RESEARCH ARTICLE

Error Burst Analysis of a Statistical Self-Synchronizing Mode of Block Ciphers

Howard M. Heys

Department of Electrical and Computer Engineering
Memorial University of Newfoundland
St. John's, Newfoundland, Canada
Email: hheys@mun.ca

ABSTRACT

In this paper, we investigate the burst error properties of the block cipher mode of operation referred to as statistical cipher feedback (SCFB) mode. SCFB was introduced as an efficient method of providing physical layer security by configuring a block cipher to operate as a self-synchronizing stream cipher. The self-synchronization property of a stream cipher allows the system to recover from data losses that might occur in the communication channel. The research presented in this paper investigates the properties of the post-decryption output of the cipher in response to bit errors in the communication channel. In particular, the probability distribution of the length of error bursts is investigated through simulation and a theoretical analysis is given for an upper bound on the probability of a burst longer than a given length. The results can be used to determine appropriate selections for parameters of the cipher mode and the specifications of the communication system, such as the error correction scheme and packet size.

KEYWORDS

block cipher; mode of operation; self-synchronizing stream cipher; burst errors

Original, unreviewed submission to SCN . . .

1. INTRODUCTION

Statistical cipher feedback (SCFB) mode, proposed in [1], can be used as a physical layer encryption scheme to configure a block cipher to process plaintext data as a stream of bits, rather than in fixed size blocks. Because the mode is self-synchronizing, the system can recover from the loss of data in the channel (eg. bit slips due to timing errors), whereas for other conventional block cipher modes, such as cipher block chaining (CBC) or counter mode [2], recovery from data loss is not readily possible. Further, although SCFB mode is modified from the conventional mode of cipher feedback (CFB) [2], it is shown in [3] to be much more efficient to apply than CFB, resulting in much higher throughputs for a hardware based implementation of a block cipher such as the Advanced Encryption Standard (AES) [4]. In fact, it can be shown that SCFB mode and the derived mode of pipelined SCFB (PSCFB) [5][*] can approach efficiencies of 100%. In contrast, conventional CFB mode, when implemented to recover from any size of data loss in the channel, has an efficiency of less than 1% when used with AES.

In [3] and [5], the error propagation characteristics of SCFB and PSCFB were analyzed. The analysis focused on a simple metric, referred to as the error propagation factor

[*] PSCFB mode was proposed to allow statistical self-synchronization to approach the throughput of a fully pipelined hardware implementation of a block oriented cipher mode like counter mode.

(EPF), by considering both simulations and theoretical analysis. EPF is defined as the average number of bit errors following decryption of an SCFB/PSCFB system, in response to a single channel bit error. While informative, this metric does not capture all the necessary characterizations of errors that occur post-decryption. In this paper, we investigate the burst error properties of SCFB mode, through simulation and analysis. Specifically, we examine the probability distribution of the length of a burst of errors, where the burst is generated following decryption of ciphertext affected by an individual channel bit error. In our work, we define the length of a burst to be the number of bits in the span from the first to the last bit error in the recovered plaintext resulting from the individual bit error in the ciphertext. As well, we develop a theoretical upper bound on the probability that the error burst exceeds a given length.

2. BACKGROUND

SCFB mode configures a block cipher to operate as a stream cipher by combining two block cipher modes: counter mode[†] and CFB mode. Stream ciphers encrypt plaintext data at a transmitter by XORing with the bits from a keyed pseudorandom sequence, referred to as the *keystream*. The resulting ciphertext is decrypted at the receiver by XORing the ciphertext bits with a keystream identical to the one used in the encryption process. The basic concept of SCFB is have the block cipher (eg. AES) to operate in counter mode, which generates keystream by processing a counter value, until a special short sequence of n bits, the *sync pattern*, is recognized in the ciphertext. Following the recognition of the sync pattern, the subsequent B bits (where B is the cipher block size) are collected and used as an initialization vector (IV) to reset the counter. Following the collected IV bits, subsequent bits are encrypted using counter mode until the sync pattern is again recognized in the ciphertext. This process is illustrated in Figure 1 where the "E" component represents a block cipher such as AES.

Since both the transmitter (which is executing encryption) and receiver (which is executing decryption) are able to observe ciphertext, if some ciphertext is lost in the communications channel, there is no need for the receiver to take special action since eventually a sync pattern will be received in the ciphertext and used in combination with the following IV bits to re-synchronize the two ends of the communication. Although loss of data in the channel will result in temporary loss of synchronization, thereby causing temporarily random output of the decryption, synchronization will be typically recovered at the next sync pattern (or certainly at a subsequent sync pattern). An illustration of the ciphertext stream associated with the mode is presented in Figure 2, which displays an interpretation of the ciphertext. The period from the end of one IV, until the next IV is all encrypted using counter mode as initialized by the IV preceding the period labelled as *scanning period*. During the IV period, both ends of the communication refrain from scanning for the sync pattern and must wait until the reinitializing of the counter mode before resuming scanning for the sync pattern.

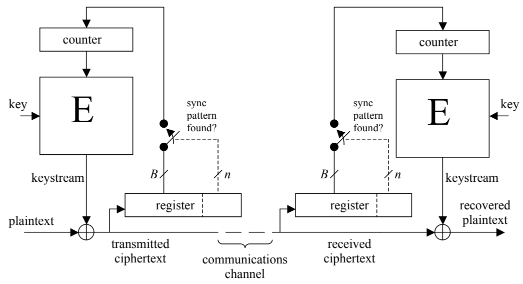

Figure 1. Conventional SCFB Mode Using Counter Mode

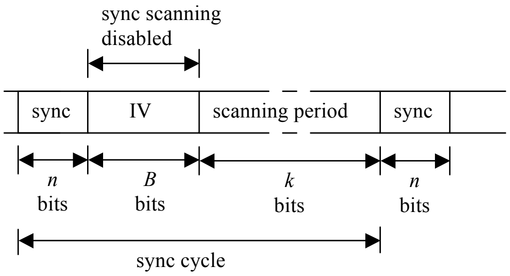

Figure 2. Synchronization Cycle for SCFB Mode

SCFB is capable of recovering from bit losses in the communication channel regardless of the number of bits lost. In comparison, using CFB to allow recovery from

[†] The original proposal for SCFB [1, 3] uses output feedback (OFB) mode in place of counter mode.

