Attack on cascaded convolutional transducers cryptosystem

M. A. Orumiechiha, S. F. Mohebipoor

***Abstract*—**Recently, the idea of design of dynamic symmetric cryptosystems is proposed. The main idea is proposed by Trinca in two separate papers which are published in IEEE and eprint. Security of proposed ciphers is compared to Raijndal algorithm called AES. According to this scheme, the cipher system characteristic varies respect to secret key and input plaintext. Namely, for each certain secret key and plaintext, a different coding is provided and ciphertext is created. By changing the secret key or plaintext, other coding system is considered. But unfortunately this cipher is not safer than whatever is claimed. In this paper, we present two attacks on the proposed scheme. The first one is a partial key recovery which for a (k,k,m) q- cascaded convolutional transducers is determined a fragment of ciphertext without calculating master key with complexity $O(k^2)$. Also the attack needs about k^6 plaintext-ciphertext pairs to solve a linear equation system. The other attack is a weak key on this system that attacker can recover longer fragment of plaintexts with complexity $O(2\times k^2)$.

***Keywords*—** Convolution Codes, Cryptanalysis, Symmetric Cryptosystem, Sequential and parallel cascaded convolutional encryption.

I. INTRODUCTION

Worldwide symmetric encryption standards such as DES (Data Encryption Standard),AES (Advanced Encryption Standard), and EES (Escrowed Encryption Standard), have been– and some of them still are – extensively used to solve the problem of communication over an insecure channel, but with today's advanced technologies, they seem to not be as secure as one would like.

Symmetric encryption ciphers play an important role in practical cryptography. They are categorized as Stream Ciphers and Block Ciphers according to their applications.

In a stream cipher, the output keystream is bitwise XORed to the plaintext stream in order to obtain the ciphertext stream. They are utilized for encrypting long streams of data over fast communication applications such as GSM. Block ciphers are other fundamental primitives in cryptography. A block cipher can encrypt fixed length strings such that ciphertext is mixed in a nonlinear manner by secret key and plaintext. Applications in general require encryption of long and arbitrary length strings. A mode of operation of a block cipher is used to extend the domain of applicability from fixed length strings to long and variable length strings.

But symmetric cryptosystems such as triple DES , AES and others have been all designed as static ciphers, in the sense that their structure do not change at all during encryption/decryption. Recently, a dynamic symmetric cryptosystem [1,2] is proposed that whose structure is based on invertible convolution codes. The scheme is considered to two versions. The first one uses linear combinations of convolution codes and the other one exploits more complicated and nonlinear combinations. These symmetric encryption techniques have at least four advantages over traditional schemes based on Feistel ciphers. First, the secret key of a cascaded convolutional cryptosystem is usually much easier to generate. Second, the encryption and decryption procedures are much simpler, and consequentially, much faster. Third, the desired security level can be obtained by just setting appropriate values for the parameters of the convolutional cryptosystem. Finally, they are much more parallelizable than symmetric encryption standards based on Feistel ciphers. However, there are some weaknesses which breach the claimed security.

In this paper, a partial attack which recovers the part of plaintext without determining secret key is investigated. This attack is applicable on both versions. For simplicity, we focus attack for vesion1 and then expand to vesion2. Also, we show that there are weak keys which can obtain partially useful information from plaintext block.

The paper is organized as follows. In section 2, we provide a brief description of the convolutional codes and globally invertible convolutional transducer. And then a new partial attack and one weak key for this system is described (section 3).

II. SEQUENTIAL AND PARALLEL CASCADED CONVOLUTIONAL ENCRYPTION PROCEDURE FOR PAPER SUBMISSION

In [1,2] were proposed a class of convolutional transducers, called cascaded convolutional transducers with local propagation. An overview of the cascaded convolutional transducers is provided below.

A. Convolutional codes: a short survey

Convolutional codes are applied in applications that require good performance with low implementation cost. Convolutional codes map information to code bits sequentially by convolving a sequence of information bits with "generator" sequences. A convolutional encoder encodes K information bits to $N>K$ code bits at one time step. Encoding is very

Manuscript received February 25, 2007; revised version June 1, 2007.

Mohammad Ali Orumiehchiha is with Zaeim Electronic Ind. R&D Department, No. 21, Nilo St., Brazil St., Vanak Sq., Tehran, Iran, phone: +98-021-88773551; fax: +98-021-88776355; e-mail: orumiehchi@ {zaeim.co.ir or yahoo.com}.

S. Fahimeh Mohebbipoor is with Zaeim Electronic Ind. R&D Department, No. 21, Nilo St., Brazil St., Vanak Sq., Tehran, Iran, phone: +98-021-88773551; fax: +98-021-88776355; (e-mail: mohebbipoor @ {zaeim.co.ir }).

simple, just use the finite state. machine to produce the codewords We use the trellis diagram of the code and apply"Viterbi algorithm" Hard decision decoding or soft decision decoding could be used. Performance with soft decision decoding is better.

These designs are based on q-cascaded dynamic convolutional transducers, and are comprised in the following definition.

A key step in understanding convolutional codes is to distinguish between the convolutional encoder, the convolutional encoding operation, and the convolutional code. Rigorous definitions of all these concepts are provided below.

