Analysis of a measure of correlation between two binary strings of different lengths

H. M. Gustafson

Centre in Statistical Science and Industrial Mathematics
Faculty of Science
Queensland University of Technology
Brisbane, Queensland, Australia

L. R. Simpson

Information Security Research Centre
Faculty of Information Technology
Queensland University of Technology
Brisbane, Queensland, Australia

J. Dj. Golić

School of Electrical Engineering
University of Belgrade
Bulevar Revolucije 73
11001 Belgrade, Yugoslavia

Abstract

The joint probability between two binary strings of different lengths has been demonstrated as a suitable basis for a correlation attack on the shrinking generator when the decimation probability is 0.5. As an extension of this, further computer simulations have been conducted to determine whether the joint probability can be used as a basis of similar correlation attacks on irregularly clocked shift registers, where the deletions occur independently with a fixed probability different from 0.5. These results show that as the decimation probability increases, the length of the known keystream required for the joint probability to be a useful measure of correlation must also be increased. Thus, the joint probability is shown to be a suitable basis for such correlation attacks, provided a sufficient length of the keystream is known.

Australasian Journal of Combinatorics 25(2002), pp.185–199

1 Introduction

The increasing use of computers and telecommunications networks has resulted in the need for secure methods for storage and transfer of data, particularly electronic data. Algorithms that transform a message, called the plaintext, into another form, the ciphertext, so that the original message is disguised are called ciphers. Persons with authorised access to the message have some secret knowledge called the key, which allows the transformation to be reversed and the original message recovered. The message transformations are known as encryption and decryption, respectively. The key is usually the initial state of a pseudorandom number generator. Each secret key used as input to the pseudorandom number generator corresponds to a longer pseudorandom output, the keystream. If the same key is used for the keystream generators on both the sending and receiving ends of a transmission, then the same keystream will be produced by both.

Stream ciphers are encryption algorithms which encrypt a plaintext message one character at a time, under a time-varying function of the key. For binary plaintext, one of the most commonly used cipher systems is the binary additive stream cipher, where both the plaintext and keystream are sequences of bits and the ciphertext is formed by the bitwise modulo two addition of the two streams. A major advantage in using a binary additive stream cipher is that encryption and decryption can be performed by identical devices. Encryption and decryption using a binary additive synchronous stream cipher are illustrated in Figure 1.

Let $p(t)$, $z(t)$ and $c(t)$ denote the plaintext, keystream and ciphertext bits at time $t \geq 0$, respectively. Under encryption, the t^{th} bit in the ciphertext stream is formed as $c(t) = p(t) \oplus z(t)$, where $\oplus$ denotes addition modulo two. Similarly, under decryption, the t^{th} bit in the plaintext stream is formed as $p(t) = c(t) \oplus z(t)$. Note that if some plaintext-ciphertext pairs are known, then some of the keystream bits are also revealed, as $z(t) = p(t) \oplus c(t)$. Therefore, the keystream generator is the critical component in the security of this type of cipher system. The level of security provided depends on the apparent randomness of the keystream. Users need to be confident that unauthorised persons cannot gain any knowledge of the actual message from intercepted ciphertext. Even if an interceptor obtains a portion of the ciphertext and the corresponding plaintext they should not be able to determine

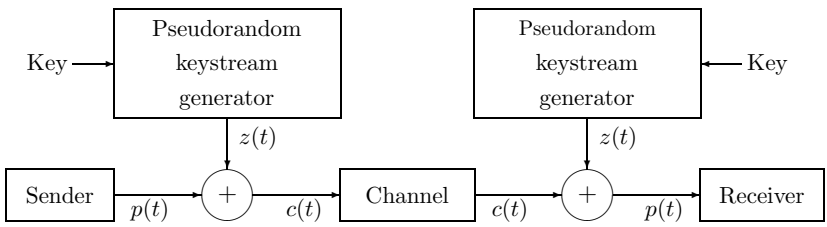

Figure 1: Encryption and decryption in binary additive stream ciphers.

either the whole keystream sequence or the secret key for the keystream generator in less time than by exhaustive search of the keyspace.

In analysing the security provided by a particular stream cipher, it is usually assumed that, except for the secret key, the entire cipher is known to the cryptanalyst. The cryptanalyst attempts to find weaknesses which can be exploited to recover the key from a known segment of keystream. Correlation attacks exploit statistical dependencies which exist between the keystream and underlying internal sequences. Originally, correlation attacks were performed on regularly clocked keystream generators ([4]). Irregular clocking was considered a means of avoiding susceptibility to these correlation attacks, and generators such as the shrinking generator [1], whose output can be considered to be the output of an irregularly clocked shift register, were proposed.

More recently, correlation attacks on irregularly clocked keystream generators have been proposed. These attacks are similar to the original correlation attack [4], but necessarily require a different measure of correlation. Measures of correlation suitable for use with irregular clocking, and attacks which use these measures are reviewed in [2]. The measures of correlation include Levenshtein and Constrained Levenshtein distances and probabilistic measures, and the corresponding embedding and probabilistic correlation attacks are outlined.

One measure of correlation, proposed in [3], is termed the joint probability. In [5], for the case of the shrinking generator, a normalised joint probability is shown to be a suitable basis for a correlation attack on the shrinking generator. For the attack, the keystream of the shrinking generator is viewed as a decimated version of an underlying shift register sequence, with a bit decimation probability of 0.5. This paper extends the application to decimation probabilities different from 0.5. The results of this application have implications for the security provided by certain cryptographic algorithms, particularly with respect to correlation attacks.

2 Shrinking Generator

The shrinking generator produces keystream bits from two binary linear feedback shift registers (LFSR's) one of which controls the clock of the other. Denote these shift registers as LFSR_A and LFSR_S, as shown in Figure 2. The output from the shrinking generator is a "shrunken" version of the output from LFSR_A, with the elements selected being those in the positions corresponding to the 1's in the output sequence of LFSR_S: the keystream sequence z consists of those bits of the sequence a for which the corresponding bit of sequence s is a 1. The other bits of a, for which the corresponding bit of s is a 0, are deleted.

