. Chapter 5 gives a layout of the realization of an encryption device for 622 Mbit/s with STM-4 interface or STS-12c respectively. This chapter furthermore gives additional information on selected implementation aspects. The presentation concludes with a summary and an outlook in chapter 6.

2 SDH and SONET

SDH and SONET are transmission systems with unlimited increasing rates that originally have been developed for the use in Wide Area Networks (WANs). Nowadays, they are however also used for Asynchronous Transfer Mode (ATM) in Local Area Networks (LANs). The standards for SDH and SONET contain not only the definitions of interfaces, e.g. transmission rates, formats and multiplexing techniques, but also recommendations for network management. SDH and SONET have similar characteristics. SONET is however mainly used in North America and is based on a standard frame, called STS-1, with a transmission rate of 51,84 Mbit/s whereas SDH, based on a standard rate of 155,52 Mbit/s is widely spread in Europe and Asia. The standard SDH frame, called STM-1, contains three concatenated STS-1 frames. Both transmission methods offer the basis for international data transmission and both support interfaces for existing as well as future techniques. The SDH-network interface is specified in ITU-T G.707 [5].

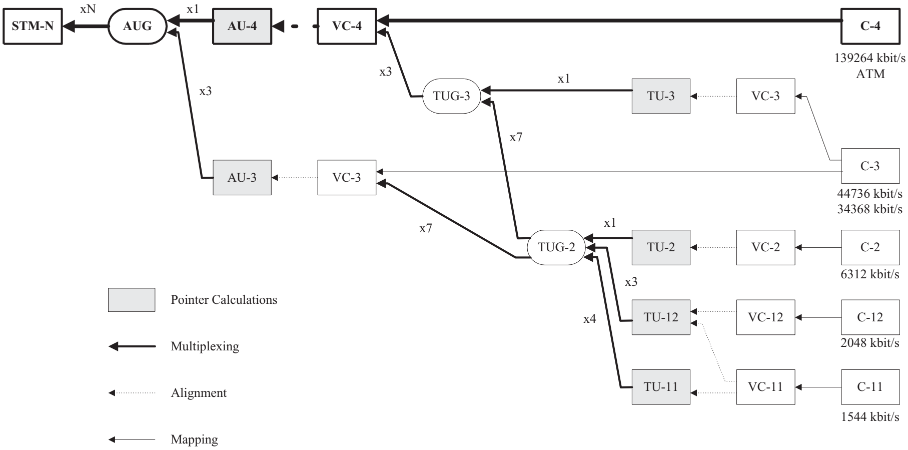

Fig. 1. SDH-Multiplex Hierarchy

Figure 1 shows, how the Synchronous Transport Module 1 (STM-1) can be constructed. The signals of the tributary systems that use SDH as a transport network are mapped into a standardized container (C-x). Through the adding of stuffing bits and control information, the so-called Path OverHead (POH), we get a Virtual Container (VC-x). The STM-1 module is constructed by adding a pointer which directs to the first byte of the Virtual Container and adding the Section OverHead (SOH) which consists of the Regenerator Section OverHead (RSOH) and the Multiplexer Section OverHead (MSOH) (see Figure 2).

The SDH is built in a modular way, whereby the STM-1 builds the basis for all higher transmission rates. Higher transmission rates are gained by bytewise multiplexing 4·n STM-1 frames (n = 1,2, etc) to one STM-4·n-frame. In this way, the next level of the hierarchy is the STM-4, which offers a capacity of 622 Mbit/s. A same structure is also defined in SONET, however, is called differently. The STM-1 corresponds to a SONET STS-3c and the STM-4 to an STS-12c frame.

The encryption device described in this paper processes STM-4 frames. The 622 Mbit/s data stream is internally split into four 155 Mbit/s streams, which are passed over to the encryption modules byte by byte.

