Tatra Mt. Math. Publ. **37** (2007), 75–91

TATRA MOUNTAINS
Mathematical Publications

CRYPTANALYSIS OF THE BLOCK CIPHER BASED ON THE HOPFIELD NEURAL NETWORK

Dubravko Culibrk — Daniel Socek — Michal Sramka

ABSTRACT. A Hopfield neural network exhibits the phenomenon of stochastic error in convergence. Based on this non-determinism, Guo-Cheng-Cheng proposed a symmetric block cipher. Cryptanalysis of this cryptosystem led to an interesting mathematical problem - given two matrices that are conjugate of each other by a permutation matrix, find such permutation matrix. We present cryptanalysis of the proposed block cipher, as well as a practical method for solving the mentioned problem in some instances. The key space and, therefore, the security of the block cipher is significantly reduced, when the proposed method can be effectively employed.

1. Introduction

In a Hopfield neural network, an input message converges to one of the special messages called attractors. A Hopfield neural network exhibits stochastic error in convergence. The errors in the convergence are significantly increased in the case of an overstoraged network. In particular, the messages in the attraction domain of an attractor are unpredictably related to each other.

Based on these facts, D. Guo, L. M. Cheng and L. L. Cheng proposed a block cipher [5]. In this paper, we examine the security and efficiency of the proposed cryptosystem. Furthermore, the cryptanalysis leads to an interesting mathematical problem: Given two matrices that are conjugate of each other by a permutation matrix, determine this permutation matrix. This problem is a specific case of the Conjugacy Search Problem over matrix groups where the conjugate element must be a permutation matrix, and so it is also closely

2000 Mathematics Subject Classification: 68P25, 94A60, 14G50.
Keywords: cryptanalysis, block cipher, Hopfield neural network, conjugation by a permutation matrix.
This work was partially supported by a Federal Earmark grant for *Research in Telecommunication Networks* (2004-05).

related to the Graph Isomorphism Problem. Although it is widely believed that these problems cannot be solved in polynomial time, we present a method which significantly reduces the complexity of our problem in some instances. The method represents a serious liability in terms of the cryptosystem analyzed.

2. Hopfield Neural Network

Let us first consider a fully interconnected synchronous *neural network* of N neurons (labeled $0, 1, \ldots, N-1$). The state of a neuron i at a time t is denoted $S_i(t)$, and the initial state is denoted $S_i(0)$. The next state of neuron i depends on the current states of all neurons as follows

$$S_i(t+1) = f\left(\sum_{j=0}^{N-1} S_j(t)T_{ij} + \vartheta_i\right),$$

where T_{ij} is the synaptic strength between neurons i and j, ϑ_i is the threshold value of neuron i, and $f(\cdot)$ is any non-linear function.

The *Hopfield Neural Network* (HNN) is a neural network with zero neuron threshold (i.e., $\vartheta_i = 0$ for every $i = 1, 2, \ldots, N-1$), and with $T = (T_{ij})$ being a symmetric matrix. J. J. Hopfield proved in [6] that such a network possesses the property that each initial state of the network converges to a stable state (equivalently, there are no cycles in the state space of a HNN). We call stable states *attractors*.

From now on, we will consider the Hopfield neural network to be *discrete* (i.e., $S_i(t) \in \{0, 1\}$) and *clipped* (i.e., $T_{ij} \in \{-1, 0, 1\}$). The non-linear function f will be the sign function $\sigma(\cdot)$, where

$$\sigma(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{if } x < 0. \end{cases}$$

The state of the network at time t can be expressed as a row vector

$$S(t) = \big(S_0(t), S_1(t), \ldots, S_{N-1}(t)\big).$$

We call an initial state $S(0)$ a *message*. If we let the function $\sigma(\cdot)$ act on vectors elementwise, we can express the next state formula of a Hopfield neural network in matrix form as follows:

$$S(t+1) = \sigma\big(S(t)T\big).$$

We will denote a discrete clipped HNN with sign function $\sigma(\cdot)$ and synaptic strength matrix T by HNN_T. We say the HNN_T is *overstoraged* [5] if the number of attractors of this network is $2N$ or more. Overstoraged HNNs exhibit an interesting property of converging in few iterations. Specifically, the larger the number of attractors the more likely it is that a random input message to the network will cause it to converge in a single step, namely

$$S(t) = S(t+1) \quad \text{for} \quad t \geq 1.$$

This property is likely due to the large number of attractor domains spread out over the entire message space, causing a random input message to be close (in terms of Hamming distance) to its attractor. The networks with a large number of attractors or nearly equidistant attractors are more likely to converge in a single step for any input.