We denote by $B_{i\times j}$ the set of $i\times j$ arrays with binary components. If $u \in B_{i\times j}$, then the number of components of u is denoted by $|u|$, i.e., $|u| = ij$. Also, we denote by $u[q,-]$ the *q-th* row of u, and by $u[-, q]$ the q-th column of u. Note that $u[q,-] \in B_{1\times j}$ and $u[-, q] \in B_{i\times 1}$. If u is a row vector (or a column vector), then we will denote by $u[i]$ the *i-th* element of u, and by $u_{i:j}$ the subvector $[u[i] \ldots u[j]]$. If $u_1, \ldots, u_h$ are binary vectors, then we denote by $vect(u_1, \ldots, u_h)$ the vector consisting of the components of $u_1, \ldots, u_h$, in the same order. For example,

$$vect([\,0\ 0],[\,1\ 1\,]) = [\,0\ 0\ 1\ 1\,].$$

Definition2.1. Let n, k, and m be nonzero natural numbers. An (n,k,m) convolutional transducer is a function $t:\bigcup_{i=1}^{\infty} B_{1\times ki} \to \bigcup_{i=1}^{\infty} B_{1\times ni}$ given by

$$t(u) = uG_{t,|u|} \tag{1}$$

Where

$$G_{t,kp} = \begin{bmatrix} G_{t,0} & G_{t,1} & \cdots & G_{t,m} & & \\ & G_{t,0} & G_{t,1} & \cdots & G_{t,m} & \\ & & \ddots & \ddots & \cdots & \ddots \\ & & & G_{t,0} & G_{t,1} & \cdots & G_{t,m} \end{bmatrix}$$

is an element of $B_{kp\times(pn+mn)}$, $G_{t,i} \in B_{k\times n}$ for all $i \in \{0,1,\cdots,m\}$, and the arithmetic in (1) is carried out over the binary field GF(2). The entries blank are assumed to be filled in with zeros.

Definition 2.2. Let $t:\bigcup_{i=1}^{\infty} B_{1\times ki} \to \bigcup_{i=1}^{\infty} B_{1\times ni}$ be an (n,k,m) convolutional transducer. The (n, k,m) convolutional code induced by t is the image $t(\bigcup_{i=1}^{\infty} B_{1\times ki})$ of t.

Definition 2.3. Let $t:\bigcup_{i=1}^{\infty} B_{1\times ki} \to \bigcup_{i=1}^{\infty} B_{1\times ni}$ be an (n, k,m) convolutional transducer. An (n, k,m) convolutional encoder is a realization by linear sequential circuits of the semi-infinite generator matrix G_t

$$G_t = \begin{bmatrix} G_{t,0} & G_{t,1} & \cdots & G_{t,m} & & \\ & G_{t,0} & G_{t,1} & \cdots & G_{t,m} & \\ & & \ddots & \cdots & \ddots & \ddots \end{bmatrix}$$

associated with t.

For a more detailed introduction to convolutional codes, we refer the reader to [3]. Let us take an example.

Example1 Let t_1 and t_2 be (2, 2, 2) convolutional transducers with

$$G_{t_1,0} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, G_{t_1,1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, G_{t_1,2} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

and

$$G_{t_2,0} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, G_{t_2,1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, G_{t_2,2} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

The corresponding convolutional encoders are represented graphically in Fig. 1 (a) and (b), respectively. It is easy to see that only t_1 is globally invertible. For example, assume that $M_1^1 = 0, M_2^1 = 1, M_1^2 = 0, M_2^2 = 1,$ and the two output bits being decrypted are 0 and 1, respectively.

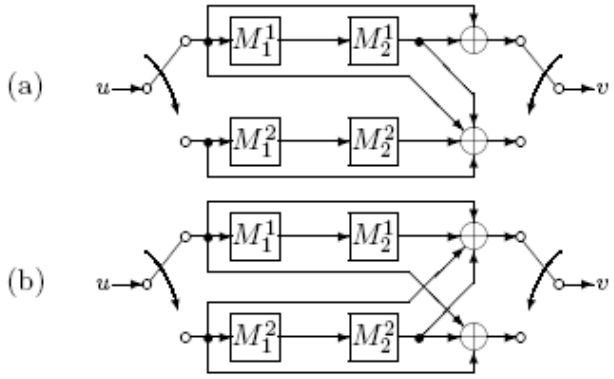

Fig.1 (a) A globally invertible (2, 2, 2) convolutional encoder, (b) A (2, 2, 2) convolutional encoder that is not globally invertible.

Given that the first output bit depends on M_2^1 and the first input bit, we conclude that the first input bit was a 1. Then, given that we already know the first input bit, we find that the second input bit was a 0. Thus, we conclude that t_1 is globally invertible, since at each step we can decode uniquely the current block of k output bits. The convolutional transducer t_2 is not globally invertible, since each of the two output bits depends on both input bits. Therefore, the current block of k output bits cannot be uniquely decoded into the corresponding input block.