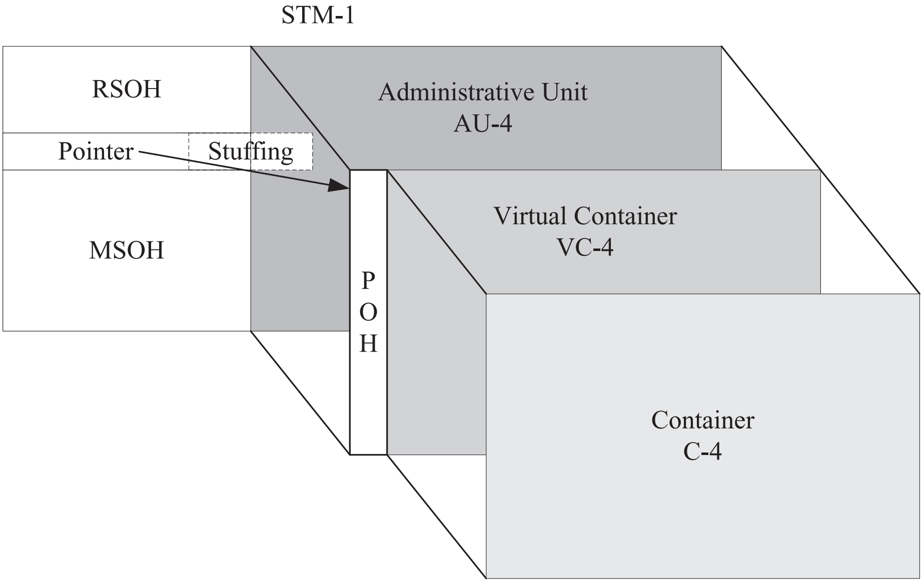

Fig. 2. SDH-Frame Structure

SOH and POH contain bytes for frame synchronization, signaling of the frame structure, service quality monitoring, path identification, alerts and alert responses. Some overhead bytes bypass the encryption and enter the next SOH or POH respectively, to be transmitted in plaintext. Others have to be re-calculated, e.g. parity bytes, which check the integrity of the payload. Again others have to be encrypted as they contain stuffed user information (stuffing). Consequently, it is required that the encryption modules are indicated which bytes are not to be enciphered.

3 Encryption

Encryption algorithms can be split into two groups. They either belong to the category of stream ciphers or block ciphers. Block ciphers encrypt blocks of bits in one step; a block length of 64 bit is usual. In contrast to this, only single bits or bytes are encrypted by stream ciphers. It would be beneficial to use stream ciphers for encryption of data units transmitted in broadband networks, as they reduce the delays in encryption devices. Cryptographically secure stream ciphers are however rarely known. The correspondingly required VLSI-chips, which in any case are needed for high-speed data rates, are not available. On the other hand, each block cipher can be turned into a stream cipher if it is used in an appropriate mode of operation. This approach is chosen here.

The four modes of operation, defined so far in ISO 10116 [4], are quite different in their properties regarding security, synchronization, error propagation, delay and throughput. (Note: We expect that this standard will be extended by new modes appropriate for high speed applications, e.g. the ATM Counter Mode, at the next release).

In order to turn block ciphers into stream ciphers they are used as key stream generators. Two modes do exist for this: either the Cipher FeedBack Mode (CFB) is used for self-synchronizing stream ciphers or the Output FeedBack Mode (OFB) is taken for synchronous stream ciphers.

3.1 The CFB-Mode

Encryption in the CFB-mode is achieved by XOR-ing the plaintext with the output of a key stream generator. The key stream is generated by the block cipher E_K, whereby K is a secret key. The input data of the algorithm is buffered in an input shift register. Since the last revision of ISO 10116, this input shift register can also be bigger than the block length of the block cipher in order to enable parallel encryption units for the high-speed generation of the key stream. In a standard case, n bits of the ciphertext are fed back into the input shift register, i.e. if n bits of the generated key stream are used for the encryption of n bits of the plaintext. Adjustments of the word formats could require the stuffing of the fed back ciphertext. This stuffing is explained in ISO 10116 in detail. Therefore, the definition of the CFB-mode in ISO 10116 is a bit more complicated than shown in Figure 3.