More precisely, let $a = \{a_i\}_{i=1}^{\infty}$ denote the LFSR_A sequence produced from a nonzero initial state $\{a_i\}_{i=1}^{r_A}$, and let $s = \{s_i\}_{i=1}^{\infty}$ denote the LFSR_S sequence produced from a nonzero initial state $\{s_i\}_{i=1}^{r_S}$, where r_A and r_S are the lengths of LFSR_A and LFSR_S, respectively. Let $z = \{z_k\}_{k=1}^{\infty}$ denote the output sequence of the shrinking generator. Then $z_k = a_{i_k}$ where i_k is the position of the k^{th} 1 in the sequence s. The keystream sequence z is an irregularly decimated version of the LFSR_A sequence

a, with the decimation controlled by the LFSR_S sequence s. An example of the keystream output, z, from a shrinking generator is given in Table 1.

s:	0	0	1	0	1	1	1	0	0	1	0	1	1	0	1	0
a:	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	1
z:			0		1	0	0			1		1	1		0	

Table 1: Shrinking generator output.

3 Joint Probability

For a probabilistic correlation attack on an irregularly clocked shift register, a measure of the correlation between the output string produced by irregular clocking and the output of the LFSR when clocked regularly is required: that is, a measure of the correlation between strings of different lengths. In [3] the measure of the correlation between two sequences X^m, of length m, and Y^n, of length n, $m \geq n$, is the *joint probability*.

The joint probability $P(X^m, Y^n)$ for arbitrary binary input and output strings $X^m = \{x_i\}_{i=1}^m$ and $Y^n = \{y_i\}_{i=1}^n$, respectively, is computed using a recursive algorithm based on string prefixes. Let $X^{e+s} = \{x_i\}_{i=1}^{e+s}$ denote the prefix of X^m of length $e+s$ and $Y^s = \{y_i\}_{i=1}^s$ denote the prefix of Y^n of length s. Let $P(e,s)$ denote the partial joint probability for X^{e+s} and Y^s, for $1 \leq s \leq n$ and $0 \leq e \leq m-n$. Let $\delta(x,y)$ denote the substitution probability, which equals 0.5 if $x = y$ and 0 otherwise, and let p denote the symbol deletion probability. The partial probability satisfies the recursion

$$P(e,s) \;=\; P(e-1,s)p + P(e,s-1)(1-p)\delta(x_{e+s}, y_s) \tag{1}$$

for $1 \leq s \leq n$ and $0 \leq e \leq m-n$, with the initial values $P(e,0) = p^e, 0 \leq e \leq m-n$, and $P(-1,s) = 0, 1 \leq s \leq n$. Note that $P(X^m, Y^n) = P(m-n, n)$. Thus the computational complexity is $O(n(m-n))$.

For practical calculations, where n is even moderately large, the joint probability values calculated in this manner quickly approach zero. Therefore the use of a

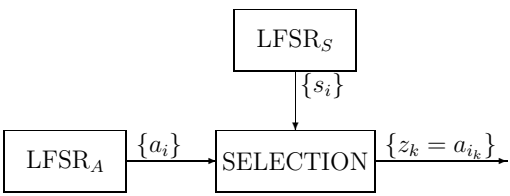

Figure 2: The shrinking generator.

modified version of this statistic, termed a *normalised joint probability* is investigated. In calculating the normalised joint probability the partial probability is multiplied by a constant factor in each iteration. For the case where $p = 0.5$, a normalising factor of $\frac{4}{1+3p}$ was identified through experimentation.

To make a decision on the relationship between two binary strings there are two hypotheses to be considered:

- $H_0 : X^m$ and Y^n are independent.
- $H_1 : X^m$ and Y^n are correlated, that is, Y^n is obtained from X^m by the decimation statistical model.

In a correlation attack on an irregularly clocked shift register, H_0 will correspond to an incorrect guess of the LFSR initial state, and H_1 will correspond to the correct guess. The statistic upon which the hypothesis testing is based is the *joint probability*.

In the application to the shrinking generator the binary input string X^m is replaced by $\{a_i\}_{i=1}^m$, the output from LFSR_A, and the binary output string Y^n by the keystream segment $\{z_i\}_{i=1}^n$, (obtained from the elements of LFSR_A in positions corresponding to the 1's in the output sequence of LFSR_S). Deletions of symbols from a are assumed to occur independently with deletion probability p, so that if a is assumed to be generated as a purely random sequence, that is, as a sequence of independent and identically distributed random variables, then z is also purely random.

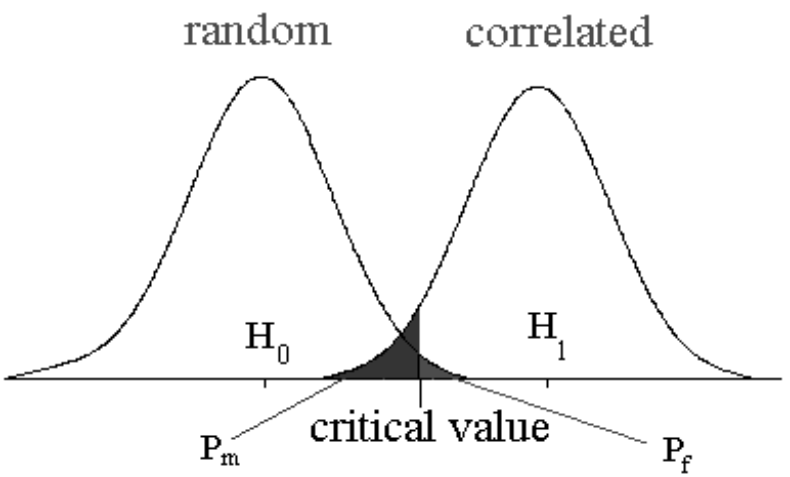

Figure 3: Errors in testing.