Neural networks have non-linear mapping property. In addition, they possess an interesting stochastic error property. In particular, the relation of any initial state $S(0)$ to its attractor (stable state) is irregular-non-deterministic [5]. For the maximum irregularity, the network requires equal concentrations of excitatory ($T_{ij} = 1$) and inhibitory ($T_{ij} = -1$) synapses. That is,

$$\sum_i T_{ij} = 0 \quad \text{and} \quad \sum_j T_{ij} = 0. \tag{2.1}$$

It is noted that the following property can be easily proved in a network, where each initial state leads to an attractor in at most one step: Let x be any attractor of HNN_T, and let $D_x = \{y \mid x = \sigma(yT)\}$ be the domain of attraction for x. Moreover, let P be an $N \times N$ permutation matrix, and let $\hat{T} = PTP^{-1}$. Then xP^{-1} is an attractor of $\mathrm{HNN}_{\hat{T}}$ and $\{yP^{-1} \mid y \in D_x\}$ is its domain of attraction.

It remains to say that the HNN_T neural network model (also called associative memory network) is suitable for fast hardware implementations, since the implementation and execution can easily be performed in parallel.

3. Encryption Scheme

We give a brief description of the *Symmetric Probabilistic Encryption Scheme Based on the Chaotic Attractors* [5]. The description is somewhat simplified without changing the properties important for this paper. See the original proposal for full details.

Fix N and fix a $N \times N$ matrix T over $\{-1, 0, 1\}$ such that #"1" = #"-1" $\approx$ #"0" $\approx N/3$ (where # means "the number of") in each row and each column of T (e.g., it was suggested to use symmetric circulant matrices). The matrix T

must be selected in such a way that the resulting Hopfield neural network HNN_T with synaptic strength matrix T and sign function $\sigma(\cdot)$ will be overstoraged.

Key selection: Choose an $N \times N$ permutation matrix H and compute new synaptic matrix $\hat{T} = HT\widetilde{H}$, where $\widetilde{H}$ denotes the transpose of H. Keyspace is of size $N!$.

Plaintext space: A subset of the attractors of the overstoraged $\mathrm{HNN}_{\hat{T}}$ network. Each attractor is a binary vector of length N.

Encryption: For a given plaintext x, randomly choose a binary vector y of length N from the domain of attraction of x. The vector y will become the ciphertext. Since there are many ciphertexts corresponding to one plaintext and the selection is random, the encryption is probabilistic (in some sense).

Decryption: For a given ciphertext y (a binary vector of length N), set the initial state of $\mathrm{HNN}_{\hat{T}}$ to be the given ciphertext. Then the $\mathrm{HNN}_{\hat{T}}$ network will converge to the corresponding plaintext x. That is, in the network $\hat{S}(t+1) = \sigma\big(\hat{S}(t)HT\widetilde{H}\big)$ with initial state $\hat{S}(0) = y$, as $t \to \infty$, $\hat{S}(t) \to x$.

4. Critique, Efficiency, Attacks

(1) For the computational security of the cryptosystem, only those attractors of $\mathrm{HNN}_{\hat{T}}$ with a large domain of attraction should be considered. One immediate problem is to determine, which attractor has a domain of attraction large enough such that computational searches over this domain are infeasible. By a property from Section 2, there is a one-to-one correspondence between domains of attraction of HNN_T and $\mathrm{HNN}_{\hat{T}}$. Hence, if the cardinalities of domains of attraction of HNN_T are known in advance (precomputed), this problem is negligible. In other words, the problem of the cardinalities of domains of attractions is the same for HNN_T and $\mathrm{HNN}_{\hat{T}}$.

(2) Another immediate problem is the realization of an encryption. For a given plaintext (attractor) x, one needs to find y in the domain of attraction of x. This can be accomplished in two ways. Either the domains of attraction are tablularized or the suitable ciphertext will be determined by random searching through the whole binary vector space of dimension N. The first approach requires huge storage and extensive pre-computations, and therefore is not practical for reasonably secure values of N. The second approach works as follows: For a given x, repeatedly and randomly

choose a binary vector y of length N and test whether y converges to x in $\mathrm{HNN}_{\hat{T}}$. Thus a single encryption requires several decryptions, making the encryption process too slow. The number of decryptions while encrypting depends on the number of plaintexts and the cardinalities of the domains of attraction.

(3) Probabilistic encryption: the cryptosystem is not really probabilistic from the traditional cryptology point of view. The decryption function from the ciphertext space to the plaintext space is deterministic. One ciphertext always corresponds to the same plaintext, compared to, e.g., ElGamal or Vigenère, where one ciphertext can be any plaintext.

(4) Ciphertext-only attack: Considering the size of the plaintext space for a given N and the type of probabilistic encryption described above, a classical statistical (frequency) analysis can be used to obtain open messages for reasonable values of N.

(5) A chosen-ciphertext attack is possible, as described in the next Section.

5. Chosen-ciphertex attack

In this scenario, an attacker should be able to choose all N weight 1 ciphertext messages (such that when put into a matrix as rows they will form an identity matrix) and obtain the corresponding plaintext messages.

