Known Plaintext Attack on the Binary Symmetric Wiretap Channel

Rajaraman Vaidyanathaswami and Andrew Thangaraj
Department of Electrical Engineering
Indian Institute of Technology, Madras
Chennai 600036, India
Email: rajaraman.v@gmail.com, andrew@ee.iitm.ac.in

***Abstract*—The coset encoding scheme for the wiretap channel depends primarily on generating a random sequence of bits for every code block. The secret message indexes into the set of cosets, and a random vector selects the actual code word to be transmitted from that coset. In the literature, it is usually assumed that a statistically perfect uniform random generator is available for this purpose. This study looks at practical ways of implementing a random number generator, especially the ones based on Linear Feedback Shift Registers. This opens up a broad class of security issues, since it is well known that stream ciphers based on LFSRs are vulnerable to correlation attacks and their improved variants. This article considers the known-plaintext attack, where the random vector used in wiretap coset encoding is attacked under the assumption that the message is known to the eavesdropper** *a priori*.

I. Introduction

The wiretap channel, introduced by Wyner [1] and later generalized by Csiszar and Korner [2], provides a framework for studying information-theoretic security. Wiretap encoding involves two requirements: (*i*) introduce redundancy so that the code word can be correctly decoded by the legitimate receiver, fulfilling the reliability objective. (*ii*) introduce randomness to keep the eavesdropper completely ignorant even if a subset of code bits are leaked out, which is the secrecy objective.

In practice, any source of randomness has to be physically implementable, and is likely to be a pseudo-random number (PRN) generator. If the eavesdropper can retrieve the seed of the PRN generator, the randomness in the encoding is broken. Harrison and McLaughlin [3][4] brought another dimension to this problem when they argued that the physical layer characteristics, that are often ignored in cryptographic literature, can be studied in conjunction with Linear Feedback Shift Register (LFSR) key stream security issues. They showed that the security of cryptographic primitives can be strengthened by exploiting the channel error rate in the physical layer.

In this paper, we consider Wyner's coset encoding scheme [1][5] for the wiretap channel, where the random vector for choice of codeword within a coset is generated by a LFSR or a LFSR-based stream cipher. We study fast correlation attacks on the random vector under the known-plaintext assumption. Our first observation is that a known-plaintext attack on coset encoding in a binary-symmetric wiretap channel is equivalent to a special case of the learning parity in noise (LPN) problem, which is NP-hard, in general. We later demonstrate several specific attacks and study their success probabilities. Based on this study, useful parameters such as key refresh rate of the PRN can be tuned in practical systems to ensure security.

II. The wiretap channel

The general wiretap channel model is shown in Fig. 1. Two legitimate users, Alice and Bob, are separated by a Discrete Memoryless Channel (DMC), and an adversary Eve listens to the transmissions through another DMC that we call the wiretapper's channel. The goal is for Alice to transmit such that both the objectives of reliability across main channel and secrecy across the wiretapper's channel can be achieved.

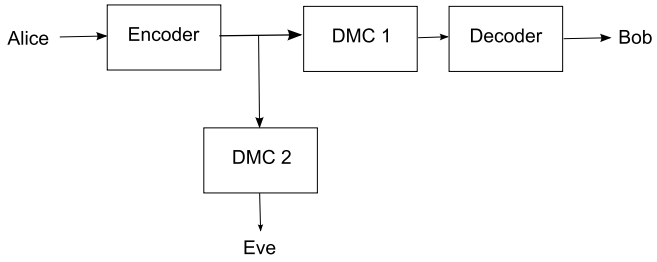

Fig. 1. Wiretap channel model.

In this article, we will be concerned with the binary symmetric wiretap channel, which is a special case of Fig. 1 with DMC1 being a noiseless channel and DMC2 being a binary symmetric channel (BSC) with a certain crossover probability.