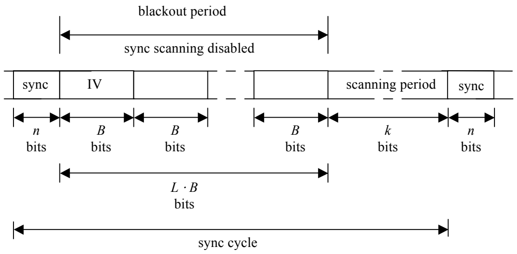

Figure 3. Synchronization Cycle for PSCFB Mode

any number of lost data bits requires a configuration such that to encrypt each plaintext bit uses the generation of an entire block of data from the block cipher. Hence, it is only possible to run CFB mode at a rate of $1/B$ times the throughput of the block cipher, which results in an efficiency of less than 1% for AES which uses a block with $B = 128$ bits. Here, we have defined efficiency as the fraction derived from the number of bits of ciphertext produced for every block cipher output divided by the block size, B. Hence, efficiency represents the throughput relative to the throughput possible using the block cipher to produce B bits of ciphertext for every B bit block generated at the block cipher output.

Sync patterns are assumed to be a modest number of bits, in the range of 6 to 16. Using a large sync pattern can result in a long delay before a resynchronization occurs following lost bits; a short sync pattern causes inefficiencies due to frequent resynchronizations [3].

PSCFB [5] is a more generalized form of SCFB. PSCFB mode is designed to allow for efficient application of statistical self-synchronization for implementations based on pipelined implementations of a block cipher. However, PSCFB can be thought as a simple extension of SCFB mode with an extra window of ciphertext following the end of the IV bits during which no scanning for the sync pattern is allowed. The window during which sync pattern scanning is not allowed is referred to the *blackout period*. The ciphertext stream of PSCFB is illustrated in Figure 3. The duration of the blackout period is defined to be a multiple of the block size, with the parameter L used to denote this multiple. In a pipelined implementation of the block cipher, L would represent the number of stages in the pipeline. Of course, when $L = 1$, then PSCFB degenerates simply to the conventional SCFB mode.

Using the more general PSCFB model, we can define the concept of a synchronization cycle of SCFB to consist of the sync pattern (n bits), blackout period ($L \cdot B$ bits, including the first B bits which contain the IV) and the scanning period. The duration of the scanning period is a random variable since the ciphertext stream may be assumed to be random. A sliding window is used during the scanning period to examine the ciphertext data for the sync pattern. Since the windows overlap and are therefore clearly not independent, the selection of the specific sync pattern affects the length of the scanning period. However, using many sync patterns, such as “100...00”, it can be shown that the duration of the scanning period can be approximated by the geometric distribution, with an expected length of $2^n - 1$ bits [3].

Although the first formal presentation of statistical self-synchronization appears to be given in [1], the mode was considered in other contexts previously. For example, reference to products making use of the self-synchronization concept appears earlier in [6]. As well, other authors have proposed similar self-synchronizing modes, such as optimized cipher feedback (OCFB) mode [7, 8]. Other papers have analyzed these modes [9, 10], but no previous work characterizes the post-decryption error behaviour of self-synchronization as is done by this paper.

3. MOTIVATION FOR THE ANALYSIS

In [3, 5], the implementation issues of SCFB and PSCFB mode are described, with structures given to ensure that an efficient implementation can be achieved. As well, the characteristics of the system in response to channel errors are presented. Specifically, the synchronization recovery delay (SRD) and error propagation factor (EPF) are examined through both simulation and analysis. SRD is defined to be the expected time from occurrence of a bit slip until the receiver has resynchronized. There is a strong dependence between the sync pattern size and the SRD, since the SRD is determined from the length of the scanning period which is random with an expected length that is exponential in the sync pattern size. EPF is defined to be the expected number of post-decryption bit errors caused by a single, isolated bit error in the communication channel and it is shown that EPF is quite close to $(n + B)/2$ for the range of n from 4 to 16 for

SCFB mode [3], while for PSCFB mode, EPF is roughly $(n+B)/2$ for large n and small L and somewhat larger for small n and large L [5].

In this paper, we investigate further the behaviour at the output of the SCFB/PSCFB system in response to single bit channel errors. This is motivated by the fact that, since it is an average, EPF often underestimates the effect of certain channel bit errors. Consider conventional SCFB (that is, $L = 1$) where an error occurs in the scanning period. For moderate and larger n (eg. $n \geq 8$), most errors will be of such a nature and most of these errors will simply result in one ciphertext bit error causing one bit error in the recovered plaintext, since plaintext is recovered at the receiver using the XOR operation between the keystream and the ciphertext data[‡]. Hence, for this specific, common instance, the error propagation is 1.

Consider now a second scenario where an error occurs during the sync pattern in the ciphertext transmitted through the channel. Since it is corrupted, the sync pattern is not recognized at the receiver and the receiver cannot resynchronize properly until the next sync pattern to appear in the ciphertext. This means that, since $L = 1$, the error propagation for this scenario is $1 + (k + n + B)/2$ where k is the length of the scanning period. This results from the fact that resynchronization will occur following the IV after the next sync pattern and, until resynchronization, half of the post-decryption bits are expected to be in error. As discussed in [3], k is a random variable, with an expected value of $E\{k\} \approx 2^n$. So we expect the error propagation to be about $1 + (2^n + n + B)/2$ for this scenario. For $n = 8$ and using AES as the block cipher where $B = 128$, the expected error propagation for this scenario is about 200 bits and can be much larger if k is a value above the mean. In comparison, EPF is substantially smaller, being only about 70.