Mathematically speaking, let e_i denote the N-bit vector with 1 at position i and 0's at all other positions (i.e., e_i is the i-th row of an $N \times N$ identity matrix). For each ciphertext e_i $(i = 1, \dots, N)$ the attacker should be able to obtain the corresponding plaintext v_i. Consider the scenario, where all the messages e_i's converge to an attractor in the $\mathrm{HNN}_{\hat{T}}$ network in just one step. In this case, for all $i = 1, \dots, N$,

$$v_i = \sigma\big(e_i\hat{T}\big) = \sigma\big(e_iHT\widetilde{H}\big) = \sigma\big(e_iHTH^{-1}\big) = e_iH\sigma(T)H^{-1}.$$

The second to the last equality follows from the fact that H is a permutation matrix and so $\widetilde{H} = H^{-1}$, and the last equality follows from the definition of the sigma function (here acting on matrices elementwise) and the fact that the weight of e_i is 1 and so is the weight of any row/column of H and H^{-1}. By letting $V = (v_i)$ and $I = (e_i)$, we obtain the equation in the matrix form

$$V = \sigma\big(I\hat{T}\big) = \sigma\big(IHT\widetilde{H}\big) = \sigma\big(HTH^{-1}\big) = H\sigma(T)H^{-1}, \tag{5.1}$$

where V is known (obtained by the attacker) and $\sigma(T)$ can be easily computed, since T is public. In the next Section, we show how this equation can be solved for unknown permutation matrix H. Since the solution (the permutation matrix H) does not have to be unique, we either obtain the secret key or its equivalent.

If one or more of the messages e_i's do not converge to an attractor in one step, denote m as the maximum number of iterations for each e_i to converge to an attractor. Now, one iteration of $\hat{S}(0) = I$ can be described as

$$\hat{S}(1) = \sigma\big(\hat{S}(0)\hat{T}\big) = \sigma\big(IHTH^{-1}\big) = H\sigma(T)H^{-1},$$

two iterations as

$$\hat{S}(2) = \sigma\big(\hat{S}(1)\hat{T}\big) = \sigma\big(H\sigma(T)H^{-1} \cdot HTH^{-1}\big) = H\sigma\big(\sigma(T)T\big)H^{-1},$$

three iterations as

$$\hat{S}(3) = \sigma\big(\hat{S}(2)\hat{T}\big) = \sigma\left(H\sigma\big(\sigma(T)T\big)H^{-1} \cdot HTH^{-1}\right) = H\sigma\left(\sigma\big(\sigma(T)T\big)T\right)H^{-1},$$

and in a similar way we can inductively determine the exact expression for k-th iteration $\hat{S}(k)$. In particular, if we define $W_1 = \sigma(T)$ and recursively $W_k = \sigma(W_{k-1}T)$ then we have

$$\hat{S}(k) = HW_kH^{-1}.$$

Once the network converges to an attractor state, the state does not change with further iterations. If a message converges to an attractor in r iterations then it also converges to this attractor after more than r iterations. Therefore the attacker can choose a sufficiently large number k (assuming $k \geq m$, and m can be determined from the public matrix T) and the attacker will obtain

$$\hat{S}(k) = HW_kH^{-1},$$

where $\hat{S}(k)$ will be the matrix V which was obtained and W_k can be computed from public values. Note that the value of m can be easily determined by an attacker. Hence, the attacker ends once again with an equation $V = HW_kH^{-1}$ with known matrices V and W_k and an unknown permutation matrix H. Moreover, if the matrix T is circulant, then so is W_i for every integer $i \geq 1$. Given any matrix W_i, finding a permutation H in this equation is known to be difficult (think Conjugacy Search Problem in an arbitrary group). However, for the key-sizes discussed in [5], the following technique could potentially be used to reduce the key space and make the exhaustive search feasible.

6. Conjugation by a permutation matrix problem

Definition. By a *conjugation by a permutation matrix problem* or simply *PROBLEM* we will understand the following: Given two matrices A and B

that are conjugate of each other by a permutation matrix P, that is,

$$B = PAP^{-1}, \tag{6.1}$$

determine the permutation matrix P.

In the following paragraphs, we show how to efficiently approach this PROBLEM in the case where matrix A is the circulant matrix T mentioned in the previous Sections, or more generally, where matrix A is a doubly-stochastic matrix. Although we do not have a polynomial time algorithm for this PROBLEM, our method significantly reduces its complexity. For example, if A is the mentioned matrix T of size 32×32, then the cryptosystem has $32! \approx 2^{117}$ keys. For practical purposes and for the purpose of breaking the cryptosystem, it suffices to reduce the complexity to less than, say, 2^{40} and then perform an exhaustive search (still not real-time time, but doable in a reasonable time on a PC).