As with any hypothesis testing, there are two types of errors which can occur. These are illustrated in Figure 3. Let the error made by deciding that the two strings

are independent when in fact they are correlated be described as "missing the event", and the error made by deciding that the two strings are correlated when, in fact, they are independent be described as a "false alarm". Denote the probabilities of these events by P_m (accepting H_0 when H_1 is actually true) and P_f (accepting H_1 when H_0 is actually true), respectively. As for any hypothesis testing, for fixed n, decreasing P_m will result in an increase in P_f. A relationship between these probabilities exists, and is dependent on the lengths of the sequences, n and m.

The application of the joint probability as a measure of correlation in attacks on the shrinking generator with deletion probability $p = 0.5$ has been examined in [5]. In this paper, the application of the joint probability as a measure of correlation for sequences obtained through decimation with deletion probability values of p in the interval $0 < p < 1$ is investigated.

4 Results

Computer simulations were used to determine whether the joint probability is a useful measure to differentiate between a pair of independent strings (H_0) and a pair of correlated strings with $0 < p < 1$ (H_1). Two cases were examined: firstly, the random case (RAND) where X^m and Y^n are two independent random strings, and secondly, the correlated case (CORR) where the string Y^n is a decimation of X^m for the given values of p.

Basic simulations were performed for the string lengths $n = 150, 225$ and 300 bits for the sequence Y, with the string lengths of the sequence X chosen as $m(n) = n/(1-p) + 3\sqrt{n}$. The additional length of $3\sqrt{n}$ in $m(n)$ is chosen to ensure that the probability that $m^* > m(n)$ is very small, where m^* is the length of the input string that actually produced the output string of length n ([5]).

Simulations were conducted for values of p ranging from 0.1 to 0.9, in steps of 0.1. For each value of p and for each string length n, five thousand pairs of strings were generated and the normalised joint probability calculated for each pair. The normalised joint probability was generated by the recursion in (1) multiplied by the normalising factor $\frac{4}{1+3p}$.

The simulated distributions of the normalised joint probabilities have been recorded using percentiles in Tables 2 to 5 and illustrated by cumulative frequency graphs in Figures 4 to 6. As the normalised joint probability values range over a wide interval, for any combination of values of n and p, Figures 4 to 6 illustrate the cumulative distribution of $log_{10}(\textit{normalised joint probability})$ for each value of n, with values of p equal to $0.6, 0.7$ and 0.9. For any given value of p, the results show a gradual decrease in normalised joint probability values as n increases. Similarly, for a given value of n there is a decrease in normalised joint probability values as p increases. Additionally, the graphs clearly show changes in the distinction between the random and correlated cases as p and n are varied.

The distributions for $p \leq 0.5$ have been summarised in terms of maximum and minimum values in Table 2. For random strings all percentiles are zero when $p = 0.1$ and 0.2. For $p = 0.3$ a few of the normalised joint probability values were non-

Table 2: End points of normalised joint probability distributions for $p \leq 0.5$.

		RAND		CORR	
p	n	Minimum	Maximum	Minimum	Maximum
	150	0	0	$4.7 \cdot 10^{6}$	$1.6 \cdot 10^{19}$
0.1	225	0	0	$1.7 \cdot 10^{14}$	$1.4 \cdot 10^{28}$
	300	0	0	$7.8 \cdot 10^{21}$	$4.4 \cdot 10^{37}$
	150	0	0	$1.7 \cdot 10^{-3}$	$2.6 \cdot 10^{9}$
0.2	225	0	0	$1.9 \cdot 10^{-2}$	$9.1 \cdot 10^{12}$
	300	0	0	$1.7 \cdot 10^{-1}$	$4.0 \cdot 10^{15}$
	150	0	$1.1 \cdot 10^{-13}$	$1.3 \cdot 10^{-13}$	$3.8 \cdot 10^{0}$
0.3	225	0	0	$3.5 \cdot 10^{-18}$	$1.5 \cdot 10^{-2}$
	300	0	0	$1.0 \cdot 10^{-23}$	$1.6 \cdot 10^{-5}$
	150	0	$1.4 \cdot 10^{-17}$	$2.1 \cdot 10^{-21}$	$1.6 \cdot 10^{-7}$
0.4	225	0	$3.1 \cdot 10^{-29}$	$6.2 \cdot 10^{-30}$	$4.9 \cdot 10^{-15}$
	300	0	$2.8 \cdot 10^{-42}$	$4.7 \cdot 10^{-39}$	$3.3 \cdot 10^{-22}$
	150	0	$3.7 \cdot 10^{-22}$	$1.3 \cdot 10^{-27}$	$2.8 \cdot 10^{-16}$
0.5	225	0	$2.7 \cdot 10^{-35}$	$5.9 \cdot 10^{-40}$	$4.0 \cdot 10^{-26}$
	300	0	$1.1 \cdot 10^{-49}$	$2.0 \cdot 10^{-51}$	$1.2 \cdot 10^{-37}$

zero when $n = 150$. These results suggest that it is extremely unlikely, given two independent random binary strings (one of length n and the other of length $m(n)$) that the random binary string of length n, could be obtained from the longer string by deleting 30% or less of the bits, for $n = 150$, 225 or 300 and the corresponding lengths $m(n) = 252$, 367 or 481, respectively. In the case of the correlated strings the majority of the normalised joint probabilities were greater than 1 for $p = 0.1$ and 0.2. This was attributed to the normalising factor, which also caused a few of the upper values to be greater than 1 for $p = 0.3$. For the case $p = 0.4$, while there were a few more non-zero values for the random strings, the majority of values were zero. In the correlated case all normalised joint probabilities were very low, yet the distributions were clearly distinct from the random case.