The CFB-mode offers the huge benefit of self-synchronizing. If a synchronization error occurs by erasing or adding a ciphertext unit of n bits length (corresponding to an n-bit slipping), the de-crypting side only generates wrong plaintext until the defect ciphertext units are shifted out of the input shift register. The same behavior occurs, if bits have been modified during transmission.

It however turns out that the implementation of the CFB-mode requires a very high encryption throughput rate. Assuming that n-bit by n-bit of plaintext are to be enciphered, then a complete input block needs to be encrypted in order to gain n-bit of cipher text. If V is the throughput rate of the block cipher implementation, the effective encryption rate, with which plain text in CFB-mode can be encrypted, applies as follows:

$$\nu = V \cdot \frac{n}{64}$$

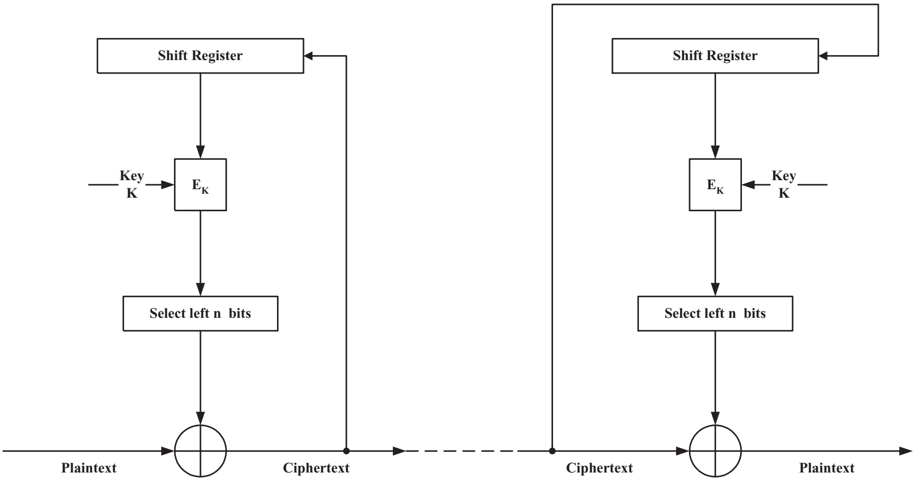

Fig. 3. CFB-Mode

If n = 1 is selected to receive a self-synchronization even in the case of bit slipping, then an encryption capacity of approximately 40 Gigabit/s per transmission direction is required to encrypt a STM-4-interface of 622 Mbit/s with a payload of approximately 600 bit/s. The CFB-mode therefore cannot be used in this way for broadband networks.

3.2 The OFB-Mode

The OFB-mode, in contrast to the CFB-mode, does not feed back the ciphertext into the input shift register, but the generated key stream. In this way, a complete output block of the key stream can be XORed with the plaintext for encryption, even if this is achieved only n-bit by n-bit (see Figure 4). The effective encryption rate therefore equals the encryption rate V of the key stream generator. A simplified description has been chosen once again, because the feeding back of smaller units, as well as adjustments of word formats are considered in more detail in ISO 10116.

The OFB mode offers the benefit of a high data throughput but not a self-synchronization. Therefore this type is also called a synchronous stream cipher. The fact that the transmitted cipher text is not used for the generation of the key stream means that the cryptographic synchronization is completely lost, and also cannot be recovered after the occurrence of synchronization errors. On the other hand, no error propagation happens if bits have been modified during transmission.