For PSCFB, where $L \neq 1$, EPF can be quite large even for small n, since the sync cycle involves a blackout period of $L \cdot B$ bits. For example, a pipelined implementation of AES in PSCFB mode might have $L = 10$ [5] and the error propagation may be many hundreds of bits for some channel bit errors.

SCFB/PSCFB is designed for use at the physical layer of communication and, hence, at the receiver the decryption is performed prior to error detection and correction. Although EPF does give an idea of the average effect of errors on the outcome of the decryption, the distribution of the post-decryption bit errors in response to a channel bit error is an important consideration when investigating the behaviour of the system. For example, when communication protocols organize data into packets, protected using ARQ protocols which make use of CRCs to detect errors, a one bit error in a packet results in a retransmission, as can a burst of errors occurring in a packet. However, for SCFB/PSCFB, if a long burst of errors results from a single channel error, then this may cause the retransmission of many packets, significantly magnifying the effect of a single channel error. The probability of occurrence of such long post-decryption error bursts is not captured by EPF and in order to understand the resulting behaviour of the communication system that uses SCFB/PSCFB, knowledge of the post-decryption error burst distribution is needed.

4. DISTRIBUTION OF BURST LENGTHS

In order to investigate the probability distribution of the length of post-decryption error bursts, we have simulated SCFB and PSCFB systems. In our work, we consider the length of a burst to be the number of bits between (and including) the first post-decryption bit error and the last post-decryption bit error, due to a single ciphertext bit corrupted in the channel. Following the injection of isolated channel errors, we have counted the burst error lengths and examined the characteristics of the resulting experimental probability distribution. Parameters such as the sync pattern size, n, and the blackout period length in blocks, as indicated by L, are varied to investigate their influence on the distribution. All the simulation results presented in this paper use AES as the block cipher with block size $B = 128$ and involve the generation of 10^{10} bits of ciphertext data with the injection of about 10^6 channel bit errors.

4.1. Average Burst Length

The most obvious characteristic to consider is the average burst length. Table I presents the average burst length, found from simulations, for varying values of n and L. In

[‡] It is also possible, although unlikely, that a ciphertext error occurring during the scanning period causes a false sync pattern to appear, which is subsequently improperly recognized at the receiver as a legitimate sync pattern.

addition, for all scenarios, EPF is given in brackets. Several observations can be made from the data in the table.

Consider first variations in n. For SCFB (where $L = 1$), there is noticeable variation in average burst length, which tends to be shorter for larger n. For small values of n, the scanning period is much shorter and the probability that a channel error occurs during a sync pattern is increased. When the sync pattern is corrupted and missed by the receiver, a long burst of errors will result because the receiver will become unsynchronized with the transmitter until a subsequent sync pattern is detected. Although this is offset somewhat by the fact that the long bursts, while more frequent, are potentially shorter for smaller n (since the scanning period is shorter), with smaller n, often resynchronization can take several sync cycles (since false synchronizations at the receiver can frequently cause the next sync pattern to be missed). Hence, smaller n with frequent long bursts, results in an increase in the average burst length. Note that while varying n has notable effect on the average burst length, it does not significantly affect EPF.

As L varies, the average burst length is significantly affected and increases with increasing L. This is particularly true for smaller n. With small n and large L, the phenomenon of taking several sync cycles to resync when a sync pattern is corrupted is pronounced. Also, when a false sync pattern occurs as the result of a bit error, the blackout period is longer for large L and this can cause a missed legitimate sync pattern and prolonged delay before resynchronization. For large L such as $L = 14$ and $L = 16$, there is only a marginal increase in average burst length. In general, the EPF is only very slightly affected by varying L.

n	$L = 1$	$L = 2$	$L = 5$	$L = 10$
6	194.6 (73.6)	239.9 (78.3)	300.0 (93.5)	363.2 (119.4)
8	169.9 (72.6)	201.2 (73.7)	246.0 (77.6)	283.0 (85.5)
10	156.3 (74.1)	170.5 (74.4)	199.6 (74.7)	230.1 (76.8)
12	152.7 (75.6)	157.4 (75.7)	170.3 (76.4)	186.5 (76.6)
14	155.2 (77.8)	158.7 (78.9)	161.8 (78.8)	166.9 (78.7)
16	154.8 (77.8)	164.6 (82.5)	160.8 (80.2)	162.0 (80.0)

Table I. Average Burst Length (EPF in brackets)

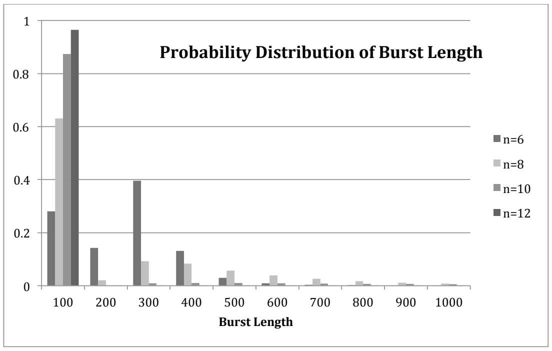

Figure 4. Probability Distribution of Bursts for Different n ($L = 1$)

4.2. Distribution of Burst Lengths

We have also investigated in detail, the probability distribution of burst lengths. This has been done by categorizing bursts lengths from the simulation results into ranges. We first explore this distribution for different values of sync pattern size, n, for conventional SCFB mode (i.e., $L = 1$). The resulting plots of burst length distributions for different values of $n = 6$, 8, 10, and 12 are presented in Figure 4. For all values of n, as burst length increases, its likelihood decreases. This is most dramatic for large values of n which have a large fraction of bursts being very short (due to the high probability of a channel error coinciding with the long scanning period which has an error propagation of only 1). Note that for all values of n the probability of burst lengths longer than 1000 bits becomes very small.