The results for $p = 0.5$ show similar values to those obtained in [5]. In this instance the majority of values for the random strings are non-zero. When $p = 0.5$ and $n = 150$, over 75% of the five thousand normalised joint probability values generated in the random case were less than $1.3 \cdot 10^{-27}$, the smallest value generated in the correlated case (this agrees with the result obtained in [5]). As p decreases this percentage increases markedly, whereas for $p > 0.5$ this percentage gradually decreases, so that when $p = 0.9$ less than 2.5% of the distribution of the random strings is less than $2.0 \cdot 10^{-42}$, the minimum value in the distribution of the correlated strings. The same pattern occurs for $n = 225$ and 300: with over 90% and 97.5%, respectively, of normalised joint probability values generated in the random case being less than the smallest value generated in the correlated case for $p = 0.5$. This percentage gradually decreases to less than 2.5% for $p = 0.9$ for both values of n. It

Table 3: Percentiles of normalised joint probability distribution for $p = 0.6$.

%	**Keystream length** n					
	150		**225**		**300**	
	RAND	CORR	RAND	CORR	RAND	CORR
0	$2.0 \cdot 10^{-41}$	$4.7 \cdot 10^{-32}$	$1.5 \cdot 10^{-58}$	$1.1 \cdot 10^{-47}$	$3.5 \cdot 10^{-78}$	$5.8 \cdot 10^{-62}$
2.5	$3.3 \cdot 10^{-36}$	$8.4 \cdot 10^{-30}$	$2.8 \cdot 10^{-52}$	$2.1 \cdot 10^{-44}$	$5.5 \cdot 10^{-69}$	$4.5 \cdot 10^{-59}$
5	$2.1 \cdot 10^{-35}$	$2.3 \cdot 10^{-29}$	$1.4 \cdot 10^{-51}$	$7.6 \cdot 10^{-44}$	$5.6 \cdot 10^{-68}$	$1.8 \cdot 10^{-58}$
10	$1.3 \cdot 10^{-34}$	$7.1 \cdot 10^{-29}$	$1.0 \cdot 10^{-50}$	$3.3 \cdot 10^{-43}$	$5.6 \cdot 10^{-67}$	$7.6 \cdot 10^{-58}$
25	$2.6 \cdot 10^{-33}$	$4.2 \cdot 10^{-28}$	$2.6 \cdot 10^{-49}$	$2.9 \cdot 10^{-42}$	$2.1 \cdot 10^{-65}$	$8.1 \cdot 10^{-57}$
50	$5.9 \cdot 10^{-32}$	$3.0 \cdot 10^{-27}$	$7.9 \cdot 10^{-48}$	$2.9 \cdot 10^{-41}$	$7.8 \cdot 10^{-64}$	$1.3 \cdot 10^{-55}$
75	$9.1 \cdot 10^{-31}$	$2.3 \cdot 10^{-26}$	$2.3 \cdot 10^{-46}$	$3.1 \cdot 10^{-40}$	$2.3 \cdot 10^{-62}$	$2.0 \cdot 10^{-54}$
90	$9.2 \cdot 10^{-30}$	$1.3 \cdot 10^{-25}$	$3.2 \cdot 10^{-45}$	$2.8 \cdot 10^{-39}$	$5.3 \cdot 10^{-61}$	$2.4 \cdot 10^{-53}$
95	$3.1 \cdot 10^{-29}$	$3.9 \cdot 10^{-25}$	$1.4 \cdot 10^{-44}$	$9.6 \cdot 10^{-39}$	$2.6 \cdot 10^{-60}$	$1.3 \cdot 10^{-52}$
97.5	$8.1 \cdot 10^{-29}$	$9.3 \cdot 10^{-25}$	$4.7 \cdot 10^{-44}$	$2.9 \cdot 10^{-38}$	$1.0 \cdot 10^{-59}$	$5.6 \cdot 10^{-52}$
100	$2.9 \cdot 10^{-26}$	$1.7 \cdot 10^{-22}$	$1.2 \cdot 10^{-41}$	$2.4 \cdot 10^{-35}$	$2.5 \cdot 10^{-55}$	$2.1 \cdot 10^{-48}$

can be seen that, as p approaches 1, these percentages appear to tend to zero.

Tables 3 to 5 show the distributions for selected values of $p > 0.5$ (specifically $p = 0.6, 0.7$ and 0.9). Table entries are the normalised joint probability percentiles: values below which a given percentage of the sample points lie. For example, in the case where $p = 0.6$ and $n = 150$ with the strings generated independently at random, ten percent of the 5,000 normalised joint probability values calculated lie below $1.3 \cdot 10^{-34}$, and in the case of the correlated strings (Y^m is obtained as a decimation of X^m with $p = 0.6$), ten percent of the 5,000 normalised joint probability values calculated lie below $7.1 \cdot 10^{-29}$. The normalised joint probability values for $p > 0.5$ were all extremely low (less than $1.7 \cdot 10^{-22}$ when $p = 0.6$ and $n = 150$ in the correlated case). The distributions for the correlated strings were higher than the corresponding ones for the random strings, with the lowest maximum value in the correlated case being $2.5 \cdot 10^{-77}$ when $p = 0.9$ and $n = 300$.

5 Analysis of Results

The results presented in Table 2 for $p \leq 0.5$ show a clear distinction between the distributions for random and correlated strings and support the conjecture that the joint probability is a useful measure of correlation as a basis of correlation attacks on a "shrinking" style generator when less than half of the output of LFSR_A is decimated to yield the keystream.

The results presented in Tables 3 to 5 and Figures 4 to 6 for $p > 0.5$ show clear differences in the distributions of the normalised joint probabilities between the random and the correlated cases, consistent with the result for $p = 0.5$ in [5]. Figures 4 to 6 clearly illustrate that, given a fixed value of p, as n increases the distribution of the normalised joint probability shifts to the left (that is, decreases) for both

Table 4: Percentiles of normalised joint probability distribution for $p = 0.7$.

