

# A New Public Key Encryption Scheme based on Layered Cellular Automata

Xing Zhang<sup>1,2</sup>, Rongxing Lu<sup>2</sup>, Hong Zhang<sup>1</sup>, and Chungen Xu<sup>3</sup>

<sup>1</sup>School of Computer Sciences and Engineering, Nanjing University of Science & Technology, Nanjing, China

<sup>2</sup>School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore

<sup>3</sup>School of Science, Nanjing University of Science & Technology, Nanjing, China

[e-mail: zhangxing8928@gmail.com; rxlu@ntu.edu.sg; zhhong@njust.edu.cn; xuchung@njust.edu.cn]

\*Corresponding author: Rongxing Lu

*Received February 24, 2014; revised August 6, 2014; accepted September 10, 2014; published October 31, 2014*

---

## Abstract

Cellular automata (CA) based cryptosystem has been studied for almost three decades, yet most of previously reported researches focus on the symmetric key encryption schemes. Up to now, few CA based public key encryption scheme has been proposed. To fill the gap, in this paper, we propose a new public key encryption scheme based on layered cellular automata (LCA). Specifically, in the proposed scheme, based on the T-shaped neighborhood structure, we combine four one-dimensional reversible CAs (set as the private key) to form the transition rules of a two-dimension CA, where the two-dimension CA is set as the corresponding public key. Based on the hardness assumption of the Decisional Dependent CA problem in LCA, we formally prove the proposed scheme is indistinguishably secure against the chosen-plaintext attack (IND-CPA). In addition, we also use a numeric example to demonstrate its feasibility. Finally, analysis of key space and time efficiency are also carried out along with RSA-1024, and the simulation results demonstrate that our proposed scheme is more efficient.

---

**Keywords:** Public key encryption, reversible cellular automata, layered cellular automata, T-shaped neighborhood, provable security.

---

This research was supported by two research grants from Natural Science Foundations of Jiangsu Province of China [Grant No. BK201123 and BK201123], and the second author would also like to thank the support of Nanyang Technological University under Grant NTUSUG (M4081196) and MOE Tier 1 (M4011177).

<http://dx.doi.org/10.3837/tiis.2014.10.017>

## 1. Introduction

As the explosive growth of information and communication technology (ICT) and its wide applications today, information security has become indispensable and crucial to the success of ICT. Cryptographic technique is an essential component of any secure communication, which ensures the data confidentiality, authentication, integrity and non-repudiation. Typical examples of cryptosystem include Data Encryption Standard (DES) [1], Advanced Encryption Standard (AES) [2], RSA [3], and ElGamal [4]. Besides these typical examples, applying cellular automata (CA) techniques to design cryptosystem is also promising in cryptography. CA can be viewed as a simple model of a spatially extended decentralized system made up of a number of individual components (cells). Each individual cell lies in a specific state and will change over time depending upon the states of its local neighbors. In general, the overall structure can be viewed as a parallel processing device. However, the simple structure, when iterated several times, will produce complex patterns indicating its potential to simulate various sophisticated natural phenomena [5]. CA as a medium for encryption is an attractive idea in theory as most CA can be implanted on very fast hardware [6,7], as well as owing to its inherent features like parallelism, locality, simplicity, unpredictability and homogeneity.

Since Wolfram studied the first secret key process based on CA [8], many researchers have explored several possible cryptographic techniques based on the CA, and CA has become one of the important tools to design cryptographic algorithms. Several variants of CA like two-dimensional and multi-dimensional automata with different types of neighborhood systems have been studied by Tomassini and Sipper for random number generation [9], and recently by Seredinsky et al. [10] and Anghelescu et al. [11] for block encryption. In addition, the concept of Reversible cellular automata (RCA) has also been discussed by Xia et al. for multi-granularity RCA data encryption [12], and the Layered cellular automaton (LCA) has been studied by Ayanzadeh et al. for generating normal random numbers [13].

While most of the investigations on CA-based cryptosystems have been focused on traditional secret key cryptosystems, few CA-based public key cryptosystems has been found in the literature. Guan [14] proposes a public key encryption algorithm used non-homogeneous CA, and the security of this algorithm is based on the difficulty of solving a system of nonlinear polynomial equations. However, he does not give any specifications like key-size, key generation procedure and real life examples [15]. Kari [16] introduces an idea for a public key encryption based on RCA, and poses the question of how to implement the key generation algorithm. Then Clarridge and Salomaa [17] prove that under certain technical assumptions a marker CA has a unique inverse with a given neighborhood, and they use the result to develop a working key generation algorithm for a public key encryption based on RCA originally conceived by Kari. Zhu et al. [18] put forward a public key algorithm, which uses four one dimension RCA to build a Moore neighborhood two-dimensional CA. The securities of these schemes are all based on the trapdoor function, which only achieves the one-way security. As a result, these schemes may not satisfy high level security requirements, i.e., secure against the chosen-plaintext attacks (CPA) [19]. To fill this gap, in this paper, we will define a layered cellular automata (LCA) and derive a new hard Decisional Dependent-CA (D-DCA) problem from the 2D CA reversibility problem. Then, built upon LCA, we propose a new public key encryption scheme and formally prove the proposed scheme is semantically secure against CPA, from the D-DCA assumption. Specifically, the main contributions of this paper are two-fold.

- Firstly, we define a layered cellular automata (LCA) with a new neighborhood structure, T-shaped neighborhood. The LCA can be viewed as a highly parallel system and the encryption schemes based on LCA are more efficient than other traditional secret key cryptosystems [13]. Then we define a new hard Decisional Dependent-CA (D-DCA) problem, which is derived from the hard 2D CA reversibility problem. Built on the LCA, we present a new efficient public key encryption (PKE) scheme that utilizes some one-dimensional reversible CAs to construct the transition rules of 2D CA with T-shaped neighborhood structure, where the 1D CA is set as the private key, and the transition rules of the constructed 2D CA are set as the corresponding public key.
- Secondly, we analyze the security of the PKE scheme. In particular, we apply the provable security technique to formally prove that the PKE scheme is semantically secure against the chosen-plaintext attacks, relative to the Decisional Dependent-CA problem.

