Novel Stream Cipher Using SCAN and Variable Ordered Recursive CA Substitutions and Its DSP+FPGA Implementation

Rong-Jian Chen\textsuperscript{a\ast}, Jui-Lin Lai\textsuperscript{a}, and Shi-Jinn Horng\textsuperscript{a, b}

\textsuperscript{a} Department of Electronic Engineering, National United University, Taiwan
\textit{rjchen@nuu.edu.tw, jllai@nuu.edu.tw}

\textsuperscript{b} Department of Computer Science and Information Engineering, National Taiwan Univ. of Sci. and Tech., Taiwan
\textit{horngsj@yahoo.com.tw}

Abstract—This paper presents a new stream cipher for data security, which is based on permutation of the data and replacement of the data values. Permutation is done by scan patterns generated by the SCAN approach. The replacement of data values using variable ordered recursive cellular automata (CA) substitutions. To achieve this aim, an encryption-specific SCAN technique was firstly developed, 2-D hybrid CA was next built, and then 1\textsuperscript{st}-ordered and 2\textsuperscript{nd}-ordered generalized CA transforms were introduced to build variable ordered recursive CA substitutions. The proposed stream cipher satisfies the properties of confusion and diffusion because of characteristics of the SCAN and the CA substitutions are flexible. Moreover, the characteristics of the proposed stream cipher are loss-less, symmetric private key, very large number of security keys (number of possible security keys is more than $10^{3768} \sim 10^{4785}$ - according to the size of the 2-D von Neumann CA), and key-dependent pixel value replacement. Experimental results obtained using some color images clearly demonstrate the strong performance of the proposed stream cipher. This paper also shows the DSP+FPGA implementation of the proposed stream cipher for the real-time image security.

Index Terms—Stream cipher, SCAN, Variable ordered recursive CA substitutions, DSP+FPGA implementation

I. INTRODUCTION

With the ever-increasing growth of multimedia applications, data security is an important issue in communication and storage of data, and encryption is one the ways to ensure security. Data security has applications in inter-net communication, multimedia systems, medical imaging, telemedicine, and military communication. There already exist several methods of data security. They include SCAN-based methods [1-5], chaos-based methods [6-8], tree structure-based methods [9-11], and other miscellaneous methods [12-15]. However, each of them has its strength and weakness in terms of security level, speed, and resulting stream size metrics. We therefore proposed a new method of data security to overcome these problems. The proposed data security belongs to stream cipher which encryption method is based on permutation of the data and replacement of the data values. Permutation is done by scan patterns generated by the SCAN approach, the SCAN approach described in [5] is used because it produces a high volume of scan pattern. The data values are replaced using the variable ordered recursive CA substitution with a sequence of CA data that is generated from 2-D hybrid CA with special evolution rules. The advantages of CA in the proposed data security are described as follows. (1) CA has been applied successfully to several physical systems, processes, and scientific problems that involve local interactions as in image processing [17], data encryption [18, 19], byte error correcting code [20]; it has also been used in pseudorandom number generators for VLSI built-in self-test [21]. (2) Number of CA evolution rules is very large. Hence, many techniques are available for producing a sequence of CA data that is generated from 2-D hybrid CA with special evolution rules. The advantages of CA in the proposed data security are described as follows. (3) Recursive CA substitution only requires integer arithmetic and/or logic operations simplifying the computation.

The proposed data security also belongs to the case of the general framework called iterated product cipher [4, 22, 23], which is based on repeated and intertwined application of permutation and substitution. This...
general framework has been extensively studied and developed in terms of cryptographic strengths and attacks [4] and forms the basis of many modern encryption methods, including Data Encryption Standard [23], Advanced Encryption Standard [24], and chaos-based methods [6-8]. The proposed image security method is differs markedly from that used elsewhere [18]. In this work, hybrid 2-D von Neumann CA was used to generate a high-quality random sequence as key-stream, with recursive CA substitution in the encryption and decryption schemes, such that the proposed image security system was secure. The cipher systems in the cited study [18] are affine and based on 1-D CA, and the encryption and the decryption schemes in [18] are non-recursive. Another study [25] showed that affine cipher systems are insecure. The proposed data security is an extended and improved version of that presented in cited study [26, 36] because the proposed one used SCAN techniques and 20 groups of CA recursive substitution to make the exhaustive searching attack much harder and to enhance the performances of system. This new data security proposed herein has additional features, such as key-dependent permutation, key-dependent pixel value replacement, very large key space, keys of variable length, and encryption of larger blocks. Moreover, it is a symmetric private key security system, meaning that the same keys are needed for encryption and decryption. Therefore, both sender and receiver must know the keys.

After well development of the proposed stream cipher, we then developed SCAN-CA-based image security system to illustrate the performance of the proposed stream cipher. Moreover, we built hardware/software implementation of the SCAN-CA-based image security system to satisfy the real-time requirement due to real-time image processing systems are widely used in many fields, such as industry, military, medical image processing and so on. For real-time image processing system, it needs high speed because of mass image data, so we can use DSP to solve this problem. On the other hand, FPGA has capable of flexible logic control, large memory and fast executing speed. So the real-time image processing system can be constituted by the combination of DSP and FPGA. In order to satisfy the real-time demand, we established hardware/software implementation of the SCAN-CA-based image security system within an embedded platform that includes a DSP and FPGA where DSP is SCAN processing unit and FPGA is the processing unit of variable ordered recursive CA substitutions.