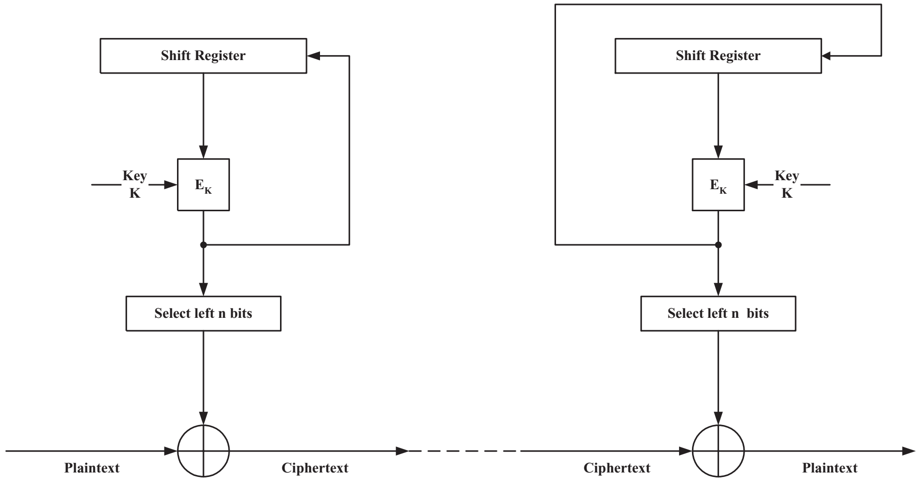

Fig. 4. OFB-Mode

4 The Statistical Self-Synchronization

The two described stream cipher modes of operation of block ciphers show big differences in their properties. The CFB is self-synchronizing, but only offers a low data throughput and error propagation. The OFB in contrast is not self-synchronizing, but rather shows no error propagation as well as a higher encryption rate.

The optimal solution would therefore be to combine the properties of both modes of operation. This is succinctly done by a new mode of operation, which we call statistical self-synchronization.

The statistical self-synchronization switches from one mode of operation to the other, and back whereby synchronization is reached between encryption and decryption by using the CFB-mode. OFB-mode is used between the synchronization phases. Loss of synchronization occurs in case of bit- or byte slipping. In order to re-synchronize, both sides need to be switched to CFB mode. The encryption and decryption are kept in CFB mode unless the input shift registers are filled with a complete block of ciphertext. This has to be identical on both sides. The content is used as a new starting value whereby OFB-mode is re-used afterwards (see Figure 5).

The decryption side, however, can not recognize when the synchronization has been lost. Both sides search for a fixed bit pattern in the ciphertext, as there is no additional communication capacity between the encryption and decryption entities to signal a switch in modes. This bit pattern occurs in a statistically distributed way in the ciphertext. Once the pattern is found, both sides switch to CFB-mode. The length of the bit pattern defines the probability of the synchronization and needs to be chosen in relation to the probability of bit slipping. The content of the bit pattern can be selected randomly, as all bit patterns of