Next we explore the effect of varying the size of the blackout period, L, assuming fixed size $n = 8$. This is illustrated in Figure 5. From the graph, it can be seen that, for all values of L, the majority of bursts are short (less than 300). In fact, the majority of bursts are only of length 1 and are the result of a channel error occurrence during either the blackout period (excluding the IV) or during the scanning period (such that the error does not cause a false sync pattern at the receiver). It can also be observed that a secondary peak of burst lengths occurs around $2LB + 2^n$. This corresponds to scenarios where the channel error occurs during either the sync pattern or the IV. In such a scenario, synchronization will be lost at the end of the blackout period between the keystream of the transmitted ciphertext stream and the keystream of

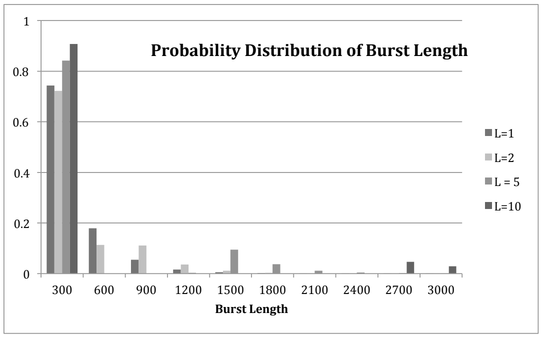

Figure 5. Probability Distribution of Bursts for Different L ($n = 8$)

the receiver and will not be corrected until the recovery of the next sync pattern and the completion of the following blackout period. Hence, the span from the occurrence of the channel error until the end of the burst will encompass roughly two blackout periods and a scanning period.

5. PROBABILITY OF LONG BURSTS

In this section, we consider the likelihood of long bursts in an SCFB/PSCFB scheme versus the system parameters of n, L, and B. Specifically, we consider (1) the value of the burst length which corresponds to a given fraction of longest bursts and (2) the probability of burst exceeding a certain length. Our investigations use simulations for (1) and simultations and theoretical analysis for (2).

5.1. Simulation Results for Long Bursts

We first consider simulation results to characterize long bursts. For this purpose, we define a long burst as having a burst length which is one of the longest 0.1% bursts. Figure 6 presents the length of burst above which represents the 0.1% longest bursts. These results are presented as a function of n, with $L = 1$ (i.e., conventional SCFB). It can be seen from the graph that there is a general trend of increasingly long bursts for larger n. For example, for $n = 16$, 0.1% of bursts are longer than 56286 bits (alternatively, 99.9 % of bursts are shorter than 56286 bits), while for $n = 8$, 0.1% of bursts are longer than 1679 (alternatively, 99.9% of bursts are shorter than 1679). The only exception to this general trend is that the long bursts for $n = 6$ are longer than for $n = 7$. In this scenario, for $n = 6$, the issues associated with frequent resynchronizing, which can result, for example, in several missed sync patterns as discussed in Section 4.1 cause longer bursts than for $n = 7$.

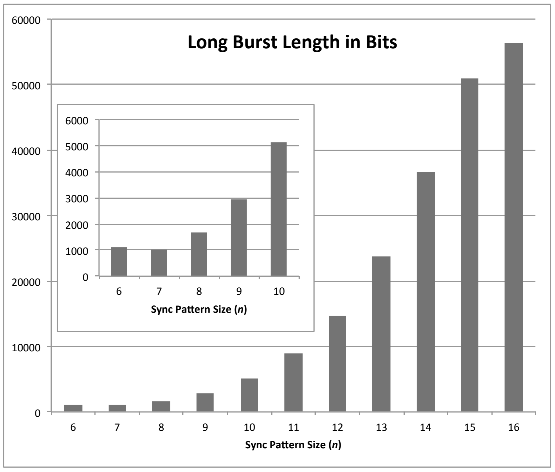

Figure 6. Long Bursts for Different n ($L = 1$)

Now consider the long burst limits for fixed $n = 8$ and 16 and varying L. When $n = 8$, as shown in Figure 7, long bursts (as defined by the longest 0.1% of bursts) increase as L increases. This is a reasonable expectation since long burst scenarios correspond to channel errors leading to post-decryption errors spanning a complete sync cycle (such as when a channel error occurs in the sync pattern or IV). Since the length of the sync cycle is on the order of $LB + 2^n$, long bursts will clearly increase with L. However, when $n = 16$, as shown in Figure 8, there is no discernable relationship between L and the length of long bursts, which vary between about 56000 and 62000 bits. In this scenario, the length of bursts resulting from channel errors leading to post-decryption errors spanning a sync cycle, are largely determined by the length of the scanning period (averaging about 2^{16}), which dominates over the blackout period length of $L \cdot B$ (which varies from 128 to 1280).

5.2. Analysis of Probability of Exceeding Given Burst Lengths

We now consider the probability of exceeding given burst lengths. Specifically, we examine the theoretical development of an upper bound on the probability of a burst greater than a given threshold. The analysis presumes an isolated, individual bit error occurs in the

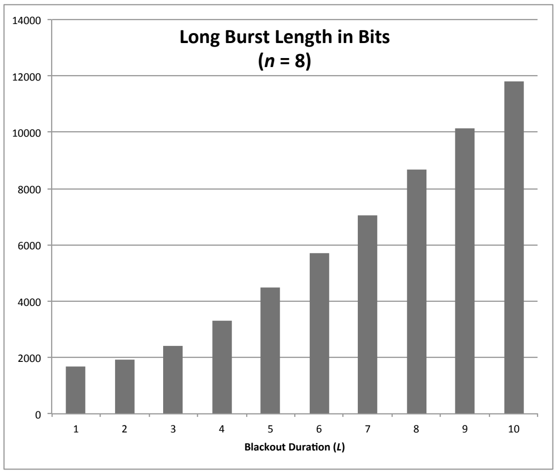

Figure 7. Long Bursts for Different L ($n = 8$)