This paper is organized as follows. Section II describes SCAN approach and 2-D hybrid CA, Section III illustrates SCAN-CA-based image security system using the proposed stream cipher. Section IV presents the possible keys and the cryptanalysis. Software simulation results of the SCAN-CA-based image security system are drawn in Section V; moreover, we also show the DSP+FPGA implementation of the proposed SCAN-CA-based image security system in Section V. Finally, section VI gives the discussions and conclusions.

II. SCAN AND 2-D HYBRID CA

A. SCAN Approach

Scanning of a 2-D array is an order where each element of the array is accessed exactly once. In other words, scanning of a 2-D array is a permutation of the array elements. Thus, scanning of a 2-D array $F_{M×N} = \{f(i,j) : 0 ≤ i ≤ M − 1, 0 ≤ j ≤ N − 1\}$ is a mapping function from $F_{M×N}$ to the set $\{g(l) : 0 ≤ l ≤ (M×N − 1)\}$. Note that an $M×N$ array has $(M×N)!$ scanning paths.

![Figure 1. Basic scan patterns](Image1)

![Figure 2. Transformations with partition](Image2)

The SCAN represents a family of formal language-based 2-D spatial accessing approaches, which can represent and generate a large number of scanning paths systematically. There are several versions of SCAN language, such as Simple SCAN, Extended SCAN, and Generalized SCAN; each of them can represent and generate a specific set of scanning paths. Each SCAN language is defined by a grammar. Each language has a set of basic scan patterns, a set of transformations of scan patterns, and a set of rules to...
recursively compose simple scan patterns to obtain complex scan patterns. Figure 1 shows eight basic scan patterns, which are used in computer simulation and can be extended or reduced according to the need of a particular application. Transformations include 6 basic operations (identity, horizontal reflection, vertical reflection, rotation by 90°, 180°, and 270°) and their combinations. The rules for building complex scan patterns from simple scan patterns are specified by the production rules of the grammar of each specific language. Readers are referred to [1-5] for a detailed description of syntax and semantics of SCAN languages and their applications.

In the proposed encryption method, the scanning patterns are used as the encryption keys to rearrange pixels of the image. The scanning patterns are generated by an encryption-specific SCAN language [16] which is formally defined by the grammar $H = \{ \Gamma, \Sigma, \Pi \}$ where $\Gamma = \{ A, S, P, U, V, T \}$ are non-terminal symbols, $\Sigma = \{ r, c, d, l, a, i, t, w, B, Z, X, , , \}$, space, 0, 1, 2, 3, 4, 5, 6, 7\} are terminal symbols, $A$ is the start symbol, and production rules $\Pi$ are given by $A \rightarrow S\{P\}$, $S \rightarrow UT$, $P \rightarrow VT(A A A A)$, $U \rightarrow r|c|d|l|a|i|t|w$, $V \rightarrow B|Z|X$, and $T \rightarrow 0|1|2|3|4|5|6|7$. The semantics of the encryption-specific SCAN language is described as follows. $A \rightarrow S\{P\}$: Process the region by scan $S$ or partition $P$. $S \rightarrow UT$: Scan the region with scan pattern $U$ and transformation $T$. Each of scan patterns has eight transformations (Figure 2) which are defined as: 0: Identity. 1: Horizontal reflection. 2: 90° clockwise rotation. 3: 90° clockwise rotation followed by vertical reflection. 4: 180° clockwise rotation. 5: Vertical reflection. 6: 270° clockwise rotation. 7: 90° clockwise rotation followed by horizontal reflection. $P \rightarrow VT(A A A A)$: Partition the region with partition pattern $V$ and transformation $T$, and process each of the four sub-regions in partition order using $A$s from left to right. Figure 2 shows that partition patterns were divided into 3 groups: $B$, $Z$, and $X$, each group has eight transformations as that of scan patterns. $U \rightarrow r|c|d|l|a|i|t|w$: Scan the region with specific scan pattern $U$ as in Figure 1, the letters $r$, $c$, $d$, $l$, $a$, $i$, $t$, and $w$ in Fig. 1 indicate the type of scan patterns. $V \rightarrow B|Z|X$: Partition with a specific partition pattern $B$, $Z$, or $X$ as in Figure 2. $T \rightarrow 0|1|2|3|4|5|6|7$: Use one of the eight transformations for a scan or a partition.

Figure 3 shows the scanning path of the scan pattern $X3(c2Z2(d0w2a4l1)r0i4)$ for a 16×16 image. The image is first partitioned into four sub-regions using $X3$ partition order. These four sub-regions are scanned using $c2$, $Z2(d0w2a4l1)$, $r0$, and $i4$. The second sub-region is further partitioned into four sub-regions using $Z2$ order and these four sub-regions are scanned using $d0$, $w2$, $a4$, and $l1$.

**B. 2-D Hybrid CA**

Cellular automata are dynamic systems in which space and time are discrete. The cells, as arranged in a regular lattice structure, have a finite number of states. These states are updated synchronously according to a specified local rule of neighborhood interaction. The neighborhood of a cell refers to the cell and some or all of its immediately adjacent cells. In 2-D CA space, the specified node $P$, with its four nearest neighbors form the von Neumann neighborhood. Figure 4a shows the 2-D von Neumann CA space. The state of the given node at time step $(t+1)$ will be determined from the states of nodes within its neighborhood at time step $t$. Using a specified rule, the states are updated synchronously in time steps for all cells. Let $a(i, j)$, represent the state of $(i,j)$th cell at time $t$, whose von Neumann neighborhoods are in the states: $a(i−1, j)$, $a(i, j−1)$, $a(i+1, j)$, and $a(i, j+1)$. Then the rule of 2-D von Neumann CA evolution way can be expressed as

$$a(i, j)_{t+1} = F(a(i+1, j), a(i, j−1),$$

$$a(i, j), a(i, j+1), a(i−1, j))$$

where $F$ is a Boolean function that defines the rule. Readers are referred to [17-21] for a detailed description of 2-D von Neumann CA.