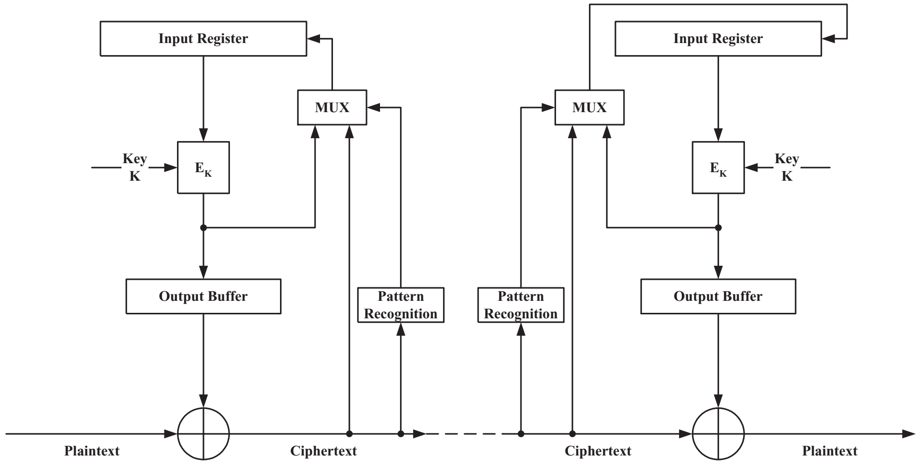

Fig. 5. Statistical Self-Synchronization

a fixed length are equally probable in the ciphertext. A bit slip causes a loss of synchronization, because the OFB mode is used between the synchronization phases. Encryption and decryption are out of synchronization till the bit pattern occurs in the cipher text. On the other hand, a switch into CFB-mode is achieved even in the case that no synchronization loss has occurred. This is the reason, why we call this mode of operation ”statistical self-synchronization”.

It should be emphasized again, that the bit pattern is generated by the encryption process itself as result of the encryption of the plaintext. No additional bandwidth is necessary to signal the synchronization start, or re-synchronization start, respectively.

A switching to the slower CFB mode implies for the encryption that during the operation in OFB-mode, as many key stream blocks need to be stored in the output buffer as are necessary to encrypt the plaintext during the next synchronization phase. Therefore, the encryption rate in OFB-mode must be higher than the transmission rate.

During a synchronization phase, another synchronization is not to be initiated. Therefore the bit pattern recognition is switched off during the synchronization process.

5 Implementation

Figure 6 shows the design of the encryption device. There are two essential components: the line interface, and the encryption and decryption module, respectively.

It is the task of the line interface to convert the 622 Mbit/s data stream into a format, which can be handled by the encryption/decryption module. During this process, plaintext and ciphertext are processed separately, i.e. different components are used (Red-Black-Separation). Otherwise it can happen that plain text could end up in the encrypted data stream due to malfunctioning of the interface components, or due to cross talking.

A direct connection between plaintext and ciphertext side exists only for the overhead, for different network alerts and for the high frequency reference clock, which are bypassed transparently.

An additional signal is passed together with the data stream from the line interface to the encryption/decryption module, in order to indicate which bytes do not have to be encrypted/decrypted as these positions contain overhead information.

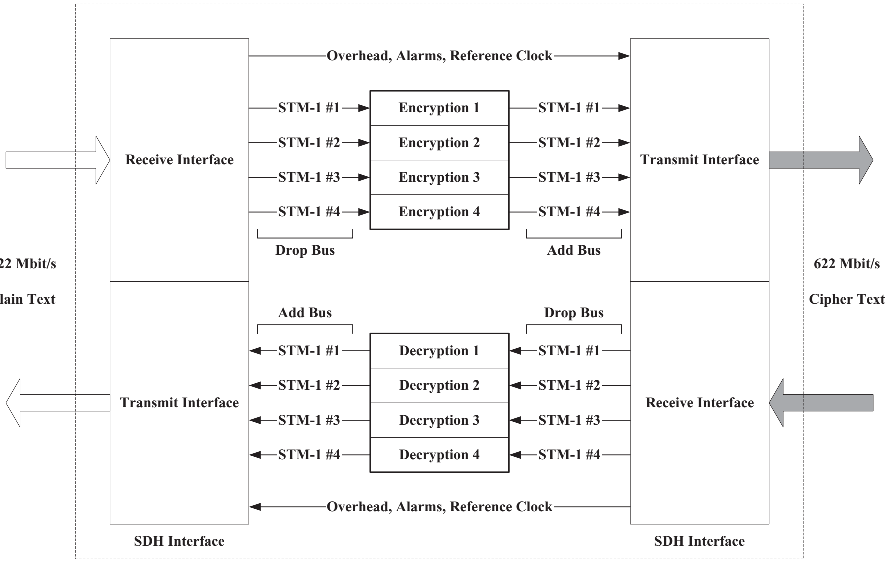

Fig. 6. Design of the Encryption Device

The SDH frame structure, which is supported by the encryption device, is shown in bold in Figure 1. It is a VC-4, which has been mapped into one STM-1 frame. One STM-4 frame consists of four STM-1 frames. This structure has been selected, as it is the most flexible one, as all tributary signals are transmitted in the VC-4. It is also used in ATM networks using SDH on the physical layer. The path via VC-3 and AU-3 is not very common in Europe and only serves as an adjustment to SONET.