%	**Keystream length** n					
	150		**225**		**300**	
	RAND	CORR	RAND	CORR	RAND	CORR
0	$7.3 \cdot 10^{-42}$	$5.0 \cdot 10^{-36}$	$2.6 \cdot 10^{-58}$	$2.0 \cdot 10^{-52}$	$3.0 \cdot 10^{-76}$	$4.2 \cdot 10^{-70}$
2.5	$1.0 \cdot 10^{-37}$	$3.3 \cdot 10^{-34}$	$2.6 \cdot 10^{-55}$	$8.9 \cdot 10^{-51}$	$9.9 \cdot 10^{-73}$	$1.8 \cdot 10^{-67}$
5	$3.0 \cdot 10^{-37}$	$6.8 \cdot 10^{-34}$	$9.8 \cdot 10^{-55}$	$2.2 \cdot 10^{-50}$	$4.8 \cdot 10^{-72}$	$5.5 \cdot 10^{-67}$
10	$1.1 \cdot 10^{-36}$	$1.6 \cdot 10^{-33}$	$4.5 \cdot 10^{-54}$	$5.4 \cdot 10^{-50}$	$2.5 \cdot 10^{-71}$	$1.6 \cdot 10^{-66}$
25	$8.3 \cdot 10^{-36}$	$6.8 \cdot 10^{-33}$	$4.9 \cdot 10^{-53}$	$3.1 \cdot 10^{-49}$	$2.4 \cdot 10^{-70}$	$1.1 \cdot 10^{-65}$
50	$6.7 \cdot 10^{-35}$	$3.1 \cdot 10^{-32}$	$5.2 \cdot 10^{-52}$	$1.9 \cdot 10^{-48}$	$3.2 \cdot 10^{-69}$	$9.2 \cdot 10^{-65}$
75	$4.9 \cdot 10^{-34}$	$1.5 \cdot 10^{-31}$	$4.6 \cdot 10^{-51}$	$1.1 \cdot 10^{-47}$	$3.6 \cdot 10^{-68}$	$7.3 \cdot 10^{-64}$
90	$2.3 \cdot 10^{-33}$	$6.3 \cdot 10^{-31}$	$3.3 \cdot 10^{-50}$	$7.1 \cdot 10^{-47}$	$2.8 \cdot 10^{-67}$	$4.8 \cdot 10^{-63}$
95	$5.3 \cdot 10^{-33}$	$1.4 \cdot 10^{-30}$	$9.7 \cdot 10^{-50}$	$2.0 \cdot 10^{-46}$	$8.3 \cdot 10^{-67}$	$1.4 \cdot 10^{-62}$
97.5	$1.1 \cdot 10^{-32}$	$3.3 \cdot 10^{-30}$	$2.3 \cdot 10^{-49}$	$4.2 \cdot 10^{-46}$	$2.4 \cdot 10^{-66}$	$4.2 \cdot 10^{-62}$
100	$2.7 \cdot 10^{-31}$	$2.2 \cdot 10^{-28}$	$2.0 \cdot 10^{-47}$	$8.4 \cdot 10^{-44}$	$1.7 \cdot 10^{-64}$	$6.3 \cdot 10^{-60}$

Table 5: Percentiles of normalised joint probability distribution for $p = 0.9$.

%	**Keystream length** n					
	150		**225**		**300**	
	RAND	CORR	RAND	CORR	RAND	CORR
0	$1.7 \cdot 10^{-43}$	$2.0 \cdot 10^{-42}$	$1.7 \cdot 10^{-63}$	$4.9 \cdot 10^{-62}$	$9.9 \cdot 10^{-83}$	$5.6 \cdot 10^{-82}$
2.5	$3.8 \cdot 10^{-42}$	$2.1 \cdot 10^{-41}$	$5.5 \cdot 10^{-62}$	$4.6 \cdot 10^{-61}$	$1.6 \cdot 10^{-81}$	$1.3 \cdot 10^{-80}$
5	$5.9 \cdot 10^{-42}$	$3.0 \cdot 10^{-41}$	$1.0 \cdot 10^{-61}$	$7.2 \cdot 10^{-61}$	$2.0 \cdot 10^{-81}$	$2.0 \cdot 10^{-80}$
10	$1.0 \cdot 10^{-41}$	$4.5 \cdot 10^{-41}$	$1.9 \cdot 10^{-61}$	$1.2 \cdot 10^{-60}$	$3.9 \cdot 10^{-81}$	$3.3 \cdot 10^{-80}$
25	$2.3 \cdot 10^{-41}$	$9.5 \cdot 10^{-41}$	$4.5 \cdot 10^{-61}$	$2.5 \cdot 10^{-60}$	$9.8 \cdot 10^{-81}$	$7.7 \cdot 10^{-80}$
50	$5.5 \cdot 10^{-41}$	$2.0 \cdot 10^{-40}$	$1.2 \cdot 10^{-60}$	$6.1 \cdot 10^{-60}$	$2.7 \cdot 10^{-80}$	$1.8 \cdot 10^{-79}$
75	$1.2 \cdot 10^{-40}$	$4.2 \cdot 10^{-40}$	$2.9 \cdot 10^{-60}$	$1.4 \cdot 10^{-59}$	$7.0 \cdot 10^{-80}$	$4.5 \cdot 10^{-79}$
90	$2.3 \cdot 10^{-40}$	$8.3 \cdot 10^{-40}$	$6.2 \cdot 10^{-60}$	$3.2 \cdot 10^{-59}$	$1.6 \cdot 10^{-79}$	$1.1 \cdot 10^{-78}$
95	$3.6 \cdot 10^{-40}$	$1.3 \cdot 10^{-39}$	$9.9 \cdot 10^{-60}$	$5.2 \cdot 10^{-59}$	$2.7 \cdot 10^{-79}$	$1.8 \cdot 10^{-78}$
97.5	$5.0 \cdot 10^{-40}$	$1.8 \cdot 10^{-39}$	$1.5 \cdot 10^{-59}$	$7.7 \cdot 10^{-59}$	$4.1 \cdot 10^{-79}$	$3.0 \cdot 10^{-78}$
100	$3.2 \cdot 10^{-39}$	$2.6 \cdot 10^{-38}$	$2.5 \cdot 10^{-58}$	$2.0 \cdot 10^{-57}$	$2.5 \cdot 10^{-78}$	$2.5 \cdot 10^{-77}$