The hardware implementation of Eq. (1) for 1-bit 2-D von Neumann CA is shown in Figure 4b. Such a structure is referred as a programmable additive CA (PACA) due to it is implemented using EXOR gates. Using the 1-bit 2-D von Neumann PACA structure, one can build the desired hybrid $N \times N$-bit cellular automata [27-29]. It costs $6N^2 + N^2 + 4N$ bits to assign boundary condition, rule control, and initial data.
to indicate the 2-D hybrid $N \times N$ von Neumann CA generator to produce CA data sequence. Given a 2-D $N \times N$-cell dual-state von Neumann CA runs over $T$ time steps, it has $2^6$ rules, $2^{5N^2-3}$ initial configurations, $2^{4N}$ boundary conditions, and results in $2^{6N^2-2N^2-4N}$ CA evolution ways for generating $T \times N$ N-bit generalized CA data. Consequently, cyclic boundary conditions were imposed on a 2-state/3-site/ $N \times N$-cells CA to generate the states of the automata.

Figure 4. 2-D von Neumann CA, (a) 2-D von Neumann CA space, (b) the structure of 1-bit PACA

III. SCAN-CA-BASED IMAGE SECURITY SYSTEM USING THE PROPOSED STREAM CIPHER

Let $F(i), 0 \leq i \leq L_i - 1$ be a sequence of $N$-bit input data, $E(i), 0 \leq i \leq L_i - 1$ be a sequence of $N$-bit output encrypted data, $D(i), 0 \leq i \leq L_i - 1$ be a sequence of $N$-bit output encrypted data, and $CA_{\mu}(i), 0 \leq i \leq L_i - 1$ be a sequence of $N$-bit CA data. We firstly defined variable ordered generalized CA transforms (GCATs) as TABLE I. There are 4 groups of 1st-ordered GCATs and 16 groups of 2nd-ordered GCATs which were labeled as GCAT1 and GCAT2 in TABLE 1, respectively. Among GCAT1 and GCAT2, $0 \leq LS_1, LS_2 \leq 2^N - 1$ are the values of level shift. Since we have 20 groups of GCAT1 and GCAT2, each group of GCAT1 consists of $2^N$ 1st-ordered GCATs and each group of GCAT2 consists of $2^{2N}$ 2nd-ordered GCATs, therefore $(5+2N)$-bit Type Selection data were used to specify the type of GCAT.

Based on the definition of variable ordered GCATs, we firstly develop the architecture of the variable ordered recursive CA encrypted/decrypted substitution which is shown in Figure 5. In Figure 5, when encryption/decryption control bit is set to one and zero, it executes the recursive CA encryption and decryption substitution respectively. For example, the encryption/decryption control bit is set to one, it will cause that the switches SW1-1 and SW1-2 are in left position and system will perform variable ordered encryption according to the status of type selection. Type selection controller in Figure 7 used control bits $T_4T_3T_2T_1T_0$ to achieve type selection, where $T_4$ is used for 1st-ordered GCAT and 2nd-ordered GCAT selection, $T_3$ is $CA_{\mu}(i)$ and $CA_{\mu}(i-1)$ selecting bit, $T_2$ is EXOR and NEXOR selecting bit, $T_1$ is the control bit for selecting $LS_1$ and $LS_2$, and $T_0$ is used for $E(i-1)$ and $E(i-2)$ selection. Type selection control bits and its corresponding GCAT are listed in TABLE 1, for example, $T_4T_3T_2T_1T_0 = 10000_2$ will cause system to perform group 1 of 2nd-ordered recursive CA-encrypted/decrypted substitutions because $T_4T_3T_2T_1T_0 = 10000_2$ makes switches SW2-1 and SW2-2 to change position from left to right, switch SW2-3 is in up position, and all other switches are keeping in their current positions.

Figure 5. Architecture of the variable ordered recursive CA encrypted/decrypted substitutions

Figure 6. Scheme of CA encryption/decryption

We then developed the 1st-ordered and the 2nd-ordered recursive CA-encrypted substitutions to achieve stream cipher as Eqs. (2) and (3), respectively. $GCAT1(E(i-1)), CA_{\mu}(i) \lor CA_{\mu}(i-1))$ in Eq. (2) means that $E(i-1)$ and $CA_{\mu}(i)$ or $CA_{\mu}(i-1)$ execute the 1st-ordered GCAT. Meanwhile, $GCAT2(E(i-1),$
$E(i-2), CA_p(i), CA_p(i-1)$ in Eq. (3) means that $E(i-1)$, $E(i-2)$, $CA_p(i)$, and $CA_p(i-1)$ execute the 2nd-ordered GCAT. Reversing the operations of the recursive CA encryption is to perform the recursive CA decryption. The 1st-ordered and the 2nd-ordered recursive CA decrypted substitutions can be expressed as Eqs. (4) and (5) respectively. Notably, the generalized CA transforms $GCAT(E(i-1), CA_p(i), CA_p(i-1))$ and $GCAT_2(E(i-1), E(i-2), CA_p(i), CA_p(i-1))$ for encryption and for decryption are identical. Since the CA encryption/decryption scheme is lossless, the sequence of $N$-bit decrypted data $D(i), 0 \leq i \leq L_1-1$ is identical to the original sequence $F(i), 0 \leq i \leq L_1-1$. To achieve the goal of data security, we further develop the scheme of CA encryption/decryption (stream cipher system) as Figure 6, where the CA generating scheme shown in dash-line block is controlled by CA key to generate N-bit CA data sequence $CA_p(i), 0 \leq i \leq L_1-1$ for CA substitution. For CA encryption, the encryption/decryption control bit is set as 1; simultaneously, the input is a sequence of $N$-bit data and the output of recursive CA-encrypted substitution is a sequence of $N$-bit encrypted data.