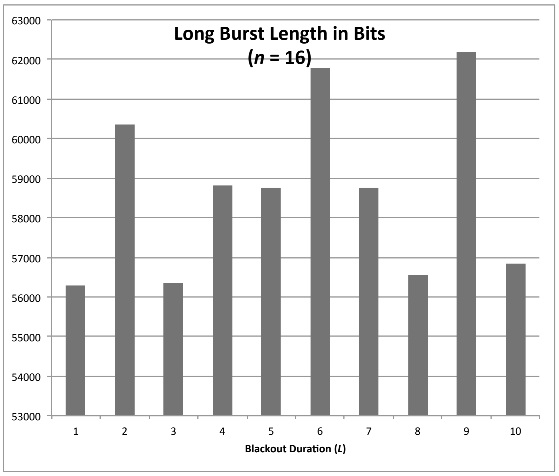

Figure 8. Long Bursts for Different L ($n = 16$)

communication channel and, as a result, a burst of one or more errors occurs at the output of the decryption process. In our analysis, we consider five possible cases for the occurrence of a bit error in the channel. The cases are defined based on the location of the bit error within the sync cycle of SCFB/PSCFB mode; the cases are listed in Table II and illustrated in Figure 9. Note that cases C3 and C4 are both divided into two sub-cases, which reflect that the effect of a bit error in the scanning period of the sync cycle can vary based on whether the error generates a false sync pattern. A false sync pattern is created, for example, for $n = 8$ with sync pattern “10000000”, if a pattern in the ciphertext of “10001000” occurs with the fifth bit having an error in the channel, thereby causing the “1” to become a “0” and the resulting sequence detected as a sync pattern at the receiver. Obviously, there are many such combinations of ciphertext data patterns and bit errors that can result in false sync patterns being detected at the receiver.

Case	Description
C1	Error in IV (B bits)
C2	Error during blackout following IV ($(L-1)B$ bits)
C3	Error in first $k-(n+LB)$ bits of scanning period (C3a) error does not create false sync (C3b) error does create false sync
C4	Error in last $n+LB$ bits of scanning period (C4a) error does not create false sync (C4b) error does create false sync
C5	Error in sync pattern

Table II. Cases for Bit Errors

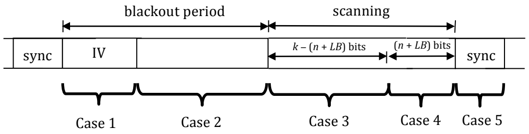

Figure 9. Illustration of Bit Error Cases

In our analysis, we make use of the following two lemmas.

Lemma 1: Let $S = \{1, 2, 3a, 3b, 4a, 4b, 5\}$ define the set of cases in Table II. As well, define N to represent the length of the post-decryption error burst given a 1 bit error in the communication channel and N^* to represent a threshold of interest for N. Then, an upper bound on the probability that N is greater than N^* is given by

$$P(N > N^*) \leq \sum_{i \in S} P_{max}(N > N^*|i) \cdot P_{max}(i), \quad (1)$$

where i represents one of the elements in set S, $P(N > N^*|i) \leq P_{max}(N > N^*|i)$ and $P(i) \leq P_{max}(i)$ with P_{max} representing an upper bound on the identified probability. Typically, we are interested in scenarios for which $N^* \gg 1$.

The Lemma simply follows from the total probability theorem. Its significance is that, using upper bounds for the occurrence of each case and the probability conditioned on each case, we are able to derive an upper bound on the probability that $N > N^*$.

Lemma 2: Assume that length of the scanning period in the sync cycle, k, follows the geometric distribution based on the probability that any n bits of ciphertext data has a probability of $1/2^n$ of matching the sync pattern. Hence, the probability that the number of bits in the scanning period exceeds a threshold θ is given by

$$P(k > \theta) = \left(1 - \frac{1}{2^n}\right)^{\theta+1} \tag{2}$$

for $\theta \geq 0$. For $\theta < 0$, $P(k > \theta) = 1$.

Consider now each case.

5.2.1. Case C1

For case C1, with a channel error occurring in the IV section of the sync cycle, a single post-decryption error will occur in the IV, followed by full sync cycle of corrupted bits following the end of the blackout period. Hence, the span of the error burst must encompass a full sync cycle of $k + n + LB$ bits plus the $(L-1)B$ bits of the blackout associated with the IV which experiences the channel error. Some number of bits, 1 to B, of the corrupted IV will also be included in the span, resulting in the length, N, of the burst error to be bounded as in

$$\begin{aligned} 1 + (L-1)B + k + n + LB &\leq N \\ &\leq B + (L-1)B + k + n + LB \end{aligned} \tag{3}$$

Of course, k is a random variable depending on the occurrence of the sync pattern within the ciphertext stream. Now, in order for the error burst length to exceed a value N^*, then $k > k_{min}$, where $k_{min} = N^* - n - 2LB$.

Hence, $P(N > N^*|\text{C1}) \leq P(k > k_{min}|\text{C1})$ where $P(k > k_{min}|\text{C1})$ is the conditional probability that the length of the scanning period exceeds the k_{min} bits given that the bit error has occurred in the IV portion of the sync cycle (i.e., case C1). This probability can be calculated using Lemma 2.

The probability of case C1 occurring is determined by the fraction of ciphertext that appears as IV in the mode. Hence,

$$P(C1) = \frac{B}{n + LB + E\{k\}} \tag{4}$$

where $E\{k\}$ represents the expected value of the length of the scanning period and is given to be $E\{k\} = 1/2^n - 1$ assuming the geometric distibution for k.

5.2.2. Case C2

Case C2, where it is assumed that the channel error occurs in the $(L-1)B$ bits of the blackout period following the IV, results in only one bit error after decryption. This occurs because, since scanning for the sync period is disabled at the receiver, the only affected bit after decryption is the bit corresponding to the channel error. Hence, $P(N > N^*|\text{C2}) = 0$ for $N^* > 1$ and this case can be ignored in the overall calculation of the probability of the upper bound of (1) for the scenarios of interest.

5.2.3. Case C3

Case C3 is the scenario which assumes that the channel error occurs in the first $k - (n + LB)$ bits of a scanning period for which $k > n + LB$. When an error occurs during the scanning period, two general phenomena can occur leading to two sub-cases: (a) the channel error does not cause a false sync to be recognized in the ciphertext stream and (b) the channel error results in a false sync being detected at the receiver.