A. Coset encoding scheme

In the coset encoding scheme for a binary symmetric wiretap channel, a k-bit message $\mathbf{m}$ is mapped to one of several possible codewords of length n. An underlying $(n, n-k)$ binary block code C is used in the one-to-many mapping. The code C divides the n-dimensional vector space $\{0,1\}^n$ into a set of 2^k disjoint cosets. The message $\mathbf{m}$ chooses a particular coset χ. In the second step, one of the 2^{n-k} vectors in the coset χ is chosen at random and is transmitted. This randomness is what provides security by enhancing equivocation.

The transmitted codeword in the coset encoding scheme can be written as

$$\mathbf{x} = \begin{bmatrix} \mathbf{v} & \mathbf{m} \end{bmatrix} \begin{bmatrix} G \\ G^* \end{bmatrix}, \tag{1}$$

where G is a generator matrix for the code C, G^* is a generator matrix for the complementary code C^* such that $C \oplus C^* = \{0,1\}^n$, and $\mathbf{v}$ is a $(n-k)$-bit random vector. Note that the rows of $\begin{bmatrix} G \\ G^* \end{bmatrix}$ form a basis for $\{0,1\}^n$.

Several choices are possible for the code C and its generator matrix. As suggested in [5], we will consider the choice of G as a low density parity check (LDPC) matrix with a certain degree distribution. Given a generator matrix G, we will randomly select rows of G^* from $\{0,1\}^n$ so that they form a full rank matrix in conjunction with G.

Since the main channel is noiseless, the legitimate receiver retrieves the message by finding the coset of C to which the vector $\mathbf{x}$ belongs. Note that Bob does not need the knowledge of $\mathbf{v}$. Supposing that the wiretapper's channel is a BSC with transition probability p, we can model the eavesdropper's received vector $\mathbf{y}$ as

$$\mathbf{y} = \mathbf{m}G^* + \mathbf{v}G + \mathbf{e}, \tag{2}$$

where $\mathbf{e} = [e_1\ e_2 \cdots e_n]$ is a random binary vector with e_i *iid* and $\Pr\{e_i = 0\} = 1-p$, $\Pr\{e_i = 1\} = p$ for $1 \le i \le n$.

B. LFSR Based PRN Generators

We now turn our attention to the random vector $\mathbf{v}$ used in the wiretap encoding process. A very popular method for generating PRN sequences is a Linear Feedback Shift Register (LFSR) based implementation. For increased security, output from a number of LFSRs will be combined in a non-linear combiner to obfuscate any discernible pattern in the output sequence. As shown in Fig. 2, the effect of having multiple LFSRs and combining them is modeled as a single LFSR (with possibly a longer register length) coupled with a BSC. The

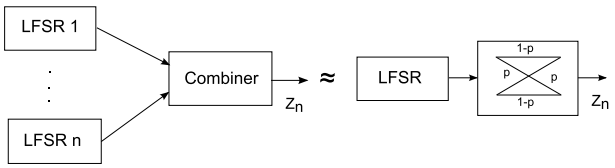

Fig. 2. LFSR Model.

transition probability of the BSC in the LFSR model of Fig. 2 is called the obfuscation probability (OP).

Security of the initial seed of the LFSR is an important factor for the security of systems that employ LFSR-based stream ciphers such as the one shown in Fig. 2. From now on, we will suppose that a LFSR-based PRN generator in used for generating the random vector $\mathbf{v}$. The initial seed of the LFSR will be assumed to be known only to Alice.

III. Known-plaintext attacks and LPN

In a known-plaintext attack, we assume that the attacker knows the message $\mathbf{m}$. This is possible in several scenarios, where Alice might be sending a preamble with universally known data to Bob. The goal of the known-plaintext attack is to try and determine the initial seed of the LFSR-based PRN generator. If Eve is successful and the initial seed is found, the random vectors in the coset encoding process of future unknown messages are revealed to Eve. Eve can now use the knowledge of the random vector to her advantage in attacking later uses of the wiretap channel with unknown messages.