Let u = [0 1 1 0] be an input vector. More precisely, we have p = 2 blocks of size k = 2 each. One can verify that $t(u) = uG_{t_1,k_p}$ = [0 1 1 1 0 1 1 1]. We can encrypt u by

v = [0 1 1 1], i.e., the first k_p = 4 bits of t(u). Given that t1 is globally invertible, we can uniquely decrypt v into u.

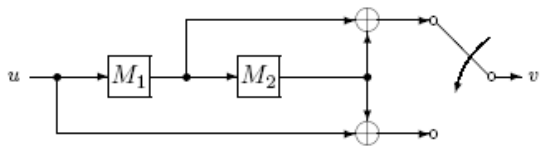

Fig.2 A (2, 1, 2) convolutional encoder.

Example2 Let $t:\bigcup_{i=1}^{\infty}B_{1\times i}\mapsto\bigcup_{i=1}^{\infty}B_{1\times 2i}$ be a (2, 1, 2) convolutional transducer, with $G_{t,0}$ =[0 1], $G_{t,1}$ = [1 0], and $G_{t,2}$ = [1 1]. Thus, the number of input bits is k = 1, the number of output bits is n = 2, and the number of memory registers is km = 2. The associated convolutional encoder can be represented graphically as in Fig.2 Note that the two output bits at each step are serialized using a multiplexer.

Let us now describe the encoding mechanism. Let b be the current input bit being encoded, and let b_1 and b_2 be the current bits stored in the memory registers M_1 and M_2, respectively. The first output bit is

$$bG_{t,0}[1]+b_1G_{t,1}[1]+b_2G_{t,2}[1]=b_1G_{t,1}[1]+b_2G_{t,2}[1]$$

Whereas the second output bit is

$$bG_{t,0}[2]+b_1G_{t,1}[2]+b_2G_{t,2}[2]=b_1G_{t,1}[2]+b_2G_{t,2}[2]$$

After both output bits have been obtained, b_1 is shifted into the memory register M_2, and b is shifted into the memory register M_1. Let us assume that the input vector u has length k_p. The actual input vector is u followed by km zeros. Thus, the total length of the output vector is pn + mn.

For example, let us take u = [0 1 0 1] as an input vector. Then, one can verify that the output vector is

$$t(u) = uG_{t,4} = [\,0\,0\,0\,1\,1\,0\,1\,0\,1\,0\,1\,1\,],$$

Where

$$G_{t,4}=\begin{bmatrix}0&1&1&0&1&1&&&&&&\\&&0&1&1&0&1&1&&&&\\&&&&0&1&1&0&1&1&&\\&&&&&&0&1&1&0&1&1\end{bmatrix}.$$

As usual, the missing entries in $G_{t,4}$ are assumed to be zeros. Note that the size of the output vector t(u) is pn + mn = 8 + 4 = 12.

Definition 2.4. A (k,k,m) convolutional cryptosystem is a globally invertible (k,k,m) convolutional transducer if each output block of k bits can be uniquely into corresponding block of k input bits.

Let us explain how we encrypt an input vector using a globally invertible (k, k,m) convolutional transducer.

Remark 1 Since we always need for encryption only the first |u| bits of t(u), we can change the standard definition of an (n, k,m) convolutional transducer by replacing equation (1) with

$$t(u)uH_{t,|u|}, \qquad (2)$$

where $H_{t,|u|}$ is the restriction of $G_{t,|u|}$ to the first |u| columns. Thus, if t is a (k, k,m) convolutional cryptosystem, then we always encrypt u by t(u).

B. Dynamic convolutional cryptosystem

In the previous subsection, we have described some symmetric encryption schemes based on globally invertible (k, k,m) (q-cascaded) convolutional transducers.The idea in [1] is based on linear q cascaded dynamic convolutional and is comprised in the following definition.

Definition 2. 5. Let *k* and *m* be nonzero natural numbers. A (k, k, m) linear q-cascaded convolutional transducer with propagation is an $(q+1)$-tuple $(t,S_1,S_2,\cdots,S_q)$, where *t* is a function given by

$$t(u)=uH^1_{t,kp}(v_0)H^2_{t,kp}(v_1)\ldots H^q_{t,kp}(v_{q-1}),$$

for all $u\in B_{1\times kp}$, where $v_0=u$, $v_i=v_{i-1}H^i_{t,kp}(v_{i-1})$ for all $i\in\{1,2,\ldots,q\}$, $H^i_{t,kp}(w)$ is the restriction of

$$G^i_{t,kp}(z)=\begin{bmatrix}G^{i,0}_{t,0,z}&G^{i,1}_{t,1,z}&\cdots&G^{i,m}_{t,m,z}&&\\&\ddots&&&\ddots&\\&&G^{i,p-1}_{t,0,z}&G^{i,p}_{t,1,z}&\cdots&G^{i,m+p-1}_{t,m,z}\end{bmatrix}$$

Description of Algorithm
Input: vector *u* as plaintext.
Output: vector *v* as ciphertext.
Private key: the sets
$S_i=\{G^i_{t,j}(0)\mid j\in\{0,1,...,m\}\}\cup\{G^i_{t,j}(1)\mid j\in\{0,1,...,m\}\}$
, $i\in\{1,...,q\}$.