The remainder of this paper is organized as follows. In Section 2, we formalize the definition of public key encryption and the corresponding security model. In Section 3, we review the definition of CA, RCA, layered CA and a security assumption, which serve as the basis of our proposed scheme. In Section 4, we present our public key encryption scheme based on layered CA, followed by a formal security proof in Section 5. Then, we give a numerical example to demonstrate the feasibility of the proposed scheme in Section 6, and analyze its strengths in Section 7. Finally, we draw our conclusion in Section 8.

## 2. Definition and Security Model

### 2.1 Notation

Let  $\mathbf{N} = \{1, 2, 3, \dots\}$  denote the set of natural numbers, and  $k \in \mathbf{N}$  be a security parameter. An event is said to be negligible if it happens with a probability less than the inverse of any polynomial in  $k$ . If  $n \in \mathbf{N}$ , then  $0^n$  denotes the string of  $n$  zeros. Let  $Z_p$  be a finite field,  $p$  is a large prime number, then  $s \xleftarrow{R} Z_p$  indicates the process of selecting  $s$  uniformly and at random in  $Z_p$ . If  $A$  is a randomized algorithm, then  $y \leftarrow A(x_1, x_2, \dots)$  denotes the processing of  $A$  on inputs  $x_1, x_2, \dots$ , and  $y$  denotes its output.

### 2.2 Definition

In general, a public key encryption scheme  $\text{PKE} = (\mathbf{Kgen}, \mathbf{Enc}, \mathbf{Dec})$  consists of three algorithms:

- The randomized key generation algorithm **Kgen** takes a system parameter  $k$  as input, and returns a pair  $(pk, sk)$  which consists of a private key  $sk$  and a corresponding public key  $pk$ , we write  $(pk, sk) \xleftarrow{R} \text{Kgen}(k)$ .
- The randomized encryption algorithm **Enc** takes a public key  $pk$ , a random number  $\alpha$ , and a plaintext  $M$  as inputs and returns a ciphertext  $C$ , we write  $C \leftarrow \text{Enc}(pk, \alpha, M)$ .
- The deterministic decryption algorithm **Dec** takes the private key  $sk$  and a ciphertext  $C$  as inputs, and returns the corresponding plaintext  $M$ , we write  $M \leftarrow \text{Dec}(sk, C)$ .

All algorithms should satisfy the standard consistency constraint of public key encryption, i.e., for any message  $M$ ,  $\text{Dec}(sk, C = \text{Enc}(pk, \alpha, M)) = M$ .

## 2.3 Security Model

We recall the standard notion of security of public key encryption schemes in terms of indistinguishability. Concretely, we consider the security notion of a public key encryption scheme is indistinguishable against the chosen plaintext attacks, call it the ‘IND-CPA’ security model for brevity [20]. In IND-CPA, a probabilistic polynomial time-bounded adversary, given a public key, generates two equal-length messages and sends to a challenger, the challenger randomly chooses one of the messages to encrypt and sends the corresponding ciphertext to the adversary. The semantic security means the adversary cannot distinguish which message was encrypted.

**Definition 1: (IND-CPA):** Let  $k$  and  $t$  be integers and  $\varepsilon$  a real number in  $[0,1]$ , and  $PKE$  a secure public key encryption scheme with the security parameter  $k$ . Let  $A$  be an IND-CPA adversary, we consider the following random experiment:

*Experiment*  $Exp_{PKE,A}^{ind-cpa}(k)$

$$(pk, sk) \xleftarrow{R} Kgen(k)$$

$$(M_0, M_1, state) \xleftarrow{R} A$$

$$b \xleftarrow{R} \{0,1\}, C_b \leftarrow Enc(pk, r, M_b)$$

$$b' \leftarrow A(pk, C_b, state)$$

If  $b = b'$  then return  $b^* \leftarrow 1$  else  $b^* \leftarrow 0$

return  $b^*$

If  $Exp_{PKE,A}^{ind-cpa}(k) = 1$ , we say  $A$  success.

We define the success probability of  $A$  via

$$Succ_{PKE,A}^{ind-cpa}(k) = 2 \Pr[Exp_{PKE,A}^{ind-cpa}(k)] - 1 = 2 \Pr[b = b'] - 1$$

The proposed PKE scheme is said to be  $(k, t, \varepsilon)$ -IND-CPA secure, if no adversary  $A$  running in time  $t$  has a success  $Succ_{PKE,A}^{ind-cpa}(k) \geq \varepsilon$ .

## 3. Cellular Automata and Security Assumption

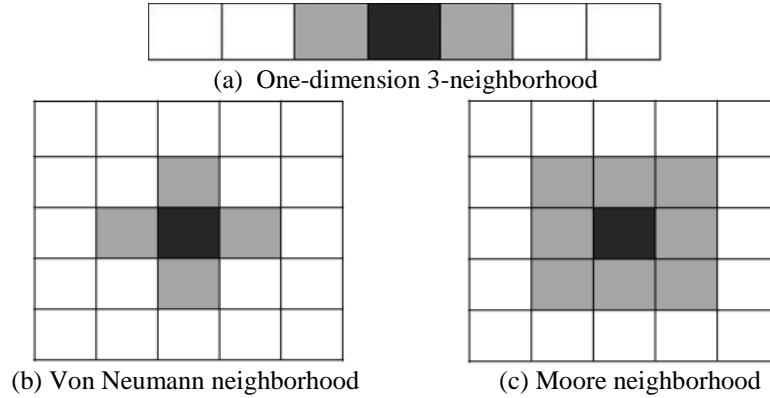
### 3.1 Cellular Automata (CA)

A CA is a discrete model that consists of grids of cells in which each cell can exist in a finite number of states. All cells change their states synchronously, according to a local rule that specifies the new state of each cell based on the old states of its neighbors. A CA is a dynamical system in which space and time are discrete, CAs exhibit some inherent features like parallelism, locality, simplicity, unpredictability and homogeneity, thus CAs are naturally efficient in hardware and software implementations.