Therefore, there are two phases in the overall attack.

1) Known-plaintext phase: In this phase, Eve knows $\mathbf{m}$ and tries to find the initial seed of the LFSR generator.
2) Unknown-plaintext phase: In this phase, Eve knows $\mathbf{v}$ and tries to find the message $\mathbf{m}$ from $\mathbf{m}G^* + \mathbf{e}$.

In the first phase, the initial seed of the LFSR-based PRN generator needs to be found from

$$\mathbf{y}' = \mathbf{v}G + \mathbf{e}, \tag{3}$$

since $\mathbf{m}$ and G^* are known to Eve.

Problems such as the one in (3) are known as the Learning Parity in Noise (LPN) Problem [6][7][8][9]. In general, for a randomly chosen G, LPN is known to be NP-hard through reduction from the maximum-likelihood decoding problem over a BSC [10]. Recently, the LPN problem has been studied in the context of the Hopper-Blum (HB) authentication protocol for cheap devices such as RFID tags.

Note that both phases of the attack involve the LPN problem. While the second phase is purely a version of the LPN problem, the first phase is more complicated because the goal is to not simply determine $\mathbf{v}$, but find the initial seed that generated $\mathbf{v}$. However, in the coset encoding context, the matrix G may not be chosen uniformly at random. For instance, G might be chosen from an ensemble of sparse LDPC codes with a certain degree distribution. In these cases, where G is sparse, the special case of the LPN problem has been studied in [11].

In the next section, we will briefly describe fast correlation attacks. This attack will be later employed in the known-plaintext phase to determine the initial LFSR seed.

IV. Fast correlation attacks

An LFSR of register length L generating an N bit sequence can be modeled [12] as an (N, L) linear block code $\mathcal{C}$ with generator matrix $\mathcal{G}$. Each of the 2^L initial seeds (or key bits) generates a code word of length N in $\mathcal{C}$. In the attacks, we assume that the connection polynomial of the LFSR (and, therefore, the matrix $\mathcal{G}$) is known, and the initial seed is secret. A system of LFSRs can be modeled as a single long LFSR followed by a Binary Symmetric Channel (BSC) that corrupts some of the output bits (see Fig. 2). Meier and Staffelbach suggested an algorithm similar to the standard decoding process used in error control coding [13][14] for determining the initial seed from corrupted output bits.

In this work, we employ fast correlation attacks based on [15]. The idea is to seek within $\mathcal{C}$ an embedded convolutional code whose length is much smaller than L resulting in implementable attacks. We will briefly describe the main parameters that control the complexity and success probability of the fast correlation attack. For more details, see [15][16].

A. Embedding a convolutional code

We begin by observing that $\mathcal{G}$ is an LxN matrix and can be written in systematic form as

$$G = \begin{bmatrix} I_{B+1} & P \\ 0_{L-B-1} & Q \end{bmatrix}, \quad (4)$$

where B is the main design parameter that provides tradeoff between processing complexity, memory requirements and success probability. Also, I_{B+1} is the $(B+1) \times (B+1)$ identity matrix, and 0_{L-B-1} is the $(L-B-1) \times (B+1)$ all-zero matrix. Let $[\mathcal{G}]_{i,j}$ denote the (i,j)-th entry of $\mathcal{G}$

Our focus is the $(L-B-1)$ x $(N-B-1)$ matrix Q. We seek column pairs i and j such that (1) within Q, columns i and j are identical i.e. $[\mathcal{G}]_{l,i} = [\mathcal{G}]_{l,j}$ for $B+2 \leq l \leq L$, and (2) $[\mathcal{G}]_{B+1,i} \oplus [\mathcal{G}]_{B+1,j} = 1$. From such a column pair i and j, we can generate a parity check condition satisfied by every codeword $[c_1\ c_2 \cdots c_N] \in \mathcal{C}$:

$$g_1 c_1 + g_2 c_2 + \cdots + g_B c_B + c_{B+1} + c_i + c_j = 0, \quad (5)$$

where $g_l = [\mathcal{G}]_{l,i} \oplus [\mathcal{G}]_{l,j}$, $1 \leq l \leq B$. Since $\mathcal{C}$ is generated by an LFSR, it is cyclic implying that (5) is satisfied for every cyclic shift of the codeword.