5.1 Implementation of the Line Interface

The line interface has an optical input and output, as an electrical transmission is hardly possible at such high bit rates. The input signal is changed to an electrical signal and paralleled. This is followed by a demultiplexer which splits the STM-4 frame into four STM-1 frames and processes the STM-4 Section Overhead. The SOH bypasses the encryption and is reassembled at the transmitter side. There exist certain bits that have to be recalculated at the transmitter side. For each byte position of the SOH it is decided whether the received byte or the re-calculated byte is forwarded to the transmitted SOH (see Figure 7).

The four STM-1-frames are bytewise passed on to the encryption or decryption. The high frequency part is designed in ECL technology, whereas TTL technology is used for the other (approx. 20 MHz) one.

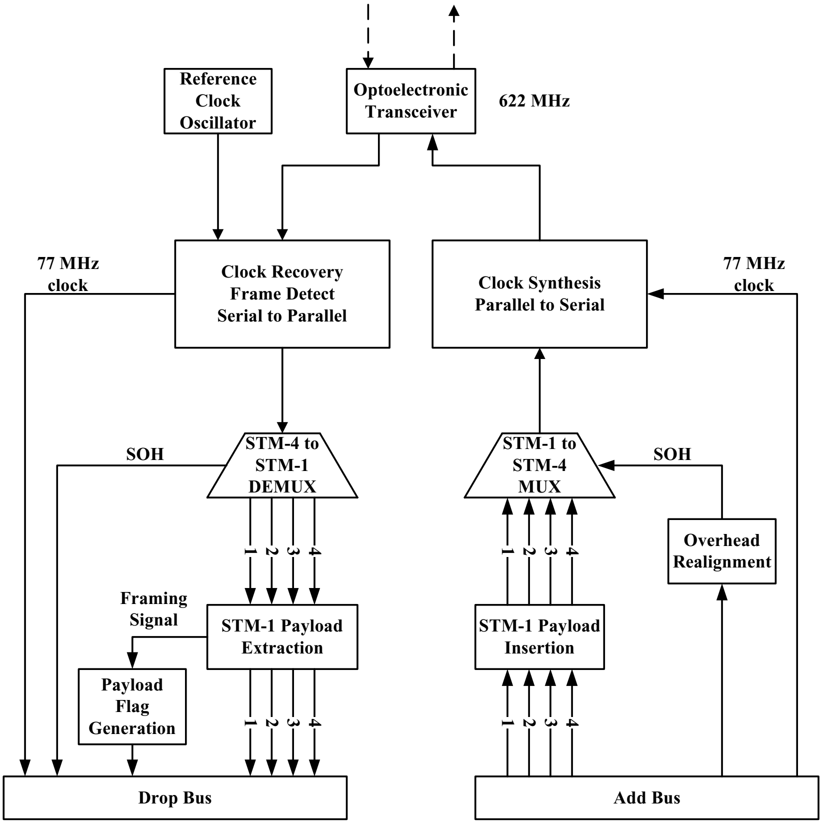

Fig. 7. Implementation of the 622 Mbit/s- Line Interface

No encrypted data is to be placed at byte positions that are used for overhead (SOH and POH). Encryption is therefore switched off during the presence of overhead data, i.e. overhead data passes the encryption in plaintext. The overhead bytes are recognized by analysis of the frame signal, evaluation of the pointer, and the counting of the byte positions.

Figure 8 shows the method for overhead bypassing. The SOH processing component has an interface that outputs the overhead in a serial form. The component supports receive clock, transmit clock, and a frame signal that indicates the overhead positions. It is required to synchronize the overhead on both sides. Two FIFOs are used for synchronization, which store the overhead, alternating frame by frame. Both FIFOs are used in an alternating way as synchronization is to be guaranteed even in the case of errors, i.e. interrupted overheads due to line drops. Each FIFO is reset after the overhead of a frame has been read out.