A CA can be defined by a quadruple  $\{D, S, N, f\}$  with the dimension  $D$ , the state set  $S$ , the neighboring states set  $N$  and the transition rule  $f$ .

- $D$ . The existing studies of cellular automata mostly focused on one-dimensional (1D) and two-dimensional (2D) CA.
- $S$ . The state set  $S$  holds the set of possible states of all cells in a CA.

- $N$ . There exist various neighborhood structures, and most popular structures are 3-neighborhood, Von Neumann neighborhood and Moore neighborhood, which are respectively shown in Fig. 1 (a), (b), and (c).



**Fig. 1.** Different neighborhood structures

- $f$ .  $f: S \rightarrow S$  is transition rule (transition function).

Let  $s_i^t$  denote the state of the  $i$ -th cell at  $t$  time step and  $s_i^{t+1}$  denote the state of the  $i$ -th cell at  $t+1$  time step, the states of all cells in a CA at  $t$  time step  $(s_0^t, s_1^t, \dots, s_i^t, \dots)$  called a configuration, denoted by  $S^t$ . The state of a cell at the next time step is determined by the transition rule along with its current state and states of neighboring cells, this can be represented by the following formula:  $s_i^{t+1} = f(s_{i-r}^t, \dots, s_{i-1}^t, s_i^t, s_{i+1}^t, \dots, s_{i+r}^t)$ , where  $r$  is the neighborhood radius.

One-dimensional CA with two-state (i.e.  $S = \{0, 1\}$ ) and 3-neighborhood (i.e.  $r=1$ ) called Elementary cellular automata (ECA). The state transition of the cell can be represented as follows:  $s_i^{t+1} = f(s_{i-1}^t, s_i^t, s_{i+1}^t)$ . There are  $8 = 2^3$  possible configurations for a cell and its two immediate neighbors. The rule  $f$  defines the CA must specify the resulting state for each of these possibilities, so there are  $2^{2^3} = 256$  possible rules. Wolfram [29] proposes a scheme, to assign each rule a number from 0 to 255 which have become standard. Each possible configuration is written in order, 111, 110... 001, 000, and the resulting state for each of these configurations is written in the same order and interpreted as the binary representation of an integer, where the number is considered to be the rule number of the automaton. For example, 90 written in binary is 01011010<sub>2</sub>, so rule 90 is defined by the transition rule:

**Table 1.** Rule 90

Rule	111	110	101	100	011	010	001	000
90	0	1	0	1	1	0	1	0

Although the cellular automata is an infinite system, yet it should be finite-dimensional in practical applications. Therefore, it is necessary to define the boundary conditions. Several types of boundary conditions can be considered, such as periodic boundary condition (Fig. 2 (a)), a CA with periodic boundary has the extreme cells are adjacent to each other. Another one is mapped boundary condition (Fig. 2 (b)), which can be obtained by mapping the extreme

cells at the boundary, often this boundary condition can be usefully combined with another, e.g. periodic boundary condition on different boundaries. To simulate a long channel, one would use periodic boundary in horizontal direction, and mapped boundary in vertical direction. The other one is fixed boundary (Fig. 2 (c)), can be obtained by simply prescribing a fixed value for the cells on the boundary. The periodic boundary comes closest to simulate an infinite lattice, and is therefore often used. So, in our proposed scheme, we set the periodic boundary condition for the CA, both in the private key and public key.

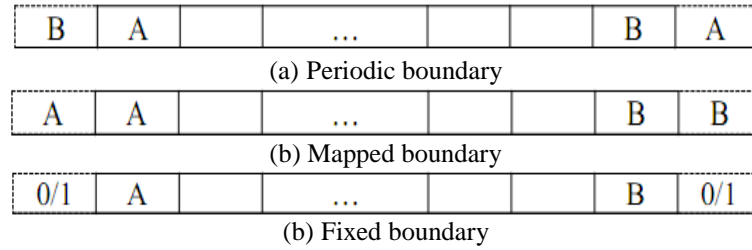


Fig. 2. Different boundary conditions

### 3.2 Reversible Cellular Automata (RCA)

A CA is said to be an RCA if for every current configuration of the CA there is exactly one past configuration. In other words, a CA is reversible, if and only if the transition function is reversible and hence every configuration not only has one successor, but also has one predecessor.

For example, we let  $f$  is the transition rule for moving forward and  $g$  is the transition rule for moving backward, let  $S_i^t = (s_{i-r}^t, \dots, s_{i-1}^t, s_i^t, s_{i+1}^t, \dots, s_{i+r}^t)$  to be the current configuration of the  $i$ -th cell and its neighbors, where  $r$  is the neighborhood radius. Then the successor  $S_i^{t+1}$  of the  $i$ -th cell can be achieved by:  $S_i^{t+1} = f(s_{i-r}^t, \dots, s_{i-1}^t, s_i^t, s_{i+1}^t, \dots, s_{i+r}^t)$ , and the predecessor  $S_i^{t-1}$  of the  $i$ -th cell is:  $S_i^{t-1} = g(s_{i-r}^t, \dots, s_{i-1}^t, s_i^t, s_{i+1}^t, \dots, s_{i+r}^t)$ . If the rule  $g$  is the reverse of the rule  $f$ , a CA moved to  $n$  time steps by using rule  $f$ , the reverse rule  $g$  can be applied till the same number of time steps to obtain the original configuration of the CA.