Notation. A natural map from the permutation group S_N to the group of permutation matrices is the following one: Let $\pi \in S_N$. Then the image of π in the group of permutation matrices is the following $N \times N$ matrix P_π:

$$P_\pi = \begin{pmatrix} e_{\pi(1)} \\ \vdots \\ e_{\pi(N)} \end{pmatrix} = \begin{pmatrix} e_{\pi^{-1}(1)} & \cdots & e_{\pi^{-1}(N)} \end{pmatrix},$$

where e_i represents the i-th row of the identity matrix. Note that this implies that the matrix P_π^T is the following $N \times N$ matrix:

$$P_\pi^T = \begin{pmatrix} e_{\pi^{-1}(1)} \\ \vdots \\ e_{\pi^{-1}(N)} \end{pmatrix} = \begin{pmatrix} e_{\pi(1)} & \cdots & e_{\pi(N)} \end{pmatrix} = P_\pi^{-1} = P_{\pi^{-1}}.$$

For the simplicity of notation we use P instead of P_π.

Theorem 1. *Let π be a permutation in S_N from which a permutation matrix P is constructed via the natural map. If A in the equation (6.1) contains an element at position (i,j), then this element is moved in matrix B to position $(\pi^{-1}(i), \pi^{-1}(j))$.*

Proof. Let $a_{(i,j)}$ denote the entry of A at position (i,j), c_i the i-th column of A, and r_i the i-th row of A. Then,

$$PA = \begin{pmatrix} e_{\pi(1)} \circ c_1 & \cdots & e_{\pi(1)} \circ c_N \\ \vdots & \ddots & \vdots \\ e_{\pi(N)} \circ c_1 & \cdots & e_{\pi(N)} \circ c_N \end{pmatrix} = \begin{pmatrix} a_{(\pi(1),1)} & \cdots & a_{(\pi(1),N)} \\ \vdots & \ddots & \vdots \\ a_{(\pi(N),1)} & \cdots & a_{(\pi(N),N)} \end{pmatrix},$$

and similarly,

$$AP^T = \begin{pmatrix} r_1 \circ e_{\pi(1)} & \cdots & r_1 \circ e_{\pi(N)} \\ \vdots & \ddots & \vdots \\ r_N \circ e_{\pi(1)} & \cdots & r_N \circ e_{\pi(N)} \end{pmatrix} = \begin{pmatrix} a_{(1,\pi(1))} & \cdots & a_{(1,\pi(N))} \\ \vdots & \ddots & \vdots \\ a_{(N,\pi(1))} & \cdots & a_{(N,\pi(N))} \end{pmatrix},$$

where $\circ$ denotes the usual dot product.

Thus, $a_{(i,j)}$ gets replaced by $a_{(\pi(i),\pi(j))}$ as a result of multiplication by P on the left and by P^T on the right. This is equivalent of saying that operation PAP^T "moves" $a_{(i,j)}$ to $a_{(\pi^{-1}(i),\pi^{-1}(j))}$. $\square$

Corollary 1. *Let $a_{(i_1,j_1)}, \dots, a_{(i_m,j_m)}$ be unique elements of A (and consequently B) from the equation (6.1). If set $\{i_1, j_1, \dots, i_m, j_m\} = \{1, \dots, N\}$, all moves of π are known.*

Proof. Let $a_{(i_1,j_1)}, \dots, a_{(i_m,j_m)}$ be unique elements of A. Then, by Theorem 1, $a_{(i_k,j_k)}$ is moved from position (i_k, j_k) in A to a position $(\pi^{-1}(i_k), \pi^{-1}(j_k))$ in B, for all $k \in \{1, \dots, m\}$. Since an element $a_{(i_k,j_k)}$ is unique in both A and B, the values $i_k, j_k, \pi^{-1}(i_k)$ and $\pi^{-1}(j_k)$ are all known. Thus, if $\{i_1, j_1, \dots, i_m, j_m\} = \{1, \dots, N\}$, then $\pi^{-1}(1), \dots, \pi^{-1}(N)$ are known. $\square$

Hence if the matrix A contains unique elements, then we can apply Theorem 1 and Corollary 1 directly. On the other hand, if A (and consequently B) does not contain unique elements (as is the case with equation (5.1), where V and $\sigma(T)$ are matrices over 0 and 1), then the matrix A^n (and consequently B^n) for some positive integer n likely does. Here, note that

$$B^n = (PAP^{-1})^n = PA^nP^{-1}.$$

We use this "powering" method to obtain unique elements and apply Theorem 1 and Corollary 1 to obtain the moves of the unknown permutation π from which the matrix P is constructed. The remaining moves can be obtained by exhaustive search. The reader should note that the "powering" method can significantly narrow down the exhaustive search space, even when the matrix A has strong symmetry (e.g., circulant matrix).