In case C3a, the individual channel bit error results in one post-decryption bit error and, hence, as for case C2, $P(N > N^*|\text{C3a}) = 0$ for $N^* > 1$ and this case has no influence of the calculation of the upper bound for $P(N > N^*)$ for scenarios of interest.

In case C3b, the receiver, having falsely detected a sync pattern due to the channel error, will proceed with resynchronization by disabling scanning for LB bits, collecting IV and then resetting the counter value and entering again into the scanning period. In case C3b, the channel error causes the false sync to occur early enough in the correct scanning period that the receiver has finished the resynchronization due to the false sync pattern and is therefore back scanning for the sync pattern and able to detect the next correct sync pattern when it arrives. Hence, for case C3b, the length for the error burst satisfies

$$1 + n + LB + n + LB \leq N \leq k + n + LB \tag{5}$$

where $k > n + LB$. Letting $k = n + LB + k_0$, it can be seen that $N > N^*$ when $k_0 > k_{min}$, where $k_{min} = N^* - 2(n + LB)$, so that $P(N > N^*|3b) \leq P(k_0 > k_{min}|\text{C3b})$, which can be calculated using Lemma 2.

Now, we must consider the likelihood of case C3b, $P(\text{C3b})$. In [3], it is noted that the probability of detecting a false sync due to a single channel bit error is upper bounded by $\frac{n}{2^n-1}$ assuming that the distribution of k follows the geometric distribution. As a result,

$$P(\text{C3b}) \leq \sum_{k=n+LB+1}^{\infty} P^*(k) \cdot \frac{k-(n+LB)}{n+LB+k} \cdot \frac{n}{2^n-1} \tag{6}$$

where the first term, $P^*(k)$, represents the probability that the channel error occurs in a sync cycle with a scanning period of length k, the second term represents the probability that the channel error occurs within the first $k-(n+LB)$ bits of a sync cycle with a scanning period of length k, and the third term represents the upper bound on the probability of a false sync pattern being created due to the channel error. It can be shown [3],

$$P^*(k) = \frac{n+LB+k}{n+LB+E\{k\}} \cdot P(k) \tag{7}$$

where $P(k) = (1-1/2^n)^k(1/2^n)$ and $E\{k\} = 2^n - 1$ as given by the geometric distribution.

Now noting that

$$\sum_{k=n+LB+1}^{\infty} \left(1-\frac{1}{2^n}\right)^k = 2^n \cdot \left(1-\frac{1}{2^n}\right)^{n+LB+1} \tag{8}$$

and

$$\sum_{k=n+LB+1}^{\infty} \left(1-\frac{1}{2^n}\right)^k \cdot k$$
$$= 2^n \cdot (n+LB+2^n) \cdot \left(1-\frac{1}{2^n}\right)^{n+LB+1}, \tag{9}$$

leads to

$$P(\text{C3b}) \leq \frac{n}{n+LB+E\{k\}} \cdot \left(1-\frac{1}{2^n}\right)^{n+LB}. \tag{10}$$

5.2.4. Case C4

Case C4 refers to scenarios where the channel bit error occurs in the last $n+LB$ bit of the scanning period of a sync cycle. As with case C3, there are two sub-cases.

Case C4a refers to the scenario where the channel error does not cause a false sync and, as a result, only one bit error occurs in the post-decryption stream of data and $P(N > N^*|\text{C4a}) = 0$ for $N^* > 1$, resulting in this case having no meaningful contribution to $P(N > N^*)$ for scenarios of interest.

Case C4b is substantially more complex and represents scenarios where the channel error results in a false sync pattern detected at the receiver. When a false sync is detected at the end of a scanning period, a resynchronization may occur at the subsequent proper sync pattern or, if the next sync pattern falls within the blackout period associated with the false sync, the next sync patten may be missed. Subsequently, another false sync may occur when the receiver erroneously detects a sync pattern in the ciphertext, which in fact was supposed to fall in a blackout period. This may result in another missed proper sync pattern and this behaviour may repeat many times.

When a sync pattern is missed, the corresponding sync cycle must have a scanning period of less than $n+LB$ bits, otherwise the proper sync pattern is guaranteed to not fall within the blackout period of the false sync detection. Hence, associated sync cycles which have the subsequent sync pattern missed must have $k < n+LB$ and overall the sync cycle length must be between $n+LB$ and $2(n+LB)-1$. Resynchronization will not occur until a proper sync does not fall into the blackout period of a false synchronization, which is only guaranteed to happen when $k \geq n+LB$ in a sync cycle. In addition, there are between $1+n+LB$ and $2(n+LB)$ bits before the scanning period of the first sync pattern and $k+n+LB$ bits from the start of the last unsynchronized scanning period to the point where the receiver properly resynchronizes.

Letting λ represent the number of times a proper sync pattern is missed following the first missed sync pattern, from the point of the channel error to the point of resynchronization, the error burst length, N, satisfies

$$\beta_{min} + \phi_{min} + k + n + LB$$
$$\leq N \leq \beta_{max} + \phi_{max} + k + n + LB, \tag{11}$$

where $\beta_{min} = n+LB+1$, $\beta_{max} = 2(n+LB)$, $\phi_{min} = \lambda(n+LB)$, and $\phi_{max} = \lambda[2(n+LB)-1]$ and k is the length of the scanning period in the last unsynchronized cycle. Note that, after the original sync cycle affected by the channel error, a legitimate sync pattern can only be missed at the receiver when the length of the scanning period is less than $n+LB$.