Concretely, we define four one-dimensional, 4-state and  $\frac{1}{2}$ -radius RCAs, labeled  $CA_1$ ,  $CA_2$ ,  $CA_3$  and  $CA_4$ , where  $CA_i = (1, S, N_r, f_i)$ ,  $(1 \leq i \leq 4)$ ,  $S$  is the state set and  $S = \{0, 1, 2, 3\}$ ,  $N_r$  is neighborhood with  $\frac{1}{2}$ -radius. Set each cell takes the cell at its right position as its neighbor (Fig. 3), the transition rules of these CAs are shown in Table 2.

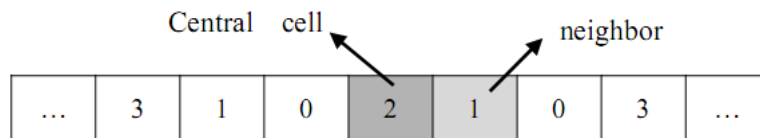


Fig. 3. The neighborhood structure of the 1D RCA

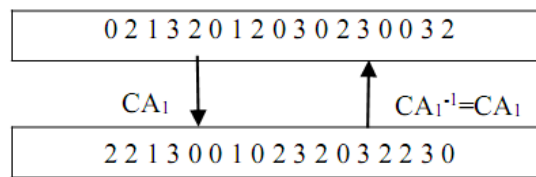
These CAs are all reversible and their reverse rules are themselves. For example, we take

01132012030130032 as the initial configuration, which adopts periodic boundary and right neighborhood structure. Then use rules of  $CA_1$  evolve one time step, and for the new configuration we can return to the initial state by  $CA_1$ .

**Table 2.** Reversible rules of four 1D 4-state 1/2-radius RCAs

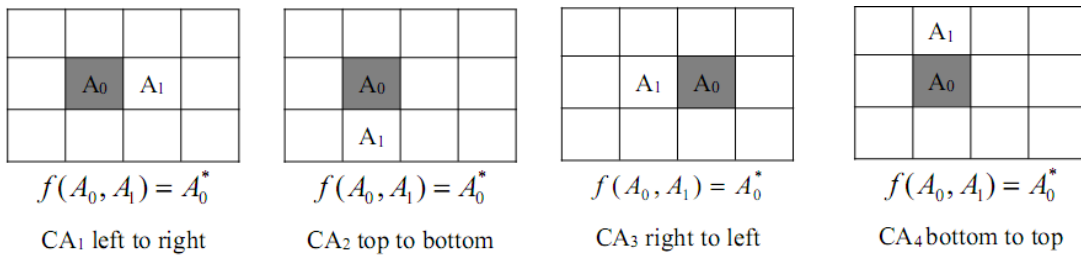
$\begin{matrix} (s_i^t, s_{i+1}^t) \\ s_i^{t+1} \end{matrix}$	$CA_1$	$CA_2$	$CA_3$	$CA_4$
00	2	0	0	1
01	0	1	0	1
02	2	0	0	1
03	2	2	0	0
10	1	1	1	0
11	1	0	1	0
12	1	2	1	0
13	1	3	1	1
20	0	3	2	2
21	2	3	3	2
22	0	1	2	2
23	0	0	2	2
30	3	2	3	3
31	3	2	2	3
32	3	3	3	3
33	3	1	3	3

From the Fig. 4, we can see that  $CA_1$  is reversible, the reverse rule is itself and the other three can also prove to be self-reversing.



**Fig. 4.**  $CA_1$  state transition

In addition, we can set that each RCA has a specific operational direction and each cell in 1D RCA only has one adjacent neighborhood, which is illustrated in Fig. 5.



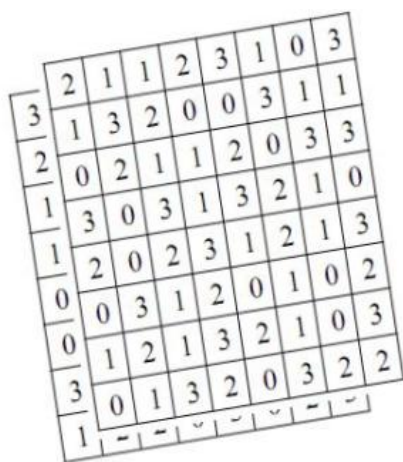
**Fig. 5.** Each 1D CA neighbor structure in construction algorithm



Reversible rules in order to be useful for cryptography only if they should be numerous and exhibit complex behavior. Analysis [21] showed that the ECA turns out that only a small number of rules have the property of being reversible. For example, among all 256 one-radius ECA transition rules, only six are reversible. For CA of two or more dimensions, it has been proved that the reversibility is undecidable for arbitrary rules [22]. So the two-dimensional CA used for encryption has an advantage. Most of the encryption schemes based on the RCA are focused on the symmetric encryption [23-28], there appears to be a very few RCA-based public key encryption schemes in the literature. In this paper, we propose a new public key encryption scheme based on RCA, which utilizes several one-dimensional (1D) RCAs to construct a two-dimensional (2D) CA, the 1D CA set as the private key and the 2D CA set as the corresponding public key. At the same time, we try to construct the encryption scheme based on the layered cellular automata (LCA) with a new T-shaped neighborhood structure, to achieve high level security.