\[ \text{TABLE 1. GENERALIZED CA TRANSFORM (GCAT)} \]

<table>
<thead>
<tr>
<th>Groups</th>
<th>Operations</th>
<th>Type selection control bits $T_4T_3T_2T_1T_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: GCAT(E(i-1), CA_p(i))</td>
<td>$(E(i-1)+LS) \oplus CA_p(i) \mod 2^8$</td>
<td>000xx</td>
</tr>
<tr>
<td>2: GCAT(E(i-1), CA_p(i))</td>
<td>$(E(i-1)+LS) \oplus CA_p(i) \mod 2^8$</td>
<td>001xx</td>
</tr>
<tr>
<td>3: GCAT(E(i-1), CA_p(i-1))</td>
<td>$(E(i-1)+LS) \oplus CA_p(i-1) \mod 2^8$</td>
<td>010xx</td>
</tr>
<tr>
<td>4: GCAT(E(i-1), CA_p(i-1))</td>
<td>$(E(i-1)+LS) \oplus CA_p(i-1) \mod 2^8$</td>
<td>011xx</td>
</tr>
<tr>
<td>5: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-1)+LS) \oplus CA_p(i)) + ((E(i-2)+LS) \oplus CA_p(i-1)) \mod 2^8$</td>
<td>10000</td>
</tr>
<tr>
<td>6: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-1)+LS) \oplus CA_p(i)) + ((E(i-2)+LS) \oplus CA_p(i-1)) \mod 2^8$</td>
<td>10100</td>
</tr>
<tr>
<td>7: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-1)+LS) \oplus CA_p(i-1)) + ((E(i-2)+LS) \oplus CA_p(i)) \mod 2^8$</td>
<td>11000</td>
</tr>
<tr>
<td>8: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-1)+LS) \oplus CA_p(i-1)) + ((E(i-2)+LS) \oplus CA_p(i)) \mod 2^8$</td>
<td>11100</td>
</tr>
<tr>
<td>9: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-2)+LS) \oplus CA_p(i)) + ((E(i-1)+LS) \oplus CA_p(i-1)) \mod 2^8$</td>
<td>10010</td>
</tr>
<tr>
<td>10: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-2)+LS) \oplus CA_p(i)) + ((E(i-1)+LS) \oplus CA_p(i-1)) \mod 2^8$</td>
<td>10110</td>
</tr>
<tr>
<td>11: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-2)+LS) \oplus CA_p(i)) + ((E(i-1)+LS) \oplus CA_p(i-1)) \mod 2^8$</td>
<td>11010</td>
</tr>
<tr>
<td>12: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-2)+LS) \oplus CA_p(i-1)) + ((E(i-1)+LS) \oplus CA_p(i)) \mod 2^8$</td>
<td>11101</td>
</tr>
<tr>
<td>13: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-2)+LS) \oplus CA_p(i-1)) + ((E(i-1)+LS) \oplus CA_p(i)) \mod 2^8$</td>
<td>10101</td>
</tr>
<tr>
<td>14: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-2)+LS) \oplus CA_p(i)) + ((E(i-1)+LS) \oplus CA_p(i-1)) \mod 2^8$</td>
<td>11101</td>
</tr>
<tr>
<td>15: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-2)+LS) \oplus CA_p(i)) + ((E(i-1)+LS) \oplus CA_p(i-1)) \mod 2^8$</td>
<td>11110</td>
</tr>
<tr>
<td>16: GCAT(E(i-1), E(i-2), CA_p(i), CA_p(i-1))</td>
<td>$((E(i-2)+LS) \oplus CA_p(i)) + ((E(i-1)+LS) \oplus CA_p(i-1)) \mod 2^8$</td>
<td>11111</td>
</tr>
</tbody>
</table>

1st-ordered CA encryption: $E(i) = \lceil F(i)+GCAT(E(i-1), CA_p(i) \lor CA_p(i-1)) \rceil \mod 2^8, 0 \leq i \leq L_1-1$;  
2nd-ordered CA encryption: $E(i) = \lceil F(i)+GCAT_2(E(i-1), E(i-2), CA_p(i), CA_p(i-1)) \rceil \mod 2^8, 0 \leq i \leq L_1-1$.  
1st-ordered CA decryption: $D(i) = \lceil E(i) - GCAT(E(i-1), CA_p(i) \lor CA_p(i-1)) \rceil \mod 2^8, 0 \leq i \leq L_1-1$.  
2nd-ordered CA decryption: $D(i) = \lceil E(i) - GCAT_2(E(i-1), E(i-2), CA_p(i), CA_p(i-1)) \rceil \mod 2^8, 0 \leq i \leq L_1-1$.  