Based on the upper bound of N, it can be shown that $N > N^*$ implies that $k > k_{min}$ where

$$k_{min} = N^* - 3(n + LB) - \lambda \cdot (2(n + LB) - 1) \tag{12}$$

for a given λ. So, now $P(N > N^*|\lambda, \text{C4b}) \leq P(k > k_{min}|\lambda, \text{C4b})$ where $P(k > k_{min}|\lambda, \text{C4b})$ can be calculated using Lemma 2 for given values of λ. Considering all possible values for λ, we can derive $P(N > N^*|\text{C4b})$ using

$$P(N > N^*|\text{C4b}) = \sum_{\lambda=0}^{\infty} P(k > k_{min}|\lambda, \text{C4b}) \cdot P(\lambda). \tag{13}$$

This summation can be broken into two parts by considering the maximum value of λ which can be realized for values of N less than N^*. That is, the maximum number of sync cycles that can occur in N^* bits. We represent this value as Γ and note that

$$\Gamma = \lfloor \frac{N^* - (n + LB + 1) - (n + LB)}{n + LB} \rfloor. \tag{14}$$

Also, an upper bound on $P(\lambda)$ is calculated using

$$P(\lambda) \leq (1 - \alpha)^{\lambda} \tag{15}$$

where

$$\alpha = P(k \geq n + LB) = \left(1 - \frac{1}{2^n}\right)^{n+LB}. \tag{16}$$

As a result, we can now write $P(N > N^*|\text{C4b})$ as

$$P(N > N^*|\text{C4b}) = \Sigma_1 + \Sigma_2 \tag{17}$$

where

$$\Sigma_1 = \sum_{\lambda=0}^{\Gamma} P(k > k_{min}|\lambda, \text{C4b}) \cdot P(\lambda) \tag{18}$$

and

$$\Sigma_2 = \sum_{\lambda=\Gamma+1}^{\infty} P(k > k_{min}|\lambda, \text{C4b}) \cdot P(\lambda). \tag{19}$$

The sum Σ_1 can be computed using Lemma 2 and (15). An upper bound on the sum Σ_2 can be determined by considering that $P(N > N^*) \leq 1$. Hence,

$$\Sigma_2 \leq \sum_{\lambda=\Gamma+1}^{\infty} P(\lambda) = \sum_{\lambda=\Gamma+1}^{\infty} (1 - \alpha)^{\lambda} = \frac{(1 - \alpha)^{\Gamma+1}}{\alpha}. \tag{20}$$

Consider now the likelihood that case C4b occurs. An upper bound on this probability can be determined by averaging across all scenarios for the length of the scanning period, k, resulting in

$$\begin{aligned} P(\text{C4b}) \quad \leq \quad & \sum_{k=0}^{n+LB} P^*(k) \cdot \frac{k}{n + LB + k} \cdot \frac{n}{2^n - 1} \\ & + \sum_{k=n+LB+1}^{\infty} P^*(k) \cdot \frac{(n + LB)}{n + LB + k} \cdot \frac{n}{2^n - 1} \end{aligned} \tag{21}$$

where $P^*(k)$ is given by (7) and we have broken the calculation into scenarios where $k \leq n + LB$ and $k > n + LB$. This can be manipulated to become

$$\begin{aligned} P(\text{C4b}) \quad \leq \quad & \frac{n}{2^n - 1} \cdot \frac{1}{n + LB + E\{k\}} \cdot [\sum_{k=1}^{n+LB} k \cdot P(k) \\ & + \sum_{k=n+LB+1}^{\infty} (n + LB) \cdot P(k)]. \end{aligned} \tag{22}$$

Solving for closed form representations of the series inside the square brackets leads to

$$P(\text{C4b}) \leq \frac{n \cdot (1 - \alpha)}{n + LB + E\{k\}} \tag{23}$$

where $\alpha = P(k \geq n + LB)$ and is given by (16).

5.2.5. Case C5

The last scenario, case C5, refers to the occurrence of a channel error in a sync pattern. In this scenario, the behaviour of resynchronization at the receiver is very similar to case C4b. In this case, it is also quite possible to miss several sync patterns before resynchronization at the receiver due to sync patterns falling within the blackout periods of false synchronizations. Hence, we can write

$$P(N > N^*|\text{C5}) = \sum_{\lambda=0}^{\infty} P(k > k_{min}|\lambda, \text{C5}) \cdot P(\lambda) \tag{24}$$

where now $k_{min} = N^* - 2(n + LB) - \lambda \cdot (2(n + LB) - 1)$. This summation can be broken into two summations, Σ_1 and Σ_2 as in case C4b with now

$$\Gamma = \lfloor \frac{N^* - (LB+1) - (n+LB)}{n+LB} \rfloor. \tag{25}$$

Finally, the probability of this case occuring is straightforwardly given by

$$P(\text{C5}) = \frac{n}{n+LB+E\{k\}}. \tag{26}$$

5.2.6. Results of Analysis

In Table III, we present the theoretical bound and the corresponding simulation results for varying values of burst length thresholds (N^*), for $n = 8$ and $L = 1$, a typical scenario for conventional SCFB mode. Note that the theoretical result is an upper bound on the probability of the burst exceeding N^*, while the simulation result is an experimental determination of the exact probability that the burst exceeds N^*. There is good correspondence between the theoretical and experimental results, where, as expected, the experimental probability falls below the theoretical upper bound. As implied by the results, as the burst length increases, the probability (and its upper bound) decrease. For example, the probability that a burst exceeds 1000 bits is upper bounded to be about 0.030 and experimentally found to be about 0.016, while the probability that a burst exceeds 3000 bits is expected to be less than about 3.4×10^{-5} and is experimentally found to be 8.0×10^{-6}.

Results for a typical scenario for PSCFB mode with $n = 16$ and $L = 10$ are shown in Table IV. For this scenario, the theoretical upper bound is relatively tight, when compared to the experimentally determined probability. It is apparent, however, that there is not as dramatic a decrease in the probability as the length of the burst increases. For example, when $N^* = 20,000$, the probability is about 1.8×10^{-3}, while for $N^* = 200,000$, the probability has only decreased by an order of magnitude to about 1.2×10^{-4}. This occurs because this scenario is dominated by case C1 in Table II and the probability is therefore roughly proportional to ϵ, where $\epsilon = (1 - 1/2^n)^{N^*}$, since the burst length, for such a large n, is roughly proportional to the length of the sync cycle, which is dominated by the scanning period whose length is given by the geometric distribution. For $N^* \ll 2^n$, such as 20,000, ϵ is not much less than 1 and increasing N^* to 200,000, decreases ϵ by a power of 10, which means that, since ϵ is close to 1, ϵ is only decreased by an order of magnitude. As ϵ gets small (that is, $\epsilon \ll 1$ which occurs when $N^* \geq 2^n$), increasing N^* by an order of magnitude will have a much more dramatic effect.