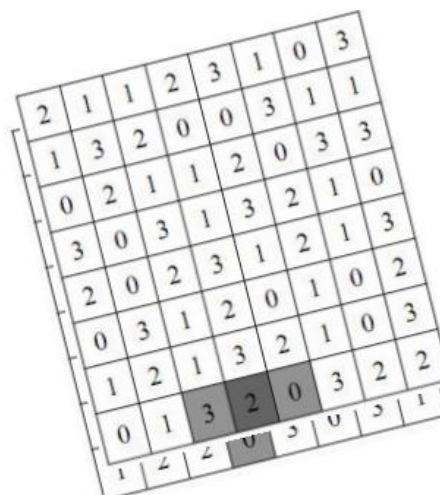
### 3.3 Layered Cellular Automata and T-shaped Neighborhood

A layered cellular automata (LCA) can be viewed as a highly parallel system that consists of layers and each layer consisting of rows of one-dimensional CA, the number of layers can be changed according to actual situation [13]. This stacked structure allows the cell in it has more complex and volatile neighborhood, which may lead to analysis of a new class of CA and is much of theoretical interest. In [28], an 8-layer CA is used in block encryption scheme, the scheme is observed to possess better confusion and diffusion properties when compared with AES, and is more efficient than AES. A two-layer CA and T-shaped neighborhood are shown in Fig. 6 and Fig. 7, respectively.



2	1	1	2	3	1	0	3
1	3	2	0	0	3	1	1
0	2	1	1	2	0	3	3
3	0	3	1	3	2	1	0
2	0	2	3	1	2	1	3
0	0	3	1	2	0	1	0
0	1	2	1	3	2	1	0
3	0	1	3	2	0	3	2

Fig. 6. A two-layer CA



2	1	1	2	3	1	0	3
1	3	2	0	0	3	1	1
0	2	1	1	2	0	3	3
3	0	3	1	3	2	1	0
2	0	2	3	1	2	1	3
0	3	1	2	0	1	0	2
1	2	1	3	2	1	0	3
0	1	3	2	0	3	2	2

Fig. 7. T-shaped neighbor in two-layer CA

From Fig. 7, we can clearly see what T-shaped neighborhood is, i.e., a cell in the first layer, its state changed based on not only its left and right neighbor, but also the cell at the same position in the next layer, so the neighbors of central cell '2' is the left neighbor '3', the right neighbor '0' and the '0' at the next layer. This neighborhood structure joins two layers of the CA and let them become a linked system, can effectively improve the diffusion property of the encryption algorithm with T-shaped neighborhood [23].

In summary, we have the reason to believe that the interaction between layers of the LCA with T-shaped neighborhood structure, in our proposed scheme, will lead to a dynamic and



complex behavior and contribute to achieve better properties of confusion and diffusion.

### 3.4 Security Assumption

The *Reversibility problem of CA*: Kari has proven that the reversibility of two-dimensional (2D) CA is undecidable, even when restricted to CA using the Von Neumann neighborhood [16]. And there does not exist any algorithms that would decide on a given two-dimensional transition rule whether it is reversible or not [22]. So it is impossible that a 2D CA retrace its computation steps backwards in polynomial time, if only known the transition rules.

That is, let  $f_{ca} : Z_p \rightarrow Z_p$  be the transition rule of a 2D CA. For arbitrary message  $m \in Z_p$ , evolved by the transition rule  $f_{ca}$  for  $k$  time steps ( $k \in \mathbf{N}$ ), will change to a new message  $m'$ , i.e.,  $m' = f_{ca}^k(m) \bmod p$ . The reversibility problem is that if given the message  $m'$  and the 2D rule  $f_{ca}$ , we cannot get the initial message  $m$ .

In our proposed encryption scheme, we set  $k \in \mathbf{N}$  as a security parameter and kept private, where the  $k$  is vitally important for the security of the scheme. The bigger  $k$  is, the more difficult the Reversibility problem of CA is. In Section 6, we will give a numerical example to discuss the influences of different  $k$  on the diffusion property of our proposed scheme.

Then we define two hard problems, the Computation Dependent-CA (C-DCA) problem and the Decisional Dependent-CA (D-DCA) problem, all based on the Reversibility problem of CA. If the Reversibility problem of CA can be solved, then we can solve the C-DCA problem, later we can solve the D-DCA problem. Based on these hard problems, we give a Decisional Dependent-CA (D-DCA) Assumption, in section 5 we will formally prove our proposed scheme is IND-CPA security, based on the D-DCA assumption.

#### Definition 2. (Computation Dependent-CA (C-DCA) problem)

Let  $f_{ca} : Z_p \rightarrow Z_p$  be the transition rule of a 2D CA and  $F_{ca} : Z_p \rightarrow Z_p$ , where  $F_{ca} = f_{ca}^k$ ,  $k \in \mathbf{N}$ . Given  $\alpha = F_{ca}(a)$  ( $a \in Z_p$ ), i.e.,  $\alpha = F_{ca}(a) = f_{ca}^k(a)$ ,  $\alpha$  is the result of  $a$  evolved by transition rule  $f_{ca}$  for  $k$  time steps. The C-DCA problem is for some unknown  $a \in Z_p$ , computing  $F_{ca}(a+1)$ .

The C-DCA assumption holds if for any probabilistic polynomial time adversary, the probability  $Succ(A)$  is negligible, where,

$$Succ(A) = \Pr[A(F_{ca}(a) \bmod p) = F_{ca}(a+1) \bmod p \mid a \xleftarrow{R} Z_p]$$

#### Definition 3. (Decisional Dependent-CA (D-DCA) problem)

The D-DCA problem is stated as follows: Let  $f_{ca} : Z_p \rightarrow Z_p$  be the transition rule of a 2D CA, and  $F_{ca} : Z_p \rightarrow Z_p$ , where  $F_{ca} = f_{ca}^k$ ,  $k \in \mathbf{N}$ . There are two distributions:

$$Rand = \{(\alpha, \beta) = (F_{ca}(a), F_{ca}(c)) \mid a, c \xleftarrow{R} Z_p\}$$

$$DCA = \{(\alpha, \beta) = (F_{ca}(a), F_{ca}(a+1)) \mid a \xleftarrow{R} Z_p\}$$

Where,  $F_{ca}(a)$  means arbitrary  $a \in Z_p$  evolved by the 2D transition rule  $f_{ca}$  for  $k$  time steps. The D-DCA problem is that for given  $(\alpha, \beta) \in Z_p$ , deciding  $(\alpha, \beta) \in Rand$  or

$(\alpha, \beta) \in DCA$ .