© 2010 ACADEMY PUBLISHER
The proposed recursive CA encrypted substitution satisfies both confusion and diffusion properties. The confusion and diffusion properties are achieved by transforming the sequence \( F(i) \), \( 0 \leq i \leq L_1 - 1 \) into the sequence \( E(i) \), \( 0 \leq i \leq L_1 - 1 \) according to Eqs. (2) and (3). The sequence \( E(i) \), \( 0 \leq i \leq L_1 - 1 \) yields uniformly distributed pixels because the high-quality random key-stream \( CA_y(i) \), \( 0 \leq i \leq L_1 - 1 \) and/or \( CA_x(i-1) \), \( 0 \leq i \leq L_1 - 1 \) are used in the transformation. The sequence \( E(i) \), \( 0 \leq i \leq L_1 - 1 \) has the diffusion property because a single change in value \( F(i) \) changes \( E(i) \) which changes \( E(i+1) \) which changes \( E(i+2) \) and changes propagate up to the end of the sequence.

SCAN-CA-based Image security system using the proposed stream cipher system is shown in Figure 7. The security keys for encryption and decryption consist of four components, namely, SCAN key, data reformation key, type selection key and CA key. These keys are identical and are known to both the sender and the receiver before the communication of encrypted image. The scan key is represented by encryption-specific SCAN language and is used for assigning the scanning path to rearrange the pixels of the input image. The data reformation key has 2 bits assigning the scanning path to rearrange the pixels of encryption-specific SCAN language and is used for image. The scan key is represented by the receiver before the communication of encrypted keys are identical and are known to both the sender and receiver site. Suppose receiver has received data SCAN key, reformation key, type selection key, CA key, and a sequence of N-bit encrypted data. The decryption will be done as follows. Firstly, the recursive CA decrypted substitution performs CA decryption to generate the sequence of N-bit decrypted data \( D(i) \), \( 0 \leq i \leq L_1 - 1 \). Inverse data reformation was next done to produce the decrypted data with the original data type, and then the SCAN key rearranges these data into 2-D array as original input image.

The proposed recursive CA substitution makes the exhaustive searching attack much harder and to enhance the performances of system.

IV. POSSIBLE SECURITY KEYS AND CRYPTANALYSIS

Let \( S(n) \) be the number of scan patterns of an 2-D \( 2^n \times 2^n \) array generated by the SCAN key defined by the encryption-specific SCAN language. For a \( 2^n \times 2^n \) image, there are eight basic scan patterns shown in Figure 1 each with eight transformations resulting in 64 basic scan-transformation patterns. When \( n \geq 3 \), there are additionally 24 ways shown in Figure 2 to partition the image into four sub-regions of size \( 2^{n-1} \times 2^{n-1} \) each having \( S(n-1) \) recursive scan patterns. This results in \( S(2) = 64 \) and \( S(n) = 64 + 24(S(n-1))^2 \), \( n \geq 3 \). However, only a portion of scan patterns with a finite number of scan iterations shall achieve a good dispersion, the length of scan key was thus carefully determined as 46 bits, meaning that only \( 2^{46} \) scan patterns shall be used in our simulation.

A 2-D hybrid \( N \times N \)-cell dual-state von Neumann hybrid CA that runs over \( T \) time steps produces \( 2^{N^2+3} \times (T \times N)! \) possible groups of \( T \times N \) N-bit CA data. However, for efficient computer simulation, a special CA rule generator with \((6+n)\)-bit rule control data is used to specify specific CA rule numbers, where \( n = \log_2 N \). Furthermore, since \( T \times N \) N-bit generalized CA data have \((T \times N)! \) possible permutations, it could be a very large number. Therefore, the linear permutation with \( n_r = \log_2(T \times N) = \log_2 T + \log_2 N = n_r + n_N \) bits
is compact in representing and generating a specific set of permutations. In summary, in the computer simulation, \(2^{(8n+3)(8n+4)}\) possible groups of \(T \times N\) N-bit generalized CA data were used. Notably, the proposed system specifies four data types and \(2^{5+2N}\) GCAT types. The basic idea of the proposed stream cipher system is that it uses a specified key to generate a key-stream and uses it to encrypt a plaintext string according to Eqs. 2 and 3. We hence choose the length of key-stream at least equals to the length of plaintext to match the goal of security. Additionally, we know that the length of a CA state cycle is very important in determining the suitability of the CA as a generator of random numbers. According to [30], the average cycle length for 2-D \(N \times N\) -cell dual-state von Neumann CA increases exponentially and is on the order of \(2^{N^2-3}\) for \(N < 8\) or \(2^{N^2-4}\) for \(N \geq 8\). Therefore, we have to choose a suitable 2-D \(N \times N\) -cell CA to produce high quality key-stream for encryption according to the size of image. The relationship between the image size and the minimum size of a suitable 2-D CA is described as follows. If an image is with size of \(2^n \times 2^n\) want to be encrypted, then the minimum size of a suitable 2-D CA will be longer than \((46)^+\) and is \(8^3\) larger than the size of tested images, i.e. \(2^{16} = 256 \times 256\). Note that the size of 2-D CA is not more than \(8 \times 8\) according to [30] even the image size is more than \(4096 \times 4096\). However, we used CA with different size of \(4 \times 4\) until \(32 \times 32\) to show its possible CA keys are high volume increasing.