1. set $p=\dfrac{lenght\ pla\,\text{int}\,ext\ u}{k}$,
2. $v_0=u$,
3. For *i=1* to *q* do
   3.1. $H^i_{t,kp}(v_{i-1})$ is Constructed by the restriction of matrix (3) to the first kp columns,
   3.2. $v_i=v_{i-1}H^i_{t,kp}(v_{i-1})$,
   end do;
4. $v=v_q$.

Fig. 3Description of algorithm

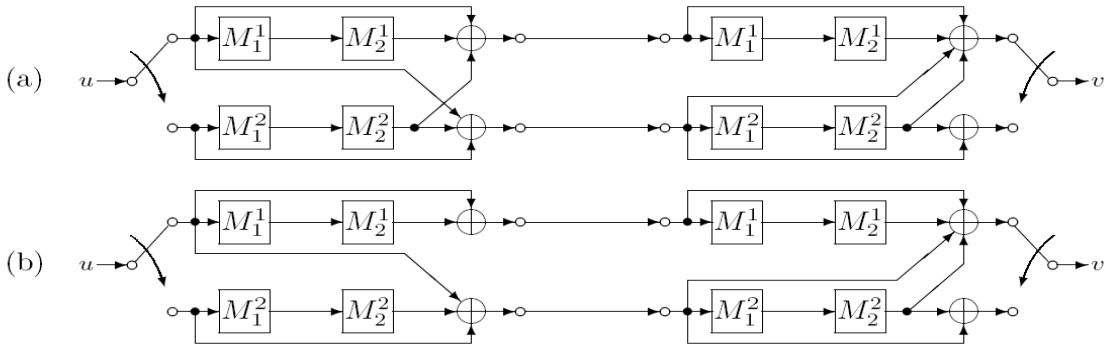

Fig. 4 A (2, 2, 2) 2-cascaded convolutional encoder with propagation: (a) the initial structure of the cascaded encoder, and (b) the structure after encoding the first block [1 0].

to the first kp columns, $z = \text{vect}(w, [0 \dots 0])$, $G^{i,0}_{t,j,z} = G^i_{t,j}(0)$, $G^{i,r}_{t,j,z} = G^i_{t,j}(f(r-1,z))$ for all $r \in \{1, 2, \dots, m+p-1\}$, $f(s,z) = (z_{sk+1:(s+1)k}[1] + \dots + z_{sk+1:(s+1)k}[k]) \bmod 2$, and $S_i = \{G^i_{t,j}(0) \mid j \in \{0,1,...,m\}\} \cup \{G^i_{t,j}(1) \mid j \in \{0,1,...,m\}\}$, is the set of state matrices corresponding to the *i*-th transducer of the cascade, $i = 1, \cdots, q$. As usual, all the operations are performed over the binary field GF(2). And also $G^i_{t,j}(0), G^i_{t,j}(1) \in B_{k\times k}$ and are invertible. The entries left blank in $G^i_{t,kp}(z)$ are assumed to be filled in with zeros.

Definition 2.6. A (k,k,m) linear q-cascaded convolutional cryptosystem with propagation is a globally invertible (k,k,m) q-cascaded convolutional transducer with propagation with encryption function *t* in which the sets $S_1, \cdots, S_q$ are kept private. The main loop of this cryptosystem described in Fig.3.

Example3 Let $t : \bigcup_{i=1}^{\infty} B_{1\times ki} \mapsto \bigcup_{i=1}^{\infty} B_{1\times ni}$ be a (2, 2, 2) 2-cascaded convolutional transducer, where

$$G^{1}_{t,0}(0) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad G^{1}_{t,0}(1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$G^{1}_{t,1}(0) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad G^{1}_{t,1}(1) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$G^{1}_{t,2}(0) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \qquad G^{1}_{t,2}(1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$G^{2}_{t,0}(0) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad G^{2}_{t,0}(1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$G^{2}_{t,1}(0) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \qquad G^{2}_{t,1}(1) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$G^{2}_{t,2}(0) = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad G^{2}_{t,2}(1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Let $u = [1001]$ be an input vector, then $p = \frac{4}{2} = 2, v_0 = u$ and

$$H^1_{t,4} = G^1_{t,4}(z) = \begin{bmatrix} G^{1,0}_{t,0,z} & G^{1,1}_{t,1,z} \\ \overline{0} & G^{1,1}_{t,0,z} \end{bmatrix}_{4\times 4},$$

where $z = [10010000]$, therefore

$G^{1,0}_{t,0,z} = G^1_{t,0}(0)$,

$G^{1,1}_{t,1,z} = G^1_{t,1}(f(0,z)) = G^1_{t,1}(1)$, $(f(0,z) = 1+0 = 1)$

$G^{1,1}_{t,0,z} = G^1_{t,0}(f(0,z)) = G^1_{t,0}(1)$, and

$$H^1_{t,4} = G^1_{t,4}(z) = \begin{bmatrix} G^1_{t,0}(0) & G^1_{t,1}(1) \\ \overline{0} & G^1_{t,0}(1) \end{bmatrix}_{4\times 4} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$v_1 = uH^1_{t,4}(v_0) = [1101]$, and

$$H^2_{t,4} = G^2_{t,4}(z) = \begin{bmatrix} G^{2,0}_{t,0,z} & G^{2,1}_{t,1,z} \\ \overline{0} & G^{2,1}_{t,0,z} \end{bmatrix}_{4\times 4}$$