The advantage of a distinguisher  $A$  denoted by  $Adv(A)$  and defined by:

$$Adv(A) = |\Pr_{Rand}[A(\alpha, \beta) = 1] - \Pr_{DPKE}[A(\alpha, \beta) = 1]|$$

**Definition 4. (Decisional Dependent-CA (D-DCA) Assumption)**

Given  $f_{ca} : Z_p \rightarrow Z_p$  be the transition rule of a 2D CA, let  $F_{ca} : Z_p \rightarrow Z_p, F_{ca} = f_{ca}^k$ ,  $k \in \mathbf{N}$ , and a pair  $(\alpha, \beta) \in Z_p$ , an adversary takes  $(\alpha, \beta)$  as input and distinguishes  $(\alpha, \beta)$  comes from the *Rand* or the *DCA* distribution. We consider the following random experiment on the D-DCA problem.

Experiment  $Exp_A^{D-DCA}$

if  $b=1$ ,  $\{(\alpha, \beta) = (F_{ca}(a), F_{ca}(c)) \mid a, c \xleftarrow{R} Z_p\}$

Else if  $b=0$ ,

$b \in \{0,1\} \leftarrow A(F_{ca}, (\alpha, \beta))$

that if exist  $\varepsilon \in [0,1]$  is non-negligible and  $Adv(A) \geq \varepsilon$ ,

then return  $b = 1$ , else return  $b = 0$ .

We define the corresponding success probability of  $A$  in solving D-DCA problem via

$$Succ_A^{D-DCA} = \Pr[Exp_A^{D-DCA} = 1]$$

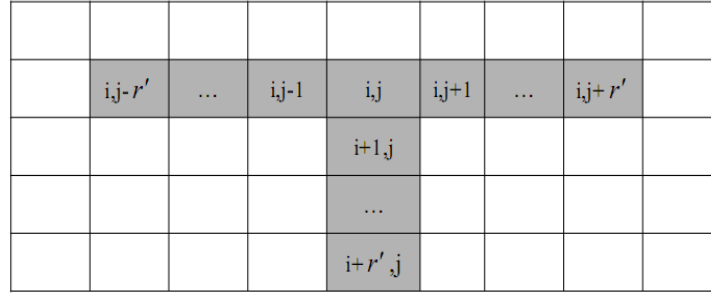
The PKE scheme is said to be  $(t, \varepsilon)$ -secure if no polynomial algorithm  $A$  running in time  $t$  has success  $Succ_A^{D-DCA} \geq \varepsilon$ .

## 4. Proposed Scheme

In this section, we present our public key encryption scheme PKE based on the layered cellular automata with T-shaped neighborhood, which mainly consists of three algorithms, namely **Kgen**, **Enc** and **Dec**.

**Kgen.** Given a security parameter  $k \in \mathbf{N}$ , we define four one-dimension (1D) RCAs, labeled  $CA_1, CA_2, CA_3$  and  $CA_4$ , and each  $CA_i = (1, S, N_r, f_i)$ , where  $S$  is the state set and  $N_r$  is the neighborhood with  $r$ -radius ( $r \in \mathbf{N}$ ). Set each  $CA_i$  has  $n$ -state, where  $n \in \mathbf{N}$  and  $n \geq 2$ , the state set  $S = \{0, 1, \dots, n-1\}$ . Set the transition rule  $f_i : S \rightarrow S$  ( $1 \leq i \leq 4$ ) to be reversible. In addition, we define all 1D CAs with periodic boundary. Choose a random string  $s = a_1 a_2 a_3 a_4$  comprised by integers  $\{1, 2, 3, 4\}$  which defines the order of the four 1D CAs in the generate 2D rules procedure. Set the private key is  $sk = (CAa_1, CAa_2, CAa_3, CAa_4)$ .

We set the corresponding public key is the transition rules of a 2D CA denoted by  $CA^*$ , where  $CA^* = (2, S, N^*, f_{ca})$  is constructed by the 1D RCAs in the private key  $sk$ , so its state set is  $S$ . Set its transition rule  $f_{ca} : S \rightarrow S$ ,  $f_{ca} = f_{a_1} \circ f_{a_2} \circ f_{a_3} \circ f_{a_4}$ . The neighborhood structure is T-shaped neighborhood, we set a number  $r' \in \mathbf{N}$  to define the neighborhood radius of  $CA^*$ . Define a function  $F_{ca} : S \rightarrow S$ ,  $F_{ca} = f_{ca}^k$ .



**Fig. 8.** A  $r'$ -radius T-shaped neighborhood structure in plane

Construct a  $r'$ -radius T-shaped neighborhood structure in a two-dimensional plane, as shown in **Fig. 8**, and set all possible configurations. Take every configuration as the input of the transition rule  $f_{ca} = f_{a_1} \circ f_{a_2} \circ f_{a_3} \circ f_{a_4}$  and evolved successively, the final state of the central cell set as the output. There are  $3r' + 1$  inputs in the transition rule and each has  $n$ -state.