Suppose μ is the number of parity equations such as (5) that we can find corresponding to distinct column pairs $(i_1, j_1), (i_2, j_2), \cdots, (i_\mu, j_\mu)$. We can construct a rate-$1/(\mu+1)$ convolutional code with memory B defined by an encoder that takes as input c_n at time $1 \leq n \leq N$ and outputs the μ bits $f_n^{(0)} = c_n$, $f_n^{(1)} = c_{n+i_1-B-1} \oplus c_{n+j_1-B-1}$, $\cdots$, $f_n^{(\mu)} = c_{n+i_\mu-B-1} \oplus c_{n+j_\mu-B-1}$. Note that the indices in the above equations can be taken modulo N. We remark that the parameter B controls the feasibility and complexity of the fast correlation attack. If B is very small, column pairs identical in Q may not be plentiful resulting in a high-rate convolutional code and poor success probability in decoding corrupted bits. If B is too large, the complexity of decoding becomes a significant obstacle.

B. Decoding

Let the corrupted LFSR sequence intercepted by the attacker be $[z_1, z_2, ...z_N]$. We proceed to generate the simulated 'received sequence' $[r_n^{(0)}\ r_n^{(1)}\ \ldots\ r_n^{(m)}]$ for the embedded rate-$1/(\mu+1)$ convolutional code as follows: $r_n^{(0)} = z_n$, $r_n^{(1)} = z_{n+i_1-B-1} \oplus z_{n+j_1-B-1}$, $\cdots$, $r_n^{(\mu)} = z_{n+i_\mu-B-1} \oplus z_{n+j_\mu-B-1}$. We use the state transition table obtained for the convolutional encoder designed above to form the trellis with 2^B states and depth L. For the Viterbi decoding process, the branch metric of each branch is the Hamming distance between the received vector and the expected encoder output.

V. Simulation Results

In this section, we will present simulation results of several known-plaintext attacks on coset encoding over the binary symmetric wiretap channel. The known plaintext attack is split into two parts: (1) finding an estimate, denoted $\hat{\mathbf{v}}$, of the random vector $\mathbf{v}$ from $\mathbf{y}' = \mathbf{v}G + \mathbf{e}$, and (2) finding the initial LFSR seed from the estimated random vector $\hat{\mathbf{v}}$. For (1), we will employ published methods for solving LPN, such as [8][9][17]. For (2), we will use the fast correlation attacks as described in Section IV.

We begin by demonstrating the fast correlation attack on an LFSR with connection polynomial set as generator polynomials of Euclidean Geometry (EG) codes. We will consider three EG codes, denoted EG(4,2), EG(2,2^3) and EG(3,2^2), with generator polynomials $x^{10}+x^8+x^5+x^4+x^2+x+1$, $x^{26}+x^{24}+x^{16}+x^{15}+x^{14}+x^{13}+x^{12}+x^{10}+x^6+x^2+1$ and $x^{15}+x^{14}+x^{13}+x^{11}+x^4+x^2+1$ [18]. So, the LFSR lengths are 10, 26 and 15, respectively. For the fast correlation, we set the memory of the embedded convolutional code as $B = 4$, 12 and 8, respectively. For the three EG codes, rate-$1/6$, rate-$1/40$ and rate-$1/240$ convolutional codes were found for embedding. The results are shown in Fig 3, where the probability of success in finding the initial seed is plotted against the error probability of the BSC through which the LFSR outputs are passed.