It should be noted here that powering both sides of the equation (6.1) may produce an expanded solution space. In other words, a solution to the PROBLEM is also a solution to $B^n = PA^nP^T$, but not necessarily vice versa. Once a solution is found to the powered equation, the following helps determining all solutions to the powered equation from which a subset of solutions to the PROBLEM is easily computable.

Firstly, suppose that T is a circulant matrix (as suggested in [5]).

Notation. If Q is a permutation matrix, let $C(Q)$ denote the centralizer of Q over the group of permutation matrices. More generally, if A is an $n \times n$ matrix, let $C(A)$ denote a set of all $n \times n$ permutation matrices that commute with A.

Theorem 2. *If permutation matrix Q is a solution to the equation (6.1), then QR is also a solution, for all $R \in C(A)$.*

P r o o f. Let Q be a solution to the equation (6.1), i.e., $B = QAQ^{-1}$. Let R be any element from $C(A)$. Then $AR = RA \Rightarrow A = RAR^{-1}$. Therefore QR is also a solution to (6.1) since

$$B = QAQ^{-1} = QRAR^{-1}Q^{-1} = (QR)A(QR)^{-1}. \qquad \square$$

Theorem 3. *Let $\sigma(T)$ be a circulant matrix, and U be a permutation matrix corresponding to the cycle $(1\,2\,\dots\,N)$. If a permutation matrix Q commutes with U, then Q commutes with $\sigma(T)$.*

P r o o f. By a result from elementary linear algebra [3], if $\sigma(T)$ is a right circulant matrix with the first row $c_0 c_1 \dots c_{N-1}$, then

$$\sigma(T) = \sum_{i=0}^{N-1} c_i U^i.$$

Take permutation matrix $Q \in C(U)$. Then Q commutes with $\langle U \rangle$ (a cyclic group generated by U). Therefore Q commutes with $\sigma(T)$ since

$$Q\sigma(T) = c_0 Q U^0 + \dots + c_{N-1} Q U^{N-1} = c_0 U^0 Q + \dots + c_{N-1} U^{N-1} Q = \sigma(T)Q. \qquad \square$$

By Theorems 2 and 3, if a single solution P to the equation (6.1) is obtained when A is circulant, N distinct solutions are immediately available, namely PQ, where $Q \in \langle U \rangle$. Further solutions can be obtained by exhaustive search of the solution space reduced with our powering method.

More generally, suppose T is an arbitrary matrix, not necessarily circulant, but still conforming to the rules in equation (2.1). In this case, similar, but more complex approach could be used.

Notation. A *doubly stochastic matrix* is a matrix over real numbers in which all entries are non-negative and each row and each column adds to exactly 1.

We state the following classical theorem of B i r k h o f f [1]:

Theorem 4 (Birkhoff)**.** *Matrix A is doubly stochastic if and only if it is a convex combination of permutation matrices. That is, A is doubly stochastic if and only if $A = f_1R_1 + f_2R_2 + \dots + f_kR_k$ for some permutation matrices R_i and some nonnegative numbers f_i whose sum equal to 1.*

Theorem 5. *Let T be a matrix with row sums and column sums equal to 0 and row sums and column sums of $\sigma(T)$ equal to m ($m \neq 0$). Let $M = \frac{1}{m}\sigma(T) = f_1R_1 + f_2R_2 + ... + f_kR_k$. If a permutation matrix Q commutes with every R_i ($i = 1, \ldots, k$), then Q commutes with $\sigma(T)$. Therefore, if P_0 is a solution to the equation (5.1) then any $P \in \big(C(R_1) \cap C(R_2) \cap \ldots \cap C(R_k)\big)P_0$ is also a solution.*

Proof. Matrix T conforms to the equations in (2.1). By definition of $\sigma(\cdot)$, all row sums and all column sums of $\sigma(T)$ are the same, say m. Then the matrix $M = \frac{1}{m}\sigma(T)$ is doubly stochastic. Thus by Theorem 4, $M = f_1R_1 + f_2R_2 + \ldots + f_kR_k$ as stated. Suppose permutation matrix Q commutes with all R_i ($i = 1, \ldots, k$). Then Q commutes with M since

$$QM = f_1QR_1 + \ldots + f_kQR_k = f_1R_1Q + \ldots + f_kR_kQ = MQ,$$

and therefore Q commutes with $mM = \sigma(T)$. Finally, if Q commutes with all R_i ($i = 1, \ldots, k$), then Q is in the intersection of the centralizers of R_i's. $\square$

By Theorem 5, if one obtains a solution to the equation (6.1), as well as the intersection of the centralizers of the permutation matrices from the Birkhoff's decomposition from Theorem 4, one has obtained t solutions to the equation (6.1), where t denotes the cardinality of the intersection. The Birkhoff's decomposition time-complexity is $O(N^{4.5})$, assuming improved Birkoff-von Neumann algorithm by Dulmage and Halperin [4]. Once we have decomposition, we can calculate the centralizers using polynomial-time approaches such as the ones by Sims [8] or Buttler [2] that have time-complexities of about $O(N^{3.5})$. Even though generally the intersection of subgroups problem is considered intractable in the sense that there is no polynomial-time algorithm in N available, this problem is solvable for the relatively small permutation groups that are suggested for use in [5]. In most experiments we performed, this intersection was just the identity.