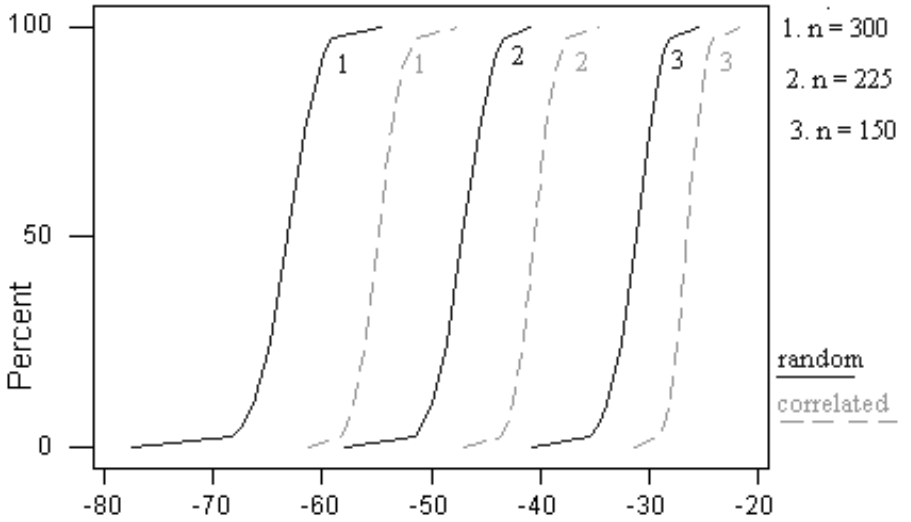

Figure 4: Cumulative distribution of log_{10}(normalised joint probability) for $p = 0.6$.

random and correlated strings, with the distinction between random and correlated cases increasing. It is also clear that, given a fixed value of n, as p increases, the distribution of the normalised joint probability shifts to the left (decreases) for both random and correlated strings, with the distinction between random and correlated cases decreasing.

For fixed n and p, in comparing distributions of the normalised joint probability for both the random and correlated cases, we observe that for every percentile, the normalised joint probability value in the random case is always less than the normalised joint probability value in the correlated case. However, as noted above, the degree of difference between the sample distributions in the random and correlated cases also depends on the values of n and p. These differences in the distributions of the normalised joint probability for both random and correlated sequences indicate that the normalised joint probability is a useful measure of the correlation between two strings of different lengths, yet decreases in effectiveness as p increases, unless n increases also.

5.1 Statistical Distinction

The statistical distinction between the normalised joint probability distributions for random and correlated strings obtained through the computer simulation can be analyzed by estimating the probabilities for the two types of errors which can occur when hypotheses are tested. These errors were described in Section 3, and termed "missing the event" and "false alarm". The probability of "missing the event", P_m,

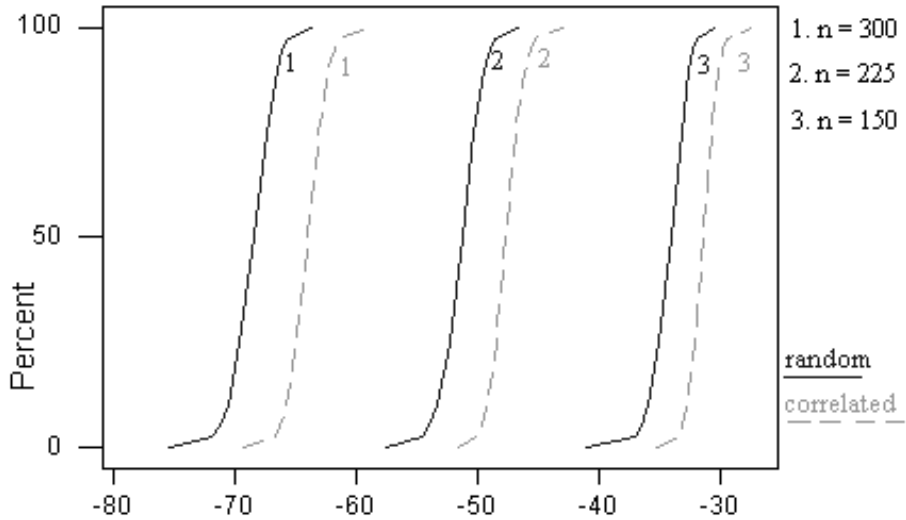

Figure 5: Cumulative distribution of log_{10}(normalised joint probability) for $p = 0.7$.

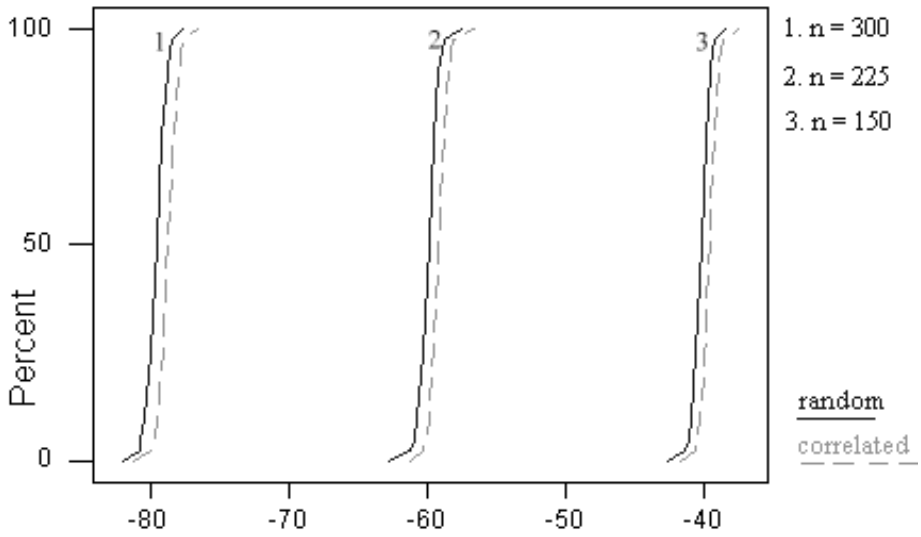

Figure 6: Cumulative distribution of log_{10}(normalised joint probability) for $p = 0.9$.

represents the probability of declaring that the strings are not correlated when they actually are. The probability of a "false alarm", P_f, is the probability that two strings are declared correlated when they actually are not.