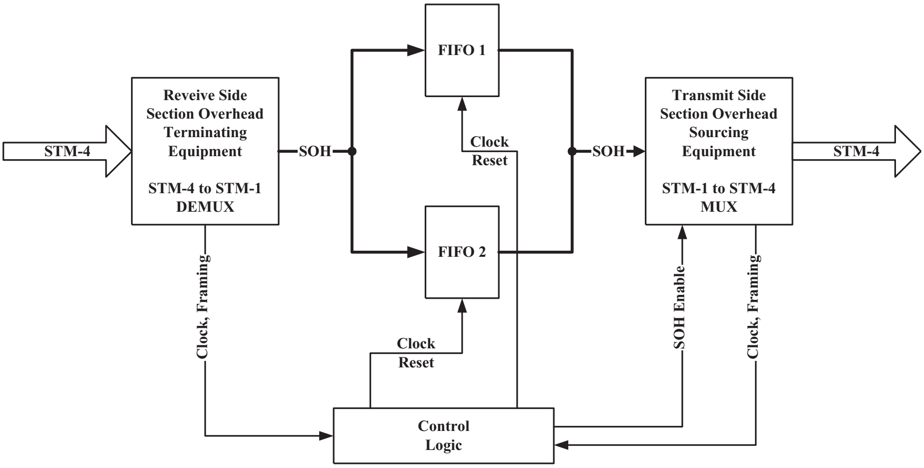

Fig. 8. Overhead-Bypassing

5.2 Implementation of the Encryption

A DES-encryption chip that offers an encryption rate of 160 Mbit/s is used for the key stream generation. DES is a block cipher with a block length of 64 bit [3]. The alternating mode that works with the double key length of 128 bit or 112 relevant bit respectively [2] is used. Each of the STM-1-data streams is encrypted separately and FIFOs are used as input as well as output registers. FIFOs are used for different reasons. Encipherment has to be done synchronously to the data transmission rate of 155 Mbit/s. The generated key stream has to be buffered as encryption does not work with this rate. In addition to this, the key stream generation works slowly (in CFB-mode) during the synchronization phase. For this reason, enough key stream bits have to be accumulated (in OFB-mode) during the encryption process so that there are sufficient key stream bits available for the then occurring synchronization phase. Therefore one encryption chip is not fast enough. DES-components are necessary which read alternate the input blocks from the input-FIFO. Thus, the input-FIFO has to be larger than one input block (see Figure 9).

In a standard case, i.e. not during synchronization phases, the encryption works in OFB-mode. A synchronization phase works like follows. The pre-defined bit pattern that triggers self-synchronization is constantly searched in the ciphertext. If this pattern is found, the multiplexer is switched over. The encryption works now in CFB-mode. Consequently, ciphertext is written into the Input FIFO unless a complete block has accumulated (64 bits). In the meantime, the plaintext continues to be XORed with the key stream bits which have been generated upfront in OFB-mode and which have been buffered in the active FIFO3 or FIFO4. This is continued until the generated and encrypted block in CFB-mode is available in the one FIFO that was not the last active one (this can be FIFO4 or FIFO3). Then the FIFOs are switched. The plaintext is now XORed with the key stream that has been generated in the CFB-mode. This process provides self-synchronization. The FIFO, active so far, is now reset and is prepared for the next synchronization phase. Please note, that the whole process is by far more complex if the layout is designed for a synchronous processing. Switching from OFB-mode to CFB-mode has to be done at the same bit position on the transmitter side and receiver side. Signal propagation and delay times have also to be considered. The fed back ciphertext bits can, for example, only be taken from the transmitted ciphertext after a certain period of delay. Furthermore, it is necessary to generate an additional key stream block in OFB-mode and store it in the new FIFO (directly behind the one in CFB-mode) before the output FIFOs can be switched. The bit pattern recognition that is required for the next upcoming bit pattern check, can only be released, once the FIFO has sufficiently been filled.