It must be noted that the converse of Theorem 5 is not necessary true in general – that is, there may exist a permutation matrix P which commutes with $\sigma(T)$ but does not commute with every R_i. Since T is a fixed matrix, and so is $\sigma(T)$, the automorphism group of an appropriate graph would consist of all the permutation matrices that commute with $\sigma(T)$. Hence this graph theoretic approach (efficient, although not polynomial) can be used as an alternative to the method described in the previous paragraph.

7. Examples of our chosen-ciphertext attack

In [5], the authors claim that the size of the search space for finding a secret permutation H is $N!$, which requires 10^{26} MIPS years when $N = 32$ for a successful search on a dedicated computer capable of testing 10^6 permutations per second, which is well above the currently acceptable security level of 10^{12} MIPS years. We disagree with that statement, and demonstrate that stronger keys with possibly much larger key-sizes must be ensured in order to provide sufficient security.

7.1. Example of breaking a system when $N = 8$

Consider the same example for $N = 8$ as discussed in [5]. Following the attack described in Section 5, the adversary obtains these values:

$$T = \begin{pmatrix} 1 & 1 & 0 & -1 & -1 & -1 & 0 & 1 \\ 1 & 1 & 1 & 0 & -1 & -1 & -1 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 \\ -1 & 0 & 1 & 1 & 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 & 1 & 1 & 0 & -1 \\ -1 & -1 & -1 & 0 & 1 & 1 & 1 & 0 \\ 0 & -1 & -1 & -1 & 0 & 1 & 1 & 1 \\ 1 & 0 & -1 & -1 & -1 & 0 & 1 & 1 \end{pmatrix},$$

$$\sigma(T) = A = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}, \quad \text{and}$$

$$\sigma(HTH^{-1}) = H\sigma(T)H^{-1} = B = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}.$$

The goal is to obtain the unknown secret permutation matrix H, or equivalently, its corresponding natural map $h \in S_N$. Note that the complexity of finding H is already not 8!, but smaller since some of the "moves" can clearly be ruled out when comparing A and B. The aforementioned powering method can be used to reduce the search space even further. In this particular example, the powering method leads to a maximum number of distinct entries for exponent $k \geq 3$. For $k = 3$ we have:

$$A^3 = \begin{pmatrix} 19 & 18 & 16 & 13 & 12 & 13 & 16 & 18 \\ 18 & 19 & 18 & 16 & 13 & 12 & 13 & 16 \\ 16 & 18 & 19 & 18 & 16 & 13 & 12 & 13 \\ 13 & 16 & 18 & 19 & 18 & 16 & 13 & 12 \\ 12 & 13 & 16 & 18 & 19 & 18 & 16 & 13 \\ 13 & 12 & 13 & 16 & 18 & 19 & 18 & 16 \\ 16 & 13 & 12 & 13 & 16 & 18 & 19 & 18 \\ 18 & 16 & 13 & 12 & 13 & 16 & 18 & 19 \end{pmatrix}, \quad \text{and}$$

$$B^3 = \begin{pmatrix} 19 & 13 & 16 & 13 & 16 & 12 & 18 & 18 \\ 13 & 19 & 13 & 16 & 18 & 18 & 12 & 16 \\ 16 & 13 & 19 & 18 & 12 & 16 & 18 & 13 \\ 13 & 16 & 18 & 19 & 13 & 18 & 16 & 12 \\ 16 & 18 & 12 & 13 & 19 & 16 & 13 & 18 \\ 12 & 18 & 16 & 18 & 16 & 19 & 13 & 13 \\ 18 & 12 & 18 & 16 & 13 & 13 & 19 & 16 \\ 18 & 16 & 13 & 12 & 18 & 13 & 16 & 19 \end{pmatrix}.$$