In summary, in the computer simulation, \(2^{50}\) scan patterns and \(2^{(2+5+2N)}(6+n)N^2+4N+n_p\) possible groups of \(T \times N\) N-bit generalized CA data were used. Notably, the proposed system specifies four data types and \(2^{5+2N}\) GCAT types. Thus, TABLE II presents a high volume of keys.

**TABLE II. POSSIBLE SECURITY KEYS OF THE SCAN-CA-BASED IMAGE SECURITY SYSTEM**

<table>
<thead>
<tr>
<th>Possible SCAN keys for (2^n \times 2^n = 256 \times 256) images</th>
<th>Possible data reformation keys</th>
<th>Possible GCAT type selection</th>
<th>2-D (N \times N) von Neumann CA</th>
<th>Time steps (T)</th>
<th>Possible CA keys (2^{(6+n)N^2+4N+n_p})</th>
<th>Possible security keys</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&gt; 10^{9568})</td>
<td>4</td>
<td>(2^{5+2N})</td>
<td>(4 \times 4)</td>
<td>32768</td>
<td>(2^{217})</td>
<td>(&gt; 10^{9568})</td>
</tr>
<tr>
<td>8 \times 8</td>
<td>8192</td>
<td>(2^{891})</td>
<td>(&gt; 10^{9770})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 \times 16</td>
<td>2048</td>
<td>(2^{1039})</td>
<td>(&gt; 10^{10088})</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32 \times 32</td>
<td>512</td>
<td>(2^{7560})</td>
<td>(&gt; 10^{14783})</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cryptosystems must withstand the most types of attack such as ciphertext only attack, known plaintext attack, chosen plaintext attack, and chosen ciphertext attack. We had shown that our image encryption method satisfies the perfect secrecy condition \(P[F = F(i) | E = E(i)] = P[F = F(i)] \forall i\) in [26], that is, the cryptanalyst can yield no information about the plaintext by observing the ciphertext because of the system’s perfect secrecy. This result proves that the system can withstand ciphertext only, and chosen ciphertext attacks. The cryptanalyst can use known plaintext and chosen plaintext attacks to this scheme guess the security key because the cryptanalyst cannot obtain information about the plaintext by observing the ciphertext. For known plaintext and chosen plaintext attacks, cryptanalyst can do exhaustive key search attacks to guess the security key; however, it is a difficult task because the proposed system has many security keys with variable length \(2^{(46)^+}(6+n)N^2+4N+n_p\) security keys with variable length \((46)^+ + (5 + 2N) + ((6+n)N^2 + 4N + n_p)\) bits, for example, the proposed system has \(2^{190}\) 190-bit and \(2^{350}\) 430-bit possible security keys to encrypt 256 \(\times 256\) images for the 2-D von Neumann CA with size of \(8 \times 8\) and \(16 \times 16\) respectively. Moreover, cryptanalyst has to know the exact length of the security keys before he mounts his exhaustive key search attacks. Cryptanalyst knows that the security key consists of four sub-keys: SCAN key, data
reformation key, type selection key, and CA key under the assumption that the length of the variable length security key is known, he hence wants to do divide and conquer attacks to separately and sequentially guess the individual sub-keys. The complexity of the worst case in the divide and conquer attack could be $2^6 + 2^4 + 2^{5 + 2N} + 2^{6 + 1} + 4N + r_{cy}$ due to all sub-keys are independently decided by authorized users. However, cryptanalyst really difficulty gets any information from the proposed system to guess all sub-keys because we have developed DSP+FPGA implementation to prevent the divide and conquer attacks. Finally, we will discuss a kind of security threat in which we assume that cryptanalyst gets the recovered part of key-stream to reveal the CA key. In this case, minimum length of the recovered part of key-stream is $2^{-3N^2(3N \times 1)}$ bytes. Cryptanalyst firstly mounts brute force search attacks with the complexity of $2^{46} + 2^7 + 2^9 + 2^{10} + 2^{11} + 2^{12}$ to guess the rule control bits, the boundary condition bits, and the permutation control bits of CA key. Next, cryptanalyst uses the recovered rule control bits, boundary condition bits, and permutation control bits to do brute force search attack with the complexity of $2^{46}$ to guess the initial bits of CA key, once the states of 2-D von Neumann CA have been built that run over 8192 time steps to generate CA data sequence of size 65536. Then 8-bit permutation control bits $0_{128}$ in CA key guides the system to do linear permutation from the first 8-bit of the CA data sequence to generate the $10_{11101}$ and $12_{11101}$. We used 1st-ordered GCAT with $L_{S1} = 128$, and 2nd-ordered GCAT with $L_{S2} = 128$ and $L_{S2} = 128$ to perform the recursive CA encryption and decryption substitutions.