Where $z = [11010000]$,

$G^{2,0}_{t,0,z} = G^2_{t,0}(0)$,

$G^{2,1}_{t,1,z} = G^2_{t,1}(f(0,z)) = G^2_{t,1}(0)$, $(f(0,z) = 1+1 = 0)$

$G^{2,1}_{t,0,z} = G^2_{t,0}(f(0,z)) = G^2_{t,0}(0)$,

and

$$H^2_{t,4} = G^2_{t,4}(z) = \begin{bmatrix} G^2_{t,0}(0) & G^2_{t,1}(0) \\ \overline{0} & G^2_{t,0}(0) \end{bmatrix}_{4\times 4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix},$$

Thus encrypt u by $t(u) = uH^1_{t,4}(v_0)H^2_{t,4}(v_1) = [01\,1\,1]$. The initial structure of the corresponding cascaded convolutional encoder can be represented graphicallyas in Fig. 4 (a). By examining the matrices $G^i_{t,j}(b)$,one can remark that the cascaded encoder is globally invertible whatever its structure would be at the current step. More precisely, the structure of the first encoder in the cascade changes after

encoding the first input block, since the number of 1's in the first input block is odd. Regarding the second encoder in the cascade, its structure changes after encoding the first input block, since the number of 1's in the first input block is even. Thus, we encrypt u by t(u), and given that t is globally invertible, we can uniquely decrypt t(u)into u.

And also In [2] proposed a new class of dynamic convolutional cryptosystems, called automata-based dynamic convolutional cryptosystems.

Definition 2.7. Let k,m be nonzero natural numbers. A (k,k,m) automata-based dynamic convolutional transducer (abbreviated *ADCT*) with q states is a *(q + 2)*-tuple *(t, f, S1, . . . , Sq)*, where

$$t(u) = uH_{t,kp}(u)$$

for all $u \in B_{1\times kp}$, $f: \{1, 2, \ldots, q\} \times B_{1\times k} \rightarrow \{1, 2, \ldots, q\}$ is the transition function, $H_{t,kp}(u)$ is the restriction of

$$G_{t,kp}(z)=\begin{bmatrix} G^0_{t,0,z} & G^1_{t,1,z} & \cdots & G^m_{t,m,z} & \\ & \ddots & \ddots & \cdots & \ddots \\ & & G^{p-1}_{t,0,z} & G^p_{t,1,z} & \cdots & G^{m+p-1}_{t,m,z} \end{bmatrix}$$

to the first *kp* columns, $z = vect(u, [\underbrace{0\cdots 0}_{mk}])$,

$$G^0_{t,0z} = G^1_{t,j}, G^i_{t,j,z} = G^{g(i,z)}_{t,j},$$

$$g(i,z)=\begin{cases} f(1, z_{1,k}) & if\ i=1 \\ f(g(i-1), z), z_{(i-1)k+1} & if\ i \in \{2,...,m+p-1\} \end{cases}$$

is the tuple of matrices corresponding to the i-th state.

Definition 2.8. A (k,k,m) automata-based dynamic convolutional cryptosystem (abbreviated *ADCC*)with q states is a globally invertible (k, k,m) *ADCT* with q states in which the transition function if and the tuples $S_1, \ldots, S_q$ are kept private.

Remark 2 Since the transition function of an *ADCC* requires $q2^k$ memory locations, we should define it in such a way that the amount of memory required is significantly reduced. For a (256, 256, 32) *ADCC* with ten states, the transition function requires $10 \cdot 2^{256}$ memory locations,which is unacceptable. One possibility would be to transform f into a periodic function. For a (k, k,m) *ADCC*, this can be done by defining *f* by

$f(i, u_1) = f(i, u_2)$ if and only if $int(u_1) \bmod P = int(u_2) \bmod P$, where

$$int(u) = 2^{k-1}u[1] + \ldots + 2^0 u[k]$$

and $P \in \{2^0, 2^1, ..., 2^k\}$. It can be easily seen that the period is *P*. Thus, the number of memory locations required by the transition function is reduced to *qP*.

Cascaded convolutional transducers with propagation are highly parallelizable. More precisely, in [1] described a parallel algorithm for cascaded convolutional cryptosystems with propagation using the shared-memory model of computation under the asynchronous mode of operation.

Let t be a globally invertible (k, k,m) q-cascaded convolutional transducer with propagation.

Suppose that we have k processors, denoted $p_1, p_2, \ldots, p_k$, and consider a binary vector $u \in B_{1\times kp}$.