The probability of "missing the event", P_m, may be estimated for selected values of the probability of a "false alarm", P_f. Estimates of P_m were obtained by determining the proportion of the distribution of the normalised joint probability for the correlated strings that falls below a "critical value" obtained from the distribution for the random strings. This critical value was the $(1 - P_f) \times 100$ percentile of the distribution for the random strings. For decimation probabilities $p = 0.6$ and $p = 0.7$, Table 6 presents estimates of P_m for the P_f values 0.01, 0.05 and 0.1 and the string lengths $n = 150, 225$ and 300. For example, in the case where a 60% decimation was applied to yield a known keystream of 300 bits, and allowing for a 5% error rate for false alarms, $P_f = 0.05$, the estimate for the probability of mistaking two correlated strings for random strings is $P_m = 0.0036$. This means that, for a given known sequence of 300 bits, if we are prepared to accept a false alarm in 5 percent of the longer sequences tested, then the probability of failing to detect the actual longer sequence that the given keystream sequence is a decimation of is less than 0.4 percent.

Similarly, the probability of a "false alarm", P_f, may be estimated for given values of the probability of "missing the event" (P_m). Estimates of P_f were obtained by determining the proportion of the distribution of the normalised joint probability for the random strings that lies above a "critical value" obtained from the distribution for the correlated strings. This critical value was the $P_m \times 100$ percentile of the distribution for the correlated strings. These estimates are presented in Table 7 for $p = 0.6$ and in Table 8 for $p = 0.7$, with the selected values of P_m taken as 0.01, 0.05 and 0.1. For example, again for the case where a 60% decimation has been applied to yield a short keystream of 300 bits, allowing a 5% error for missing the event, $P_m = 0.05$, the error of mistaking two random strings as being correlated is estimated to be $P_f = 0.0058$. This means that, if we accept an error rate of 5 percent for failing to detect that two streams are correlated, then we should should expect to mistakenly decide that the known keystream is a decimation of the longer test stream for approximately 0.6 percent of longer strings.

Further values for n have been included in Tables 7 and 8 to investigate the exponential relationship $P_f(n) = a \cdot b^n$, proposed in [5]. The results show that for a fixed value of P_m, although P_f decreases as n increases, P_f increases as p increases. That is, for some fixed p, the greater the amount of keystream known, then the more likely it is for an attacker to correctly guess the underlying longer sequence from the known keystream. However, for some fixed n, the greater the decimation rate applied to form the keystream, then the more difficult it is for an attacker to correctly guess the underlying longer sequence from the known keystream.

This relationship may be approximated by an exponential function in n, written as $P_f(n) \approx a \cdot b^n$. When $P_m = 0.1$ the results of applying regression analysis to the estimates of P_f in Tables 7 and 8 give $P_f(n) \approx 0.46 \cdot 0.98^n$ for $p = 0.6$ and $P_f(n) \approx 0.45 \cdot 0.99^n$ for $p = 0.7$. These relationships both give a 99% correlation. In accordance with [5] we obtained $P_f(n) \approx 0.44 \cdot 0.97^n$ for $p = 0.5$ with a 96%

Table 6: P_m versus n given P_f for $p = 0.6$ and $p = 0.7$.

		$p = 0.6$		$p = 0.7$	
n	P_f	**Critical value**	P_m	**Critical value**	P_m
150	0.01	$3.3 \cdot 10^{-28}$	0.2138	$2.5 \cdot 10^{-32}$	0.4568
	0.05	$3.7 \cdot 10^{-29}$	0.0590	$5.5 \cdot 10^{-33}$	0.2246
	0.10	$1.0 \cdot 10^{-29}$	0.0244	$2.2 \cdot 10^{-33}$	0.1248
225	0.01	$2.1 \cdot 10^{-43}$	0.0832	$4.4 \cdot 10^{-49}$	0.3004
	0.05	$1.4 \cdot 10^{-44}$	0.0154	$7.2 \cdot 10^{-50}$	0.1116
	0.10	$3.1 \cdot 10^{-45}$	0.0042	$2.7 \cdot 10^{-50}$	0.0634
300	0.01	$7.1 \cdot 10^{-59}$	0.0298	$6.5 \cdot 10^{-66}$	0.2004
	0.05	$3.0 \cdot 10^{-60}$	0.0036	$8.9 \cdot 10^{-67}$	0.0690
	0.10	$5.2 \cdot 10^{-61}$	0.0014	$2.7 \cdot 10^{-66}$	0.0318

Table 7: P_f versus n given P_m for $p = 0.6$.

	$P_m = \mathbf{0.01}$		$P_m = \mathbf{0.05}$		$P_m = \mathbf{0.10}$	
n	Critical value	P_f	Critical value	P_f	Critical value	P_f
100	$1.1 \cdot 10^{-20}$	0.3086	$7.3 \cdot 10^{-20}$	0.1404	$1.9 \cdot 10^{-19}$	0.0792
125	$1.8 \cdot 10^{-25}$	0.2304	$1.3 \cdot 10^{-24}$	0.0970	$3.8 \cdot 10^{-24}$	0.0506
150	$3.4 \cdot 10^{-30}$	0.1630	$2.9 \cdot 10^{-29}$	0.0554	$8.0 \cdot 10^{-29}$	0.0314
175	$4.0 \cdot 10^{-35}$	0.1422	$4.4 \cdot 10^{-34}$	0.0442	$1.3 \cdot 10^{-33}$	0.0254
200	$5.4 \cdot 10^{-40}$	0.0982	$6.8 \cdot 10^{-39}$	0.0236	$2.4 \cdot 10^{-38}$	0.0100
225	$8.4 \cdot 10^{-45}$	0.0644	$8.7 \cdot 10^{-44}$	0.0170	$3.0 \cdot 10^{-43}$	0.0080
250	$8.3 \cdot 10^{-50}$	0.0536	$1.2 \cdot 10^{-48}$	0.0126	$4.4 \cdot 10^{-48}$	0.0052
275	$1.1 \cdot 10^{-54}$	0.0382	$1.7 \cdot 10^{-53}$	0.0084	$7.6 \cdot 10^{-53}$	0.0032
300	$1.3 \cdot 10^{-59}$	0.0262	$1.9 \cdot 10^{-58}$	0.0058	$1.2 \cdot 10^{-58}$	0.0026

Table 8: P_f versus n given P_m for $p = 0.7$.