For example, we let  $s_{i,j}^t$  denote the state of the central cell at  $i$ -row and  $j$ -column at  $t$  time step, and the states of the central cell and its  $3r'$  neighbors constitute a configuration, this configuration as the input of the transition rules and evolved to get a new configuration, the final state of the central cell as the output of evolving procedure. This procedure can be presented as follows:

$$\begin{aligned}
 & f_{ca}(s_{i,j-r'}^t, \dots, s_{i,j-1}^t, s_{i,j}^t, s_{i,j+1}^t, \dots, s_{i,j+r'}^t, s_{i+1,j}^t, \dots, s_{i+r',j}^t) \\
 &= f_{a_4}(f_{a_3}(f_{a_2}(f_{a_1}(s_{i,j-r'}^t, \dots, s_{i,j-1}^t, s_{i,j}^t, s_{i,j+1}^t, \dots, s_{i,j+r'}^t, s_{i+1,j}^t, \dots, s_{i+r',j}^t)))) \\
 &= s_{i,j}^{t+4}
 \end{aligned}$$

And we set the map of the states of the central cell and its neighbors to its new state, i.e.,  $f_{ca} : (s_{i,j-r'}^t, \dots, s_{i,j-1}^t, s_{i,j}^t, s_{i,j+1}^t, \dots, s_{i,j+r'}^t, s_{i+1,j}^t, \dots, s_{i+r',j}^t) \rightarrow s_{i,j}^{t+4}$ , is a 2D transition rule. For all cells, performing this procedure to get the new states, and then get the corresponding 2D transition rules.

Set the private key  $sk = (CAa_1, CAa_2, CAa_3, CAa_4)$  and the corresponding public key  $pk = f_{ca}$ .

In general, the value of state number  $n$  is always chosen 2, 3 or 4, i.e. the state set  $S$  is always set as  $\{0,1\}$ ,  $\{0,1,2\}$  or  $\{0,1,2,3\}$ , the radius of the T-shaped neighborhood structure  $r'$  is always chosen 1, 2 or 3. With the increase of the state number and the radius, the number of the constructed 2D transition rules will grow exponentially. The more rules in the public key, the more secure the algorithm will be. However, if the values of the state number  $n$  and the radius  $r'$  are too large, the computation complexity will increase and the efficiency of the algorithm will be greatly reduced.

**Enc.** Randomly chosen a random number  $\alpha \in Z_p$ , given message  $M \in Z_p$  and the public key  $pk = CA^*$ .

- For the random number  $\alpha \in Z_p$ , because the state set of CA in the private and public key is  $S = \{0,1,\dots,n-1\}$ , so coding  $\alpha$  to  $\alpha'$ , where  $\alpha' = \alpha_1\alpha_2\alpha_3\cdots$ ,  $\alpha_i \in S$ . Then

arranged  $\alpha'$  into a layered CA, as the radius of the T-shaped neighborhood of the  $CA^*$  is  $r'$ , we set the layer number equal to  $r' + 1$ , in this way, each cell in the layered CA can select  $r'$  neighbors from the other  $r'$  layers.

For example, if we set  $r' = 3$ , the layered CA consists of 4 layers (Figure 9), and the number of cells in each layer depends on the length of the plaintext. Fig. 9 shadows the cells which are the neighbors of the central cell '2' at the first layer, and the central cell has  $10 = 3r' + 1$  neighbors.

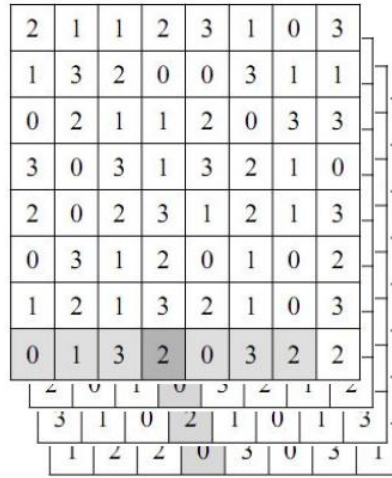


Fig. 9. A 4-layer CA with 3-radius T-shaped neighborhood

- Let  $s_{l,i,j}^t \in Z_n$  ( $l = r' + 1, 1 \leq i \leq a, 1 \leq j \leq b$ ) denote the state of the cell at the  $l$ -layer,  $i$ -row and  $j$ -column at  $t$  time step, where  $a$  is the row number and  $b$  is the column number.
- Select the neighbors of each cell according to the T-shaped neighborhood structure with  $r'$ -radius, and each cell has  $3r' + 1$  neighbors. Take the states of the neighbors as the input of the rule  $F_{ca} = f_{ca}^k$ , then compare with the input of 2D transition rules in the public key and get the corresponding output. That is,

$$s_{l,i,j}^{t+k} \leftarrow f_{ca}^k(s_{l,i,j-r'}, \dots, s_{l,i,j-1}^t, s_{l,i,j}^t, s_{l,i,j+1}^t, \dots, s_{l,i,j+r'}^t, s_{l+1,i,j}^t, \dots, s_{l+r',i,j}^t)$$

All the outputs will comprise the ciphertext of the  $\alpha$ , denoted by  $F_{ca}(\alpha)$ , and then it will be coded into  $F_{ca}'(\alpha) \in Z_p$ .

- Encrypt  $\alpha + 1$  into  $F_{ca}'(\alpha + 1) \in Z_p$ , then compute  $(F_{ca}'(\alpha + 1) + M) \bmod p$ .
- Set  $C_1 = F_{ca}'(\alpha) \bmod p$  and  $C_2 = (F_{ca}'(\alpha + 1) + M) \bmod p$ .
- Set the ciphertext of  $M$  is  $C = (C_1, C_2)$ .