The attacker assumes all $N = 8$ possibilities for moving the first row from A^3 to B^3. If it is assumed that row 1 is mapped to row i then $1 \to i$ under h^{-1}. Then, possibilities for the remaining $N - 1$ entries can be exhausted. In this case, we have $1 \times 2 \times 2 \times 2$ possibilities for each N assumptions, a total of $8 \times 1 \times 2 \times 2 \times 2 = 64$ possibilities to test, as opposed to a false estimation of 8!. One of these 64 possibilities is $h^{-1} = 6\ 4\ 3\ 7\ 1\ 8\ 5\ 2$, which reveals the secret key $h = 5\ 8\ 3\ 2\ 7\ 1\ 4\ 6$, or:

$$H = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

7.2. Example of breaking a system when $N = 32$ and T is circulant

Let $cir(v)$ denote the $n \times n$ circulant matrix generated from n-dimensional vector v in the following way: v is taken as the top row, while each further row is obtained by circularly shifting the previous row by one to the left. Consider the following cryptosystem parameters: $N = 32$, $v = (1, 1, 1, 0, -1, -1, -1, 0, 1, 1, 0, -1, 0, -1, 0, 0, -1, 1, -1, 1, 0, -1, 1, 0, 1, -1, 0, 0, -1, -1, 1, 1)$ chosen at random from all the vectors over $\{-1, 0, 1\}$ with exactly 11 1's, 11 -1's, and 10 0's, $T = cir(v)$, $\sigma(T) = A = cir(1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1)$, $B = HAH^T$, where H corresponds to a randomly chosen $h =$ 32 20 28 17 9 12 2 4 27 21 8 13 6 26 5 23 16 25 10 18 3 1 7 14 11 19 30 31 15 29 22 24 $\in S_{32}$. It is assumed that an adversary knows A and B, but not H. In this example, by using the powering method with an exponent $k \geq 7$, an adversary can obtain a pair A^k and B^k with 32 distinct entries per row. Thus an exhaustive search space is reduced to a mere 32 possibilities. By testing the possibility $1 \to 22$, an adversary obtains the correct value of $h^{-1} =$ 22 7 21 8 15 13 23 11 5 19 25 6 12 24 29 17 4 20 26 2 10 31 16 32 18 14 9 3 30.

7.3. Example of breaking a system when $N = 32$ and T is non-circulant

Consider the following cryptosystem parameters: $N = 32$, $v = (1, 0, -1, 0, -1, 1, 1, 0, -1, 0, -1, 1, 1, -1, 1, 0, 0, 0, -1, -1, 0, 1, 0, 1, 1, -1, 1, 0, -1, -1, 1, -1)$ chosen at random from all the vectors over $\{-1, 0, 1\}$ with exactly 11 1's, 11 -1's, and 10 0's, $R = cir(v)$, $T = PRQ$, $A = \sigma(T)$, $B = HAH^T$, where P, Q, H correspond to randomly chosen permutations:

32 20 28 17 9 12 2 4 27 21 8 13 6 26 5 23 16 25 10 18 3 1 7 14 11 19 30 31 15 29 22 24,
28 31 10 4 11 8 27 1 3 18 16 14 30 26 24 20 23 32 9 7 13 19 21 15 29 2 6 22 5 25 17 12,
15 3 26 7 29 19 17 8 21 13 9 18 32 28 1 10 24 11 27 14 31 5 2 4 6 12 25 20 23 30 16 22,
respectively. Again an attacker has values A and B, but not H. Table 1 shows the number of distinct entries in matrices A^k and B^k for different values of k in this particular example. In this case, all entries are distinct in A^{14} (or B^{14}), and using the fact from Corollary 1 an adversary immediately obtains all moves of H since there is only one possibility to test. Table 2 shows the complete recovery of the moves of H using only the elements from the first row of A^{14}: $h^{-1} =$ 15 23 2 24 22 25 4 8 11 16 18 26 10 20 1 31 7 12 6 28 9 32 29 17 27 3 19 14 5 30 21 13.

k	Num. of different values in A^k (or B^k)
1	2
2	9
3	24
4	57
5	135
6	299
7	556
8	790
9	933
10	986
11	1004
12	1016
13	1022
14 or more	1024

Table 1. Powering of a non-circulant 32×32 matrix which is one of the valid parameters for the cryptosystem defined in [5]. All distinct entries are achieved when $k \geq 14$.

7.4. Remarks

As the presented examples show, the described chosen-ciphertext attack is effective in practice for some of the keys with the key-sizes recommended in [5]. Some matrices are clearly "weaker-keys" than others, depending on how many different entries can be obtained with the powering method. If more entries can be obtained, the exhaustive search is further reduced more, and vice versa. Preliminary experiments suggest that highly symmetric matrices, such as circulant matrices, imply stronger keys in that sense than less symmetric matrices do. While we showed that weak keys exist in the scheme from [5], we do not discuss how to select strong keys of strength necessary in modern cryptography. A strong key must be such that it is resistant to our powering attack.

8. Conclusion

Although the characteristics and phenomenons of Hopfield neural networks are interesting, they do not always lead to straightforward cryptographic applications.