V. SOFTWARE SIMULATION RESULTS AND DSP+FPGA IMPLEMENTATION

A. Software Simulation Results

The proposed stream cipher system performed well encryption not only the general text data but also compressed images and uncompressed images. Several simulations of SCAN-CA-based image security system were conducted to test various properties of the proposed data security system that include confusion and diffusion properties. All of the software implementations were performed using personal computer with Intel® P4 CPU (3.2 GHz), Microsoft® Windows® XP, and Borland® C++ builder® 6.0. In our simulations, $Z0(Z0(i0i1i0i1)Z4(d1d0d1d0)X3(i0i1i0i1)X2(i1i0i1i0))$ scan key with number of scan iterations 8 is used to rearrange the pixels of these tested images. The data reformation key is $0_{128}$ which makes the data type for encryption and decryption to be 8-bit. The CA key is selected that chooses $6C_{16}$ uniform initial states, zero boundaries with cyclic boundary at lower right corner, and a special rule control to produce 2-D hybrid $8 \times 8$ von Neumann CA as Figure 8. Once the 2-D hybrid $8 \times 8$ von Neumann CA has been built that run over 8192 time steps to generate CA data sequence of size 65536. Then 8-bit permutation control bits $0_{128}$ in CA key guides the system to do linear permutation from the first 8-bit of the CA data sequence to generate the $CA_{p(i-1)}, 0 \leq i \leq L_{r}-1$. We used 1st-ordered GCAT with $LS_{1} = 128$, and 2nd-ordered GCAT with $LS_{1} = 128$ and $LS_{2} = 128$ to perform the recursive CA encryption and decryption substitutions.
Note that in all the following experiments, images are used for simulation and all images are of size $256 \times 256$. Figures 9a and 9b show YUV formatted color Lena and Lincoln Tower images that were used for testing the performance of SCAN-CA-based image security system. Encrypted images of Lena and Lincoln Tower are shown in Figure 10. Histograms of the encrypted Lena and the encrypted Lincoln Tower in Figure 11 show that the encrypted images get uniformly distributed pixels. This fact illustrates that the proposed data security system satisfies the confusion property. Encrypted images perform the process of decryption will produce the decrypted images. The decrypted images are exactly identical to the original images. This fact shows that the proposed system works well as our expectation.

In order to determine the diffusion property of the proposed system with respect to images, Lena image was modified by incrementing the value of one randomly chosen pixel by 1. The Y-value of pixel $(0, 0)$ was incremented from 161 to 162. Both the original Lena and the modified Lena were encrypted using the same keys. The pixel-wise absolute difference of two encrypted images is displayed in Figure 12, which shows that the two encrypted images have no similarities even though their original images differ by only one pixel. Thus, it proves the diffusion property of the proposed system with respect to images.
(a) Corrupted ciphertext (encrypted image), (b) Decryption of the corrupted ciphertext, (c) Inverse SCAN of the decrypted image.

We previously mentioned that the SCAN-CA-based image security system is a synchronous cipher because the key stream of the proposed system is generated independently of the plaintext and the ciphertext, making it susceptible to synchronization problems. In a synchronous stream cipher, the sender and receiver must be in step for decryption to be successful. If digits are added or removed from the image during transmission, then synchronization is lost. One approach to solve the synchronization problem is to tag the ciphertext with markers at regular points in the output. However, if a small fraction of the ciphertext is corrupted in transmission, rather than added or lost, then only the corresponding fraction in the plaintext is affected and the error does not propagate to other parts of the plaintext; a large area of the image therefore survives. This characteristic makes the proposed system reliable in transmissions with high error rate. The survival properties of the proposed system are shown in Figure 13. Figure 13c shows that the pixels of the corrupted block are dispersed all over the image after inverse scan, which act noise-like pixels. These noise-like pixels can be filtered using the techniques of low-pass filtering.

### TABLE III. SUMMARIES OF COMPARISONS

<table>
<thead>
<tr>
<th>Item</th>
<th>Proposed system (Enhanced version of [26])</th>
<th>System of [26] with one iteration ($N \times N$)</th>
<th>DCPcrypt</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classify</td>
<td>Stream cipher</td>
<td>Stream cipher</td>
<td>Stream cipher</td>
</tr>
<tr>
<td>Key length (bits)</td>
<td>$length^a$</td>
<td>$length^b$</td>
<td>256</td>
</tr>
<tr>
<td>Complexity of cryptanalysis</td>
<td>$2^{39+6N+N^2+n_y}$</td>
<td>$2^{111+64N+N^2+n_y}$</td>
<td>$2^{306}$ [33]</td>
</tr>
<tr>
<td>Possible GCA type selection</td>
<td>$2^{5+2N}$</td>
<td>6</td>
<td>$2^{112}$ [34]</td>
</tr>
<tr>
<td>CPU encryption time (msec/3 Mbytes)</td>
<td>46 ± 16</td>
<td>31 ± 16</td>
<td>609 ± 16</td>
</tr>
<tr>
<td>CPU decryption time (msec/3 Mbytes)</td>
<td>46 ± 16</td>
<td>31 ± 16</td>
<td>609 ± 16</td>
</tr>
<tr>
<td>Entropy of ciphertext (bits)</td>
<td>7.9999</td>
<td>7.9999</td>
<td>7.9999</td>
</tr>
</tbody>
</table>

**a.** $length^a = 46$ bits (SCAN key) + 2 bits (Data reformation key) + (5+2N) bits (Type selection key) + ($6 + n + N^2 + 4N + n_y$) bits (CA key).

**b.** $length^b = 2$ bits (Data reformation key) + 3 bits (Type selection key) + ($6 + n + N^2 + 4N + n_y$) bits (CA key).

**c.** CPU encryption/decryption time is the CPU processing time for encrypting/decrypting 3 Mb of plaintext/ciphertext excluding the time required for hard disk storage.