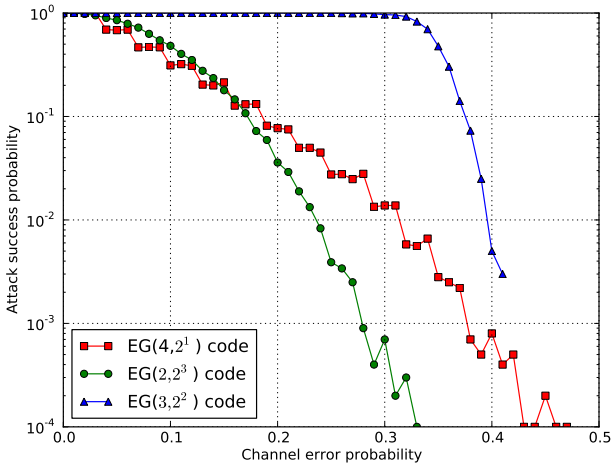

Fig. 3. Fast Correlation Attack on an EG LFSRs.

In Fig. 3, we see that the behavior of success probability depends on the LFSR length and the rate of the embedded convolutional code. The code EG(3,2^2) with LFSR length 15 fares poorly when compared to EG(4,2) with LFSR length 10 mainly on account of the rate-1/240 embedded convolutional code. Also, EG(2,2^3) is better than EG(4,2) (at larger channel error probabilities) because of the longer register length, in spite of the lesser rate-1/40 embedded code. Notice that the channel error probability can also be thought of as the obfuscation probability in the model of Fig. 2 for stream ciphers with multiple LFSRs.

In the next few plots, we will show results of known-plaintext attacks on the binary symmetric wiretap channel. We will fix a length-8 LFSR with connection polynomial $x^8+x^7+x^5+x^3+1$, and the initial seed will be 8 bits long. The reason for choosing a low-length LFSR is to keep all simulation parameters tractable. The model of Fig. 2 will be used and simulations will be done for several obfuscation

probabilities (OPs).

Fig. 4 shows plots of success probability of fast correlation attacks on $\mathbf{v}+\mathbf{e}$ as a function of channel error probability with which $\mathbf{e}$ was generated. In the setting used for this figure, there

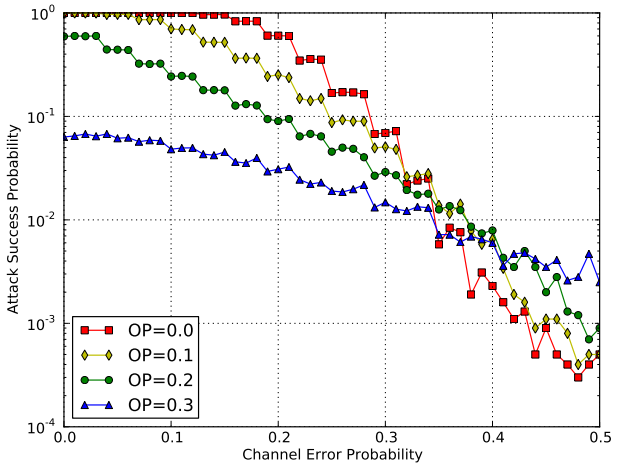

Fig. 4. Fast Correlation Attack on an LFSR sequence.

is no wiretap coset encoding. We see that the success probability shows the expected declining behavior with channel error probability with a flooring at $1/256 \approx 4 \times 10^{-3}$, which is the probability of success of a random guess. The parameter B of the fast correlation attack was set as $B = 4$. Note that the probability of success deteriorates with increase in obfuscation probability.