N^*	Theoretical Upper Bound on $P(N > N^*)$	Experimentally Determined $P(N > N^*)$
500	1.633e-01	1.169e-01
1000	2.971e-02	1.5862e-02
1500	5.352e-03	2.091e-03
2000	9.792e-04	2.82e-04
2500	1.821e-04	4.5e-05
3000	3.441e-05	8e-06
3500	6.533e-06	1e-06
4000	1.247e-06	0

Table III. Upper Bound on Probability of Exceeding Given Burst Length for $n = 8$ and $L = 1$

N^*	Theoretical Upper Bound on $P(N > N^*)$	Experimentally Determined $P(N > N^*)$
20000	1.839e-03	1.839e-03
40000	1.355e-03	1.313e-03
60000	9.990e-04	9.45e-04
80000	7.362e-04	6.99e-04
100000	5.426e-04	5.19e-04
120000	3.999e-04	3.81e-04
140000	2.947e-04	2.93e-04
160000	2.172e-04	2.02e-04
180000	1.601e-04	1.51e-04
200000	1.180e-04	1.14e-04

Table IV. Upper Bound on Probability of Exceeding Given Burst Length for $n = 16$ and $L = 10$

Although the scenarios presented above result in reasonably tight upper bounds, there are scenarios for n and L, for which the upper bound is very loose and, therefore, not as effective as a measure for the probability of long bursts. This occurs, for example, for small n and large L, such as $n = 8$ and $L = 10$ where cases C4b and C5 dominate the theoretical upper bound and these cases are formulated with looser restrictions, resulting in the overall bound being quite loose.

6. CONCLUSION

In this paper, we have investigated the post-decryption error burst characteristics of SCFB mode and its derivative, PSCFB mode. Simulation results examining the burst error length distribution confirm the dominance of single bit error bursts, with only small probability for longer bursts. The burst distribution is characterized through simulations for various sync pattern sizes, n, and blackout durations, represented in terms of the number of blocks, L, which is also the number of pipeline stages. In general, larger n leads to more likelihood of shorter (1 bit) bursts, but the potential for longer bursts due to scenarios where loss of synchronization occurs. Increasing L also results in scenarios for which longer bursts are more significant.

Theoretical analysis is presented giving an upper bound on the probability bursts are longer than a given length. The appropriateness of this bound is confirmed through simulations and simulations are also used to explore the distribution of long bursts (defined as the longest 0.1% bursts) as a function of n and L. For $L = 1$, the length of long bursts increases with n and, for $n = 8$, the length of long bursts increases with L. However, for large n (eg. $n = 16$), there was found to be no correlation between L and long burst length.

The value of this paper is that through characterizing the burst lengths, it is possible to recommend values for n and L based on the requirements of the communication system. For example, for small L (eg. $L = 1$ as in a non-pipelined conventional SCFB system), using a modest size sync pattern, such as $n = 8$, will minimize the likelihood of long bursts. As another example, in a PSCFB system that will implement a pipelined architecture of the block cipher, it may be desirable to use a large sync pattern size (eg. $n = 16$) to minimize the frequency of resynchronizations and, in this scenario, there is little identifiable impact from the selection of the number of pipeline stages, L, on the long burst length and, hence, L can be selected based on the implementation constraints and the efficiency of the system. Another valuable outcome of this work is that, using the theoretical upper bound on the probability of bursts exceeding a given value, it would be possible to design and analyze the error detection and correction aspects of a communication system which uses SCFB/PSCFB in the underlying physical layer security.

REFERENCES

1. O. Jung and C. Ruland, “Encryption with Statistical Self-Synchronization in Synchronous Broadband Networks,” *Cryptographic Hardware and Embedded Systems (CHESS ’99), Lecture Notes in Computer Science (LNCS)*, vol. 1717, pp. 340-352, Springer, 1999.
2. William Stallings, *Cryptography and Network Security*, Pearson Prentice-Hall, 6th ed., 2013.
3. H.M. Heys, “Analysis of the Statistical Cipher Feedback Mode of Block Ciphers”, *IEEE Transactions on Computers*, vol. 52, no. 1, pp. 77-92, IEEE, 2003.
4. National Institute of Standards and Technology, *Advanced Encryption Standard*, Federal Information Processing Standards (FIPS) Publication 197, 2001.
5. H.M. Heys and L. Zhang, “Pipelined Statistical Cipher Feedback: A New Mode for High-Speed Self-Synchronizing Stream Encryption”, *IEEE Transactions on Computers*, vol. 60, no. 11, pp. 1581-1592, IEEE, 2011.
6. C. Brookson, “Cryptography Systems: Putting Theory into Practice”, *Information Security Technical Report*, vol. 2, no. 4, Elsevier, pp. 66-72, 1998.
7. A. Alkassar, A. Geraldy, B. Pfitzmann, and A-R. Sadeghi, “Optimized Self-Synchronizing Mode of Operation”, *Fast Software Encryption (FSE 2001), Lecture Notes in Computer Science*, vol. 2355, Springer, pp. 78-91, 2001.
8. K. Burda, “Modification of OCFB Mode for Fast Data Links”, *International Journal of Computer Science and Network Security*, vol. 7, no. 12, pp. 228-232, 2007.
9. O. Jung and C. Ruland, “Analysis of the Statistical Self-Synchronization Mode of Operation”, *Proceedings of ITG Conference on Source and Channel Coding (SCC)*, pp. 119-126, 2004.
10. K. Burda, “Resynchronization Interval of Self-Synchronizing Modes of Block Ciphers”, *International Journal of Computer Science and Network Security*, vol. 7, no. 10, pp. 8-13, 2007.