The processors work as follows. First, each processor copies (concurrently) the input vector u and the matrices $G^i_{t,j}(b)$ from the global memory into its local memory. Second, for all $l \in \{1, 2, \ldots, q\}$, each processor p_j computes the columns

$$H^l_{t,kp}(u)[-j], \cdots, H^l_{t,kp}(u)[-, j + (p-1)k]$$

Input: Two binary row vectors u, v of size kp each, and the set of matrices $\{G^i_{t,j}(b) \mid i \in \{1, \ldots, q\}, j \in \{0, \ldots, m\}, b \in \{0, 1\}\}$ stored in the global memory; also, consider a shared variable *Var* initially set to 0.
Output: The product $u\boldsymbol{H}^1_{t,kp}(v_0)\boldsymbol{H}^2_{t,kp}(v_1)\ldots\boldsymbol{H}^q_{t,kp}(v_{q-1})$ stored in the shared location v.

```
1:  X_j ← u   {concurrent READ}
2:  For each i_1 ∈ {1,2,...,q} do
3:    For each i_2 ∈ {0,1,...,m} do
4:      For each i_3 ∈ {0,1} do
5:        Local^j_{i_1,i_2,i_3} ← G^{i_1}_{t,i_2}(i_3)   {concurrent READ}
6:  For l ← 1 to q do
7:    For i ← 1 to p do
8:      Compute the (j + (i − 1)k)-th column of H^l_{t,kp}(X_j)
9:      using the matrices Local^j_{i_1,i_2,i_3} and store it in the local
10:     variable Y_j.
11:     Z_j ← X_j Y_j
12:     v[j + (i − 1)k] ← Z_j[1]
13:   Endfor
14:   Var ← Var + 1   {concurrent WRITE}
15:   If Var = k then   {concurrent READ}
16:     Var ← 0
17:   Else
18:     WAIT until Var = 0   {concurrent READ}
19:   Endif
20:   X_j ← v   {concurrent READ}
21: Endfor
```

Fig. 5Asynchronous parallel cascaded convolutional encryption with propagation Asynchronous parallel cascaded convolutional encryption with propagation.

Then computes the products

$$uH^l_{t,kp}(u)[-j], \cdots, uH^l_{t,kp}(u)[-, j + (p-1)k]$$

and finally stores the corresponding output bits into the shared location v. The shared variable *Var* ensures that, at any time, each processor either works on the current vector-matrix product or stays idle. In other words, we do not allow the processors to work on different vector-matrix products. In line 20, each processor p_j copies (concurrently) the current output vector v into its local memory, since at the next iteration v becomes the new input vector. (starting at line 6) is finished, the product lies in the shared location v.

A complete description of the parallel encryption algorithm (for processor p_j) is provided in Fig.5.

The parallel decryption algorithm is esentially the same sequence of operations, but in reverse order. When the main loop Lines 1,6,14,15,18, and 20 are executed concurrently, whereas the other lines are executed independently by each processor. Let us denote by W1 the maximum amount of time

spent by each processor to read (concurrently) the vector u (line 1, and also line 20 for v), by W2 the amount of time spent by each processor to read the matrices $G_{t,j}^{i}(b)$ from the global memory (lines 2,3,4,5),by W3 the maximum amount of time spent by each processor to execute (concurrently) the lines 14–19 at the current iteration, and by T_{seq} the runtime of the sequential algorithm. Then, the parallel running time is at most

$$W1 + W2 + q(T_{seq}/qk + W3 + W1),$$

since each processor spends at most W1 time to execute line 1, at most W2 time to execute the lines 2–5, and at each of the q iterations starting at line 6, each processor spends exactly T_{seq}/qk time to execute the lines 7–13, at most W3 time to execute the lines 14 through 19, and at most W1 time to execute line 20.

This encryption technique has at least four advantages over traditional schemes based on Feistel ciphers. First, the secret key of a convolutional cryptosystem is usually much easier to generate (just generate the matrices $G_{t,j}^{i}(b)$ such that the cascaded encoder is globally invertible whatever its structure would be at the current step). Second, the encryption and decryption procedures are much simpler, and consequentially, much faster. Third, the desired security level can be obtained by just setting appropriate values

for the parameters of the convolutional cryptosystem. Finally, they are much more parallelizable than symmetric encryption standards based on Feistel ciphers.

C. Claimed Security and performance

The security level can be obtained by just setting appropriate values for 2q(m + 1) + 3 parameters: k, m, q and 2q(m + 1) elements of sets $S_i (i = 1,...,q)$. As claimed in [1,2], the highest level of security is obtained when all the parameters are kept secret, since this increases the complexity of any cryptanalytic attack.

If the input vector has length p_k, and if $d(p_k)$ denotes the number of divisors of p_k that are less than or equal to k, then the maximum number of decoding attempts is

$$\sum_{i=1}^{d(pk)} \sum_{j=1}^{m} \sum_{l=1}^{q} [2^{i^2 2l(j+1)+1}],$$

since the maximum number of trials for k is $d(p_k)$ and we have 2q(m+ 1) binary matrices of size k^2 each. Note that for each of the $2^{i^2 2l(j+1)}$ maximum trials for fixed values of i, j, and l, there are two decoding attempts: one attempt for the case in which the number of 1's in the first input block is even, and another attempt for the case in which the number of 1's in the first input block is odd. In fact, in [1,2] the measure of safety considered is huge secret key entropy. We show that this measure is only a necessary condition not a sufficient condition.