In Fig. 5, we bring out the effect of wiretap encoding on the fast correlation attack. The same length-8 LFSR is used

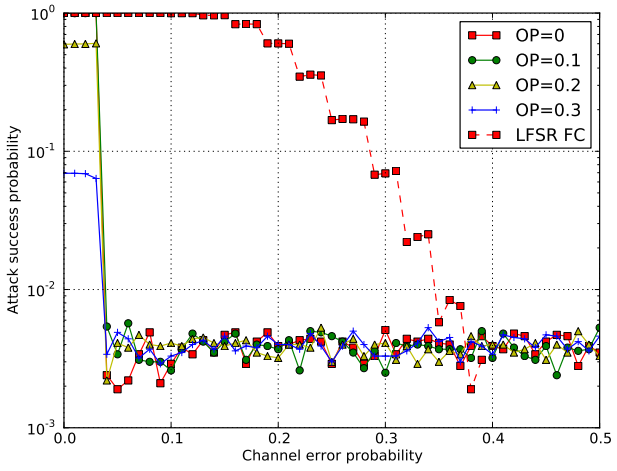

Fig. 5. Fast Correlation Attack with wiretap coset encoding.

for generating the random vector $\mathbf{v}$. Wiretap coset encoding is performed with the matrix G set to be a (3,6) regular LDPC matrix of dimensions 32×64. So, the eavesdropper processes $\mathbf{y}' = \mathbf{v}G + \mathbf{e}$ under the known-plaintext assumption (as opposed to $\mathbf{v} + \mathbf{e}$ for the results in Fig. 4). To estimate $\mathbf{v}$ from $\mathbf{y}'$, we employ the Carrijo solution [17] to LPN, where 32 columns of G are randomly chosen so as to get an invertible submatrix G_s. Then, we set $\hat{\mathbf{v}} = (G_s)^{-1}\mathbf{y}'$ assuming that $\mathbf{e}$ is zero at those positions. The estimate $\hat{\mathbf{v}}$ is input to a fast correlation attack with $B = 4$.

In Fig. 5, we see that the success probability quickly floors to the random guess level in the presence of wiretap encoding. For contrast, the dotted line shows the success probability without wiretap encoding. The channel probability at which the success probability reaches the random guess level is significantly reduced from about 0.36 to about 0.04. Also, in the presence of wiretap encoding, the obfuscation probability appears to have very little effect on the fast correlation success probability. A sharp threshold effect is being observed in the wiretap encoded case - beyond a certain channel probability (around 0.04), the success probability floors to the random guess level. The Carrijo attack [17] is not the strongest possible attack on the LPN problem, and that is likely to be the main reason for the effects observed in Fig. 5.

We now consider stronger attacks such as the BKW and LF2 attacks[8][9]. Fig. 6 shows a plot of probability of success of the BKW algorithm for three different choice of parameters. In the figure, BKW(a,b,n) indicates that, in $\mathbf{v}G+\mathbf{e}$, the length

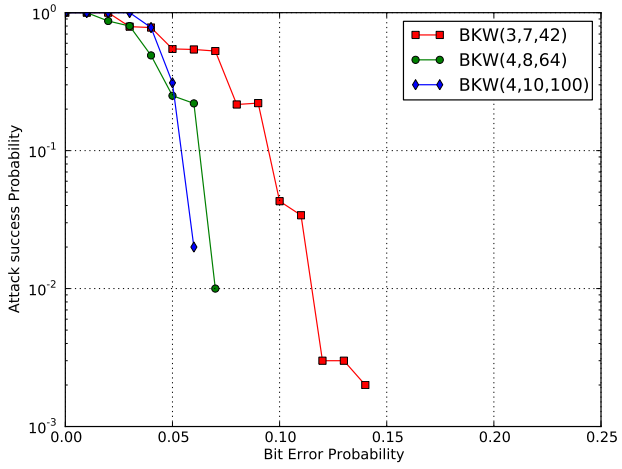

Fig. 6. BKW solution for the LPN Problem.

of $\mathbf{v}$ is $k = ab$ and that G is a $k \times n$ matrix. The parameters a and b are, respectively, the block count and block size in the notation of [8]. Each entry of G was chosen to be a uniformly random bit. We notice that success probability becomes poorer with an increase in k. Beyond $k = 64$, the BKW attack is exceedingly complex to run, if not impossible. Typically, n needs to be of the order of $a2^b$ for the attack to be possible.