$A^{14}(i,j)$ and $B^{14}(s,t)$	(i,j)	(s,t)
101372497891456104	(1,1)	(15,15)
101372497891469693	(1,2)	(15,23)
101372497891244268	(1,3)	(15,2)
101372497891313792	(1,4)	(15,24)
101372497891277379	(1,5)	(15,22)
101372497891277584	(1,6)	(15,25)
101372497891295978	(1,7)	(15,4)
101372497890794209	(1,8)	(15,8)
101372497891437292	(1,9)	(15,11)
101372497891300967	(1,10)	(15,16)
101372497891543766	(1,11)	(15,18)
101372497891068543	(1,12)	(15,26)
101372497891373843	(1,13)	(15,10)
101372497891118348	(1,14)	(15,20)
101372497891392507	(1,15)	(15,1)
101372497891420529	(1,16)	(15,31)
101372497891337040	(1,17)	(15,7)
101372497891474894	(1,18)	(15,12)
101372497891156738	(1,19)	(15,6)
101372497891392528	(1,20)	(15,28)
101372497891190385	(1,21)	(15,9)
101372497891374364	(1,22)	(15,32)
101372497891024868	(1,23)	(15,29)
101372497891240304	(1,24)	(15,17)
101372497891315091	(1,25)	(15,27)
101372497891300552	(1,26)	(15,3)
101372497891287357	(1,27)	(15,19)
101372497891251478	(1,28)	(15,14)
101372497891296841	(1,29)	(15,5)
101372497891278963	(1,30)	(15,30)
101372497891382617	(1,31)	(15,21)
101372497891419859	(1,32)	(15,13)

TABLE 2. Matching 32 distinct entries in the first row of A^{14} to entries in B^{14} yields the unique solution for the unknown permutation.

We studied the 1999 paper of Guo-Cheng-Cheng that proposes the use of specific Hopfield neural network as a basis for a symmetric block cipher. It turns out that this approach, although worthwhile studying, is impractical in the terms of the speed of computation and communication. The cryptanalysis of various security aspects of this cipher revealed a few severe weaknesses. In particular, the cipher is vulnerable to ciphertext-only and chosen-ciphertext attacks. Thus, the analyzed cipher should not be used in applications where such attacks are feasible.

The chosen-ciphertext attack requires determining a hidden permutation matrix at some point. We proposed a practical solution for this *conjugation by a permutation matrix problem*. Namely, we developed a heuristics for reducing the complexity of this problem and described practical methods to obtain a set of permutation matrices (from just one permutation matrix) that are solution to the mentioned problem.

Acknowledgments

The authors would like to thank Professors Spyros S. Magliveras, Wandi Wei, and Tran van Trung for their helpful comments and suggestions. This work is partially supported from the *Center for Cryptology and Information Security* (CCIS) at Florida Atlantic University and by a Federal Earmark grant for *Research in Telecommunication Networks* (2004-2005).

REFERENCES

[1] BIRKHOFF, G.: *Three observations on linear algebra*, Univ. Nac. Tucumán. Rev. Ser. A **5** (1946), 147–151.

[2] BUTLER, G.: *An inductive schema for computing conjugacy classes in permutation groups*, Math. Comp. **62(205)** (1994), 363–383.

[3] DAVIS, P. J.: *Circulant Matrices* (2nd ed.), John Wiley & Sons, New York, 1994.

[4] DULMAGE, L.—HALPERIN, I.: *On a theorem of Frobenius-König and J. von Neumann's game of hide and seek*, Trans. Roy. Soc. Canada III(3) **49** (1955), 23–29.

[5] GUO, D.—CHENG, L. M.—CHENG, L. L.: *A new symmetric probabilistic encryption scheme based on chaotic attractors of neural networks*, Applied Intelligence **10**, (1999), 71–84.

[6] HOPFIELD, J. J.: *Neural networks and physical systems with emergent collective computational abilities*. In: Proc. Natl. Acad. Sci. USA 79, Mat. Acad. Sci. Washington, DC, 1982, pp. 2554–2558.

[7] JENNER, B.—KÖBLER, J.—MCKENZIE, P.—TORÁN, J.: *Completeness results for graph isomorphism*, J. Comput. Syst. Sci. **66** (2003), 549–566.

[8] SIMS, C. C.: *Determining the conjugacy classes of a permutation group.* In: Computer in Algebra and Number Theory (G. Birkhoff and M. Hall Jr., eds.), Proceedings of a Symposium in Applied Mathematics, New York, 1970. SIAM/AMS Proc., Vol. 4, Amer. Math. Soc. Providence, RI, 1971, pp. 191–195.

Received October 31, 2005

Dubravko Culibrk
Department of Computer Science and Engineering
Florida Atlantic University
777 Glades Road
Boca Raton, FL 33431
U.S.A.
E-mail: dculibrk@fau.edu

Daniel Socek
Michal Sramka
Center for Cryptology and Information Security
Department of Computer Science and Engineering
Florida Atlantic University
777 Glades Road
Boca Raton, FL 33431
U.S.A.
E-mail: dsocek@fau.edu
sramka@math.fau.edu