Regarding the performance, in [1] have made several comparisons between a globally invertible (16, 16, 1) 2-cascaded convolutional transducer t with propagation and the well-known AES Encryption Benchmark. It is shown that the proposed cipher is much faster than a certain AES implementation. Also, due to being more complex, it is considered that this dynamic cryptosystem is secure against standard cryptanalytic attacks such as linear and differential cryptanalysis. Although these standard attacks may be infeasible, but there are attacks which are able breach the proposed security.

III. Cryptanalysis of scheme

The idea design of cipher is attractive but unfortunately cipher is not as safe as is claimed. The claimed security is comparable to a homogeneous strong block cipher.

A dynamic convolutional cryptosystem is considerd

Broken as soon as attacker finds the bits of the matrices $G_{t,0}^{1},...,G_{t,m}^{1},...,G_{t,0}^{q},...,G_{t,m}^{q}$. Breaking the cipher completely is a very diffcult task, However, finding partial information about the cipher is easy, the idea of such an attack described in detail in following.

In [1,2] is claimed that security of design is based privacy of the sets $S_1,\cdots,S_q$. But in this section, it is shown that one can easily obtain the first sub-block plaintext without any information from the sets $S_1,\cdots,S_q$. Also, if these sets are chosen improper then one recovers more sub-block plaintexts.

A. Partial attack

In this scheme, the first output sub-block ciphertext is constructed only by multiplying the first input sub-block plaintext (denote u^1) and matrix

$$A^1 = G_{t,0}^{1}(0)G_{t,0}^{2}(0)\cdots G_{t,0}^{q}(0).$$

Therefore, with changing the other input sub-blocks and fixing the first input sub-block, the first cipher sub-block remains fixed. And also, the contents of matrixes $G_{t,0}^{1}(0), G_{t,0}^{2}(0),\cdots, G_{t,0}^{q}(0)$ are always fixed, and hence A^1 is fixed for all of arbitrary plaintext. The number of contents of matrix A^1 is K^2, hence one can determine all of contents A^1. By using about K^2 output bits belong to the first output sub-blocks and write a linear system and solve it. Therefore, for each ciphertext one can calculate the first sub-block of plaintext without recovering master key.

Additionally, if plaintext is chosen such that the first sub-block be equal to zero and other sub-blocks of plaintext be random then always the first sub-block will be zero and is distinguishable with probability 1 from a random sequence. we can described similar to this attack for a (k,k,m) automation-based dynamic convlutional cryptosystem.

Example4 Let $t : \bigcup_{i=1}^{\infty} B_{1\times ki} \mapsto \bigcup_{i=1}^{\infty} B_{1\times ni}$ be a (2, 2, 2) 2-cascaded convolutional transducer, where

$$G^{1}_{t,0}(0)=\begin{bmatrix}1&1\\0&1\end{bmatrix}\qquad G^{1}_{t,0}(1)=\begin{bmatrix}1&1\\0&1\end{bmatrix}$$
$$G^{1}_{t,1}(0)=\begin{bmatrix}0&0\\0&0\end{bmatrix}\qquad G^{1}_{t,1}(1)=\begin{bmatrix}0&0\\0&0\end{bmatrix}$$
$$G^{1}_{t,2}(0)=\begin{bmatrix}1&1\\0&1\end{bmatrix}\qquad G^{1}_{t,2}(1)=\begin{bmatrix}1&0\\0&1\end{bmatrix}$$
$$G^{2}_{t,0}(0)=\begin{bmatrix}1&0\\1&1\end{bmatrix}\qquad G^{2}_{t,0}(1)=\begin{bmatrix}1&1\\0&1\end{bmatrix}$$
$$G^{2}_{t,1}(0)=\begin{bmatrix}0&0\\0&0\end{bmatrix}\qquad G^{2}_{t,1}(1)=\begin{bmatrix}0&0\\0&0\end{bmatrix}$$
$$G^{2}_{t,2}(0)=\begin{bmatrix}1&0\\1&1\end{bmatrix}\qquad G^{2}_{t,2}(1)=\begin{bmatrix}1&1\\0&1\end{bmatrix}$$

In this example, the matrices

$$A^1=G^1_{t,0}(0)G^2_{t,0}(0)=\begin{bmatrix}1&1\\0&1\end{bmatrix}\begin{bmatrix}1&0\\1&1\end{bmatrix}=\begin{bmatrix}0&1\\1&1\end{bmatrix}$$

the contents of matrixes $A^1=G^1_{t,0}(0),G^2_{t,0}(0)$, are always fixed, and hence A^1 is fixed for all of arbitrary plaintext. The number of contents of matrix A^1 is 4, hence one can determine all of contents A^1. By using about $K^2=2^2$ output bits belong to the first output sub-blocks and write a linear system and solve it. Therefore, for each ciphertext one can calculate the first sub-block of plaintext without recovering master key e.i for any plaintext the first of sub block of cipher text obtain of multiply firstth sub block plaintext and matrix A^1.