Fig. 7 shows results of a known-plaintext attack with a modified version of the BKW algorithm. The BKW algorithm [8] is used to find the estimate $\hat{\mathbf{v}}$ from the equation $\mathbf{y}' = \mathbf{v}G + \mathbf{e}$, where G is a (3,6)-regular 21×42 LDPC matrix (3 1s per

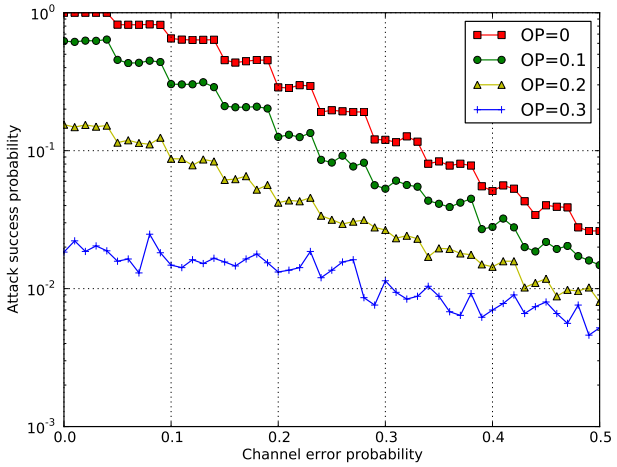

Fig. 7. Fast correlation attack with BKW preprocessor.

column, and 6 1s per row). The block count and size of BKW were set as $a = 3$ and $b = 7$. To improve success with the BKW algorithm, we first convert the equation $\mathbf{y}' = \mathbf{v}G + \mathbf{e}$ to $\mathbf{y}'' = \mathbf{v}G' + \mathbf{e}'$, where G' is a $21 \times \binom{42}{2}$ matrix that contains all pairwise XORs of the columns of G, and $\mathbf{y}''$ is a length-$\binom{42}{2}$ vector that contains the corresponding XORs of $\mathbf{y}'$. The idea, which was used in [9], is to increase the number of output bits used by the BKW algorithm at the cost of a marginally higher error rate in $\mathbf{e}'$. The same LFSR of length 8 with connection polynomial $x^8 + x^7 + x^5 + x^3 + 1$ was used for generating $\mathbf{v}$. The estimate $\hat{\mathbf{v}}$ from the modified BKW algorithm was subjected to a fast correlation attack with $B = 4$. The success probability of finding the initial seed is plotted as a function of channel error probability in Fig. 7 for various values of obfuscation probabilities.

From Fig. 7, we see that the modified BKW preprocessor is much more successful when compared to the Carrijo attack even for high values of channel error and obfuscation probabilities. In fact, the similarity of the two figures, Fig. 4 and Fig. 7, shows that the BKW attack is largely successful in negating the effect of the multiplication by G. However, for larger blocklengths the BKW attack is infeasible, while the Carrijo attack can still be launched.

In summary, we have seen that known-plaintext attacks are successful in finding the initial key for small LFSR register lengths and small lengths of random vector. On the basis of the attacks, we conclude that LFSR register lengths of 8 (or lower) combined with random vector lengths of 60 (or lesser) result in possible known-plaintext attacks with a nonzero probability of success. With higher computational capabilities longer register lengths might become attackable. The refresh rate for the initial seed that generates the random vector can be decided based on the probability of success.

Finally, to attack the message in the unknown-plaintext phase (see Section III), another LPN problem (finding $\mathbf{m}$ from $\mathbf{m}G^* + \mathbf{e}$) still remains to be solved. The probability of success for this second LPN will follow the plot in Fig. 6.