	$P_m = \mathbf{0.01}$		$P_m = \mathbf{0.05}$		$P_m = \mathbf{0.10}$	
n	Critical value	P_f	Critical value	P_f	Critical value	P_f
100	$1.2 \cdot 10^{-23}$	0.5370	$6.6 \cdot 10^{-23}$	0.2814	$1.4 \cdot 10^{-22}$	0.1920
125	$4.1 \cdot 10^{-29}$	0.4848	$2.4 \cdot 10^{-28}$	0.2452	$5.3 \cdot 10^{-28}$	0.1538
150	$1.1 \cdot 10^{-34}$	0.4422	$7.0 \cdot 10^{-34}$	0.2040	$1.7 \cdot 10^{-33}$	0.1228
175	$3.1 \cdot 10^{-40}$	0.3922	$2.3 \cdot 10^{-39}$	0.1650	$5.4 \cdot 10^{-39}$	0.1020
200	$1.0 \cdot 10^{-45}$	0.3324	$6.4 \cdot 10^{-45}$	0.1492	$1.7 \cdot 10^{-44}$	0.0808
225	$2.6 \cdot 10^{-51}$	0.3110	$1.9 \cdot 10^{-50}$	0.1184	$5.9 \cdot 10^{-50}$	0.0586
250	$6.9 \cdot 10^{-57}$	0.2886	$5.6 \cdot 10^{-56}$	0.1060	$1.7 \cdot 10^{-55}$	0.0514
275	$1.9 \cdot 10^{-62}$	0.2562	$1.8 \cdot 10^{-61}$	0.0912	$5.5 \cdot 10^{-61}$	0.0442
300	$5.6 \cdot 10^{-68}$	0.2220	$5.2 \cdot 10^{-67}$	0.0698	$1.6 \cdot 10^{-66}$	0.0344

correlation.

Tables 6 to 8 show that for a fixed length n, in order to reduce the probability of one type of error, the probability of the other type of error must be increased. However, if the string length is increased and the probability of one of the two types of errors is fixed, then the probability of the other type of error will decrease.

6 Discussion

The results clearly show firstly, that the normalised joint probability is a useful measure to differentiate between correlated and independent pairs of binary strings of different lengths and secondly, that the probabilities of errors for decisions based on this statistic decrease as the string lengths increase. Additionally, the results show that as the decimation probability increases, the length of the known keystream required for the normalised joint probability to be a useful measure of correlation must also be increased. ¿From a cryptanalytic point of view, the normalised joint probability is thus shown to be a suitable basis for correlation attacks on an irregularly clocked shift register where the deletions occur independently with probability p, provided a sufficient length of the keystream is known.

It follows that the greater the length of known keystream available to the cryptanalyst, the easier it becomes to discriminate between correlated and random strings. That is, the probability that the correct secret key may be recovered in a correlation attack increases as the amount of keystream known increases. For a fixed decimation probability and for a given probability of missing the event, the probability of a false alarm decreases as n increases, which renders the attack more likely to succeed. The minimal keystream length required for a successful recovery of the secret-key-controlled initial state of the underlying LFSR_A depends on its length, r_A. It can be estimated by using the criterion $2^{r_A} \cdot P_f(n) \leq 1$, in view of the fact that $2^{r_A} - 1$ is the number of incorrect guesses about the LFSR_A initial state ([5]). If $P_f(n) = a \cdot b^n$, then it follows that $n \geq \frac{1}{-log_2 b} \cdot r_A$, neglecting the small term corresponding to a. For example, given $P_m = 0.1$, it turns out that the minimal keystream length is approximately $20 \cdot r_A$, $35 \cdot r_A$ and $70 \cdot r_A$ for $p = 0.5, 0.6$ and 0.7, respectively.

As the decimation probability increases, for a given probability of missing the event, the probability of a false alarm increases. Therefore, from a cryptographic point of view a keystream generator with a high decimation probability offers greater security, as it is less likely that an attacker can discriminate between correlated and random strings. Quite a lot of keystream is required to make the distinction with only a small probability of error. For example, for a keystream generator like the shrinking generator, if more of the output of LFSR_A is deleted, then it is harder to discriminate between independent and correlated strings.

References

[1] Coppersmith, D., Krawczyk, H. and Mansour, Y. (1993) The shrinking generator, in Advances in Cryptology – CRYPTO '93, *Lecture Notes in Computer Science*, **773**, Springer-Verlag, 22–39.

[2] Dawson, E., Simpson, L. and Golić, J. (2001) A survey of divide and conquer attacks on certain irregularly clocked stream ciphers, in Cryptography and Computational Number Theory, *Progress in Computer Science and Applied Logic*, **20**, Birkhauser Verlag, 165–185.

[3] Golić, J. Dj. and O'Connor, L. (1995) Embedding and probabilistic correlation attacks on clock-controlled shift registers, in Advances in Cryptology – EUROCRYPT '94, *Lecture Notes in Computer Science*, **950**, Springer-Verlag, 230–243.

[4] Siegenthaler, T. (1985) Decrypting a class of stream ciphers using ciphertext only, *IEEE Transactions on Computers*, **C-34(1)**, 81–85.

[5] Simpson, L., Golić, J. Dj. and Dawson, E. (1998) A probabilistic correlation attack on the shrinking generator, in Information Security and Privacy – ACISP'98, *Lecture Notes in Computer Science*, **1438**, Springer-Verlag, 147–158.

(Received 9 Jan 2001)