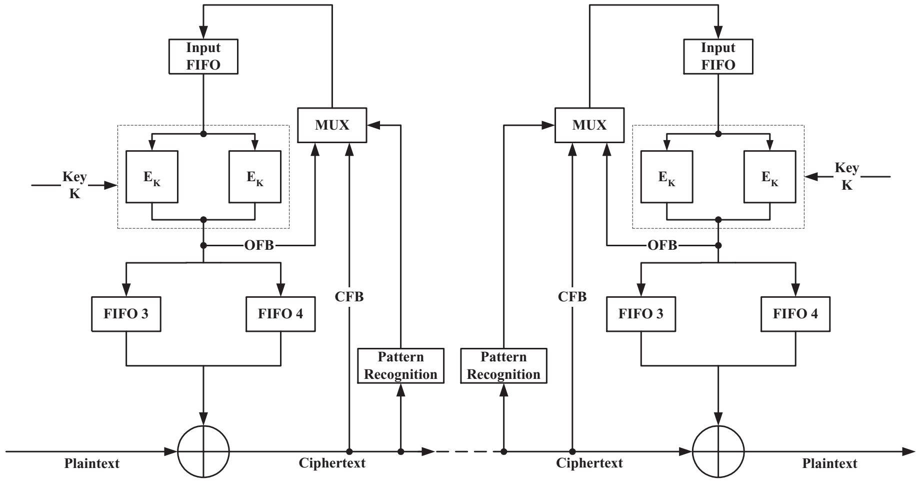

Fig. 9. Realization of the Self-Synchronization

There is an alternative concept for the realization of the statistical self-synchronization, which uses two separate encryption modules. Each of them contains two DES chips. The usage of two complete encryption modules has the benefit that the switching between the two different modes of operation is less complicated. One of two encryption modules works in OFB-mode the other one is idle as long as the bit pattern does not appear. If the bit pattern has been found, the second encryption module is initialized using the ciphertext in the input shift register. After the generation of one output block, in CFB-mode, this encryption module is switched to the OFB-mode. Now, this encryption module becomes the working unit. The benefit of this concept lies in the simpler relationships between the FIFOs and the encryption components. On the other hand, two complete encryption modules are required which results in higher costs per device.

6 Summary and Outlook

It has been demonstrated that it is possible to use encryption technology in high-speed networks. Additional channel capacity for synchronization purposes is consequently not necessary. The line interface has been developed in cooperation with the Worcester Polytechnic Institute and is available. At present the realization of the encryption module is in progress.

In addition to the used STM-4 frame which has been described in this paper, there exists another chained STM-4-frame, the so-called STM-4c. The STM-4c does not have a container with multiplexed four STM-1-frames but consists of one VC-4c-container which offers a transmission capacity of 600 Mbit/s. Currently we are working on an interface for the STM-4c. The challenges with the realization of the line interface for the STM-4c lies in the premise that there are no standard chips available which support the STM-4c interface. This means that standard components need to be complemented by programmable logic components (FPGA) in order to realize a STM-4c interface, e.g. to calculate the parity byte over the VC-4c. The encryption can be done through the use of the same modules that will be used in the device with the STM-4 interface.

References

1. ATM Forum: ATM Security Specification, Version 1.0, Final Ballot. January 1999
2. CE Infosys: CE99C003B Technical Reference (Priliminary). 1997
3. ANSI X3.92: Data Encryption Algorithm Standard, 1981
4. ISO/ICE: ISO 10116, Modes of Operation for an n-bit block cipher algorithm, Revision 1997
5. ITU-T Recommendation G.707: Network node interface for the synchronous digital hierarchy (SDH), 3/96
6. Schneier, Bruce: Applied Cryptography, 2nd Edition, John Wiley & Sons, 1996