VI. Concluding Remarks

In this work, we studied security issues in LFSR-based pseudo-random number generators for coset encoding for the binary symmetric wiretap channel. The tools used for attacks are the fast correlation attack and solutions to the learning parity in noise (LPN) problem. For a long enough blocklength, where strong attacks against LPN are infeasible, we show that wiretap encoding provides significant protection against fast correlation attacks in the known-plaintext scenario. Simulation results show that, under wiretap encoding, the success probability of combined LPN and fast correlation attacks drops to the probability of success of a random guess beyond a certain threshold probability of error for the wiretapper's channel. Therefore, beyond this threshold, key refresh rates are controlled purely by the register length of the LFSR.

References

[1] A. D. Wyner, "The wire-tap channel," *Bell System Technical Journal*, vol. 54, no. 8, pp. 1355–1387, 1975.

[2] I. Csiszar and J. Korner, "Broadcast channels with confidential messages," *IEEE Trans. on Info. theory*, vol. 24, no. 3, pp. 339–348, May 1978.

[3] W. Harrison and S. McLaughlin, "Physical-layer security: Combining error control coding and cryptography," in *Communications, 2009. ICC '09. IEEE International Conference on*, June 2009, pp. 1 –5.

[4] ——, "Tandem coding and cryptography on wiretap channels: EXIT chart analysis," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, Jun 28 - July 3 2009 2009, pp. 1939 –1943.

[5] A. Thangaraj, S. Dihidar, A. Calderbank, S. McLaughlin, and J.-M. Merolla, "Applications of LDPC codes to the wiretap channel," *Information Theory, IEEE Transactions on*, vol. 53, no. 8, pp. 2933 –2945, Aug 2007.

[6] A. Blum, M. Furst, M. Kearns, and R. J. Lipton, "Cryptographic primitives based on hard learning problems," in *Proc. Advances in cryptology*. New York, NY, USA: Springer-Verlag New York, Inc., 1994, pp. 278–291.

[7] N. J. Hopper and M. Blum, "A secure human-computer authentication scheme," Carnegie Mellon University, Tech. Rep., 2000.

[8] A. Blum, A. Kalai, and H. Wasserman, "Noise-tolerant learning, the parity problem, and the statistical query model," *J. ACM*, vol. 50, pp. 506–519, July 2003.

[9] É. Levieil and P.-A. Fouque, "An improved LPN algorithm." in *SCN'06*, 2006, pp. 348–359.

[10] E. R. Berlekemp, R. J. McEliece, and H. C. A. V. Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. on Info. theory*, vol. 24, no. 3, pp. 384–386, May 1978.

[11] B. Applebaum, B. Barak, and A. Wigderson, "Public-key cryptography from different assumptions," in *Proceedings of the 42nd ACM symposium on Theory of computing*, ser. STOC '10. New York, NY, USA: ACM, 2010, pp. 171–180.

[12] V. V. Chepyzhov and B. J. M. Smeets, "On a fast correlation attack on certain stream ciphers," in *EUROCRYPT*, 1991, pp. 176–185.

[13] T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *Computers, IEEE Transactions on*, vol. C-34, no. 1, pp. 81 –85, Jan 1985.

[14] W. Meier and O. Staffelbach, "Fast correlation attacks on certain stream ciphers," *Journal of Cryptology*, vol. 1, no. 3, pp. 159–176, Oct 1989.

[15] T. Johansson and F. Jönsson, "Improved fast correlation attacks on stream ciphers via convolutional codes," in *EUROCRYPT*, 1999, pp. 347–362.

[16] C. S. Bruwer, "Correlation attacks on stream ciphers using convolutional codes," Master's thesis, University of Pretoria, 2005.

[17] J. Carrijo, R. Tonicelli, H. Imai, and A. C. A. Nascimento, "A novel probabilistic passive attack on the protocols HB and HB+," *IEICE Transactions*, vol. 92-A, pp. 658–662, 2009.

[18] S. Lin and D. J. Costello, *Error Control Coding, Second Edition*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2004.