### B. DSP+FPGA Implementation

At present, real-time data security has become one of main key problems of real-time multimedia applications. We hence have the motivation to develop a real-time image security system based on DSP and FPGA to evaluate the performances of our proposed system. In the proposed SCAN-CA based image security system, the SCAN approach rearrange 2D image pixels into 1D array which costs a large amount of computations and consumes a lot of memory, we therefore use DSP to implement. Moreover, the variable ordered recursive CA substitutions replace the rearranged pixel values which cost a lot of arithmetic and logic operations, and therefore FPGA is one good solution to implement. Figure 14 shows the architecture of SCAN-CA based image security system where TI DSP development board provides TMS320C6416T DSP processor to serve as SCAN processing unit, Altera GFEC Stratix II FPGA board gives EP2S60F1020 FPGA chip to serve as the processing unit of variable ordered recursive CA substitutions. However, due to TI DSP development board and Altera GFEC Stratix II FPGA board have different the clock rate and the voltage supply, it is hence need a additional connecting board to perform the
interfacing buffer. The detailed architectures of SCAN processing unit, processing unit of variable ordered recursive CA substitutions, and connecting board for interfacing between DSP board and FPGA board are shown in Figs. 15, 16, and 17, respectively.

We used TI Code Composer Studio 3.1 to program SCAN approach and then the compiled code should be downloaded into TI DSP development board using USB560 JTAG emulator. Meanwhile, Altera provided Quartus II 5.0 can be used for the implementation of the variable ordered recursive CA substitutions, and the output POF file should be written into the EPROM of EP2S60F1020 FPGA chip. Figure 18 shows the integration of DSP+FPGA platform of the proposed SCAN-CA-based image security system which can be used to develop the real-time image security system. Readers are referred to [36] for a detailed description of DSP+FPGA platform of the proposed SCAN-CA-based image security system.

Figure 14. Architecture of SCAN-CA-based image security system

Figure 15. Architecture of SCAN processing unit

Figure 16. Architecture of processing unit of variable ordered recursive CA substitutions

Figure 17. Architecture of connecting board for interfacing between DSP board and FPGA board

Figure 18. DSP+FPGA platform of SCAN-CA-based image security system

VI. DISCUSSIONS AND CONCLUSIONS

This paper presented a new stream cipher system based on SCAN and variable ordered recursive CA substitutions. Its security method is based on permutation of the data and replacement of the data values. Permutation is done by scan patterns generated by the SCAN approach. The replacement of data values using variable ordered recursive cellular automata (CA) substitutions. The salient features of the proposed stream cipher system can be summarized as follows. (1) Keys consist of SCAN key, data reformation key, type selection key, and CA key, which are of variable lengths producing a large number of possible keys, more than \(10^{568} \sim 10^{4785}\) - according to the size of the 2-D von Neumann CA. (2) Choosing a suitable size for the 2-D CA, according to the size of the image, enables the system to withstand the cropping-and-replacement attack. (3) The system is economic in consuming computational resources because the encryption/decryption scheme uses integer arithmetic and logic operations. Comparative results
show that the performance of the proposed system is superior to those of [26], RC4, Triple-DES, and AES because of its more complicated cryptanalysis and shorter CPU decryption and decryption time for particular ciphertext entropy. Moreover, this system withstands the survival against attack as well as RC4, Triple-DES, and AES. Finally, we established DSP+FPGA platform for developing real-time image security system. The experiments on such a real system demonstrate that the proposed image security system is suitable for the real-time image security.

ACKNOWLEDGEMENTS

The authors would like to thank anonymous reviewers for their valuable suggestions to improve this paper. The National Science Council of Republic of China under the contract NSC-97-2221-E-239-036-MY3 supported this work.

REFERENCES


Rong-Jian Chen was born in 1958, Taiwan. He received the B.S., M.S., and Ph.D. degrees in electronic engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 1987, 1991, and 1995, respectively. He joined the faculty of the National Lien-Ho Institute of Technology in August 1995, where he was the Chair of the Department of Electronic Engineering during 1999–2001 academic years. At present, he is Associate Professor of the Department of Electronic Engineering, National United University. Dr. Chen had been elected as IEEE Senior Member in 2004-2007. His research interests include digital image/video processing, neural networks, and VLSI design of multimedia system.

Jui-Lin Lai was born in 1955, Taiwan. He received the B.S degree from the Electronic Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, 1984. He received the M.S and Ph.D degree in the Institute of Control Engineering and the Institute of Electronic Engineering from the National Chiao-Tung University, Taiwan, R.O.C., in 1990 and 2004, respectively. He joined the faculty of the National Lien-Ho Institute of Technology in August 1984, where he was the Chair of the Department of Electronic Engineering during 1993–1996 academic years. At present, he is Associate Professor of the Department of Electronic Engineering, National United University. Dr. Lai had been elected as IEEE Senior Member in 2005. His research interests include analog and digital VLSI design, neural networks, and computing architecture, and nanotechnology.

Shi-Jinn Horng was born in 1957, Taiwan. He received the BS degree in Electronics Engineering from National Taiwan Institute of Technology, Taipei, the MS degree in Information Engineering from National Central University, Taiwan, and the PhD degree in Computer Science from National Tsing Hua University, Taiwan, in 1980, 1984, and 1989, respectively. He was a Professor and Dean of the College of Electrical Engineering and Computer Science, National United University, Miaoli, Taiwan. Currently, he is a Professor at the Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology. Dr. Horng has published more than 170 research papers and received many awards; especially, the Distinguished Research Award between 2004 and 2006 from the National Science Council in Taiwan; Outstanding I. T. Elite Award, in 2005; Outstanding EE. Prof. Award, the Chinese Institute of Electrical Engineering; Outstanding Research and Invention Award between 2006 and 2008 from National Taiwan University of Science and Technology.