Example5 Let a (16,16,1) 2- cascaded convolutional transducer that proposed in [1]. For an input vector of length 64, can determine the first 16 bits of input block with 2^8 output bits belong to the first output sub-blocks and write a linear system and solve it. Additionally, if plaintext is chosen such that the first sub-block be equal to zero and other sub-blocks of plaintext be random then always the first sub-block will be zero and is distinguishable with probability 1 from a random sequence.

B. Weak key

In this system, If matrices $G^1_{t,0}(0),G^2_{t,0}(0),\cdots,G^q_{t,0}(0)$ in the sets $S_1,\cdots,S_q$ be equal, means that $A^1=(G^1_{t,0}(0))^q$, and so we can obtain matrix A^l and $G^1_{t,0}$. Regarding to the hamming weight of $u^1,v^1_0,v^1_1,\cdots,v^1_q$ are either even or odd, we can parse all possible first sub-blocks u^1 to 2^q classes. Let which for this matrix only second column is unknown. Hence, by using about $2\times K^2$ output bits belong to the second output sub-block, attacker can write a linear system and solve it, so A^2 can be determined. Therefore with determining u^1, we can deduce the second sub-block.

If this weak point continues for system, attacker can decrypt longer fragments of ciphertexts. This attack is described in the following example.

Example6 Let $t:\bigcup_{i=1}^{\infty}B_{1\times ki}\mapsto\bigcup_{i=1}^{\infty}B_{1\times ni}$ be a (2, 2, 2) 2-cascaded convolutional transducer,

$$G^{1}_{t,0}(0)=\begin{bmatrix}1&1\\0&1\end{bmatrix}\qquad G^{1}_{t,0}(1)=\begin{bmatrix}1&1\\0&1\end{bmatrix}$$
$$G^{1}_{t,1}(0)=\begin{bmatrix}0&0\\0&0\end{bmatrix}\qquad G^{1}_{t,1}(1)=\begin{bmatrix}0&0\\0&0\end{bmatrix}$$
$$G^{1}_{t,2}(0)=\begin{bmatrix}1&1\\0&1\end{bmatrix}\qquad G^{1}_{t,2}(1)=\begin{bmatrix}1&0\\0&1\end{bmatrix}$$
$$G^{2}_{t,0}(0)=\begin{bmatrix}1&1\\0&1\end{bmatrix}\qquad G^{2}_{t,0}(1)=\begin{bmatrix}1&1\\0&1\end{bmatrix}$$
$$G^{2}_{t,1}(0)=\begin{bmatrix}0&0\\0&0\end{bmatrix}\qquad G^{2}_{t,1}(1)=\begin{bmatrix}0&0\\0&0\end{bmatrix}$$
$$G^{2}_{t,2}(0)=\begin{bmatrix}1&0\\1&1\end{bmatrix}\qquad G^{2}_{t,2}(1)=\begin{bmatrix}1&1\\0&1\end{bmatrix}$$

In this example, the matrices $G^1_{t,0}(0)=G^2_{t,0}(0)$ are identical. Hence, $A^1=(G^1_{t,0}(0))^2$ and so we can obtain matrix A^1 and $G^1_{t,0}(0)$. Regarding that the hamming weight of u^1,v^1_0, are either even or odd, we can parse all possible first sub-block u^1 to 2^2 classes.

Let $A^2=\begin{bmatrix}G^1_{t,0}(0)&G^{1,1}_{t,1,z}\\0&G^{1,1}_{t,0,z}\end{bmatrix}\begin{bmatrix}G^1_{t,0}(0)&G^{2,1}_{t,1,z}\\0&G^{2,1}_{t,0,z}\end{bmatrix}$ which for this matrix only second column is unknown. Hence, by using about $2\times K^2$ output bits belong to the second output sub-block, attacker can write a linear system and solve it, so can determine A^2 and then the second sub-block plaintexts are recovered too.

Example7 Let a (16,16,1) 2- cascaded convolutional transducer that proposed in [1]. For an input vector of length 64, if the key be weak, as mentioned before, then for any output block, we can calculate the first 32 bits of input block and also if the key is not week, then can determine the first 16 bits of input block.

IV. CONCLUSION

In this paper, we investigated two attacks on the proposed design. The first attack is a partial key recovery which for a (k,k,m) q-cascade was determined a fragment of ciphertext without calculating master key with complexity $O(k^2)$. In addition, a weak key on this system was found that one could recover longer fragment of plaintexts with complexity $O(2\times k^2)$.

REFERENCES

[1] D. Trinca," Sequential and Parallel Cascaded Convolutional Encryption with Local Propagation: Toward Future Directions in Symmetric Cryptography," *the 3rd International Conference on Information Technology: USA*, 2006, pp. 464–469, *IEEE Computer Society Press*.

[2] D. Trinca, "Efficient FPGA Implementations and Cryptanalysis of Automata-based Dynamic Convolutional Cryptosystems," Available: http://www.eprint.iacr.org/2006/263.

[3] A. Dholakia.: Introduction to Convolutional Codes with Applications. Kluwer (1994).

[4] Ph. Piret : Convolutional Codes: An Algebraic Approach. MIT Press, Cambridge, MA, USA(1988).