**Dec.** Given a ciphertext  $C = (C_1, C_2)$  and the private key  $sk = (CAa_1, CAa_2, CAa_3, CAa_4)$ .

- Compute the reverse rules of the four 1D CAs in the private key, get  $f_{a_1}^{-1}, f_{a_2}^{-1}, f_{a_3}^{-1}, f_{a_4}^{-1}$ .
- For given  $C_1 = F_{ca}'(\alpha) \bmod p$ , coding it to  $C_1' = c_1 c_2 c_3 \dots$ ,  $c_i \in S$ , arranged into the layered CA, then successively cyclic evolved  $k$  times by four 1D transition rules  $f_{a_1}^{-1}$ ,

- $f_{a_2}^{-1}$ ,  $f_{a_3}^{-1}$  and  $f_{a_4}^{-1}$ , the final states of all cells made up the plaintext  $\alpha'$ . That is,
- $$s_{l,i,j}^{t+k} \leftarrow \left( f_{a_1}^{-1} \left( f_{a_2}^{-1} \left( f_{a_3}^{-1} \left( f_{a_4}^{-1} \left( s_{l,i,j-r'}^t, \dots, s_{l,i,j-1}^t, s_{l,i,j}^t, s_{l,i,j+1}^t, \dots, s_{l,i,j+r'}^t, s_{l+1,i,j}^t, \dots, s_{l+r',i,j}^t \right) \right) \right) \right) \right)^k$$
- Encode  $\alpha'$  to  $\alpha \in Z_p$ , and compute  $\alpha+1 \in Z_p$ , then use the **Enc** algorithm encrypt  $\alpha+1$  into  $F'_{ca}(\alpha+1) \bmod p$ .
  - Compute  $M = (C_2 - F'_{ca}(\alpha+1)) \bmod p$ , i.e., the plaintext of  $C$ .

## 5. Security Analysis

In this section, we formally prove that the ciphertext  $C = (C_1, C_2)$  in the proposed PKE scheme is semantically secure against chosen-plaintext attack under the assumption that the D-DCA problem is hard.

The proposed PKE consists of three algorithms, namely **Kgen**, **Enc** and **Dec**. The private key is  $sk = (CAa_1, CAa_2, CAa_3, CAa_4, k)$  which consists of four 1D RCAs and a security parameter  $k$ , and the corresponding public key is  $pk = CA^*$ . The transition rule of  $CA^*$  is denoted as  $f_{ca} : S \rightarrow S$ ,  $S$  is the state set, set function  $F_{ca} : S \rightarrow S$ ,  $F_{ca} = f_{ca}^k$ . For arbitrary  $a \in Z_p$ , coding  $a$  to  $a' = a_1 a_2 a_3 \dots$ , where  $a_i \in S$ . We set  $F_{ca}(a)$  denote  $a'$  evolved by function  $F_{ca}$ , and code  $F_{ca}(a)$  to  $F'_{ca}(a)$ , where  $F'_{ca} \in Z_p$ .

Assume that there is an adversary  $A$  which runs in polynomial time and has a non-negligible advantage  $\varepsilon$  to break the semantic security of the ciphertext  $C = (C_1, C_2)$  in PKE scheme, then we can construct another adversary  $B$  which has access to  $A$  and achieves a non-negligible advantage to break the D-DCA problem.

First,  $A$  chooses two messages  $m_0 \in Z_p$  and  $m_1 \in Z_p$ , and returns them to  $B$ . At this moment,  $B$  flips a bit  $b \in \{0,1\}$  and generates a ciphertext  $C = (C_1, C_2) = (\alpha, m_b + \beta) \bmod p$ , where  $(\alpha, \beta) \in Z_p$ . In the end,  $B$  sends  $C = (C_1, C_2)$  to  $A$ . After received  $C = (C_1, C_2)$ ,  $A$  returns  $B$  a bit  $b'$  as the guess to  $b$ .  $B$  then returns 1 if  $b' = b$ , else returns 0.

On one hand, if the pair  $(\alpha, \beta) \in Z_p$  comes from the random distribution  $Rand$ , the pair  $(C_1, C_2) \in \{(F'_{ca}(a), m_b + F'_{ca}(c)) \mid a, c \in Z_p\}$  is uniformly distributed, hence independently of  $b$ . Then  $\Pr_{Rand}[B_{success} \mid b' = b] = \frac{1}{2}$ .

On the other hand, when the pair  $(\alpha, \beta) \in Z_p$  comes from  $DCA$  distribution, one can remark that  $(C_1, C_2)$  is a valid ciphertext of  $m_b$ , following a uniform distribution among the possible ciphertexts. Then  $\Pr_{DCA}[B_{success} \mid b' = b] \stackrel{def}{=} \frac{1}{2} \pm \frac{Adv(A)}{2}$ .

The advantage of  $B$  in distinguishing the  $DCA$  and  $Rand$  distributions is

$$Adv(B) = \left| \Pr_{Rand}[B_{success} \mid b' = b] - \Pr_{DCA}[B_{success} \mid b' = b] \right| = \frac{Adv(A)}{2},$$

therefore greater than  $\varepsilon/2$ .

Since  $\varepsilon$  is non-negligible, the above result contradicts with the assumption that the D-DCA problem is hard. As a result, the ciphertext  $C = (C_1, C_2)$  is semantically secure under the chosen-plaintext attack (IND-CPA).

## 6. Numeric Example

In this section, we give a numeric example of our proposed encryption scheme, and discuss the selection of the security parameter  $k$ , which will show that our proposed scheme is correct and feasible.

**Kgen.** Randomly choose the security parameter  $k = 5$ . We set the four one-dimensional 4-state  $1/2$ -radius RCAs (i.e.  $n = 4$ , the state set  $S = \{0, 1, 2, 3\}$ , the radius  $r = 1/2$ ), described in Section 3.2, as the private key, their reversible rules are shown in Table 2. Randomly generate a string  $s = 1234$ , so the private key is  $sk = (CA_1, CA_2, CA_3, CA_4)$ .

Then we define the corresponding public key  $pk = CA^* = (2, S, N^*, f_{ca})$ ,  $CA^*$  is a 2D CA where the transition rule  $f_{ca} = f_4 \circ f_3 \circ f_2 \circ f_1$ , set the radius of the T-shaped neighborhood to be one, i.e.  $r' = 1$ . There are  $(3r' + 1)^n = 4^4 = 256$  possible 2D rules. We set all possible configurations  $(s_{i,j-1}^t, s_{i,j}^t, s_{i,j+1}^t, s_{i+1,j}^t)$  as the input of, where  $s_{i,j}^t \in S$  is the state of the  $i$ -th row  $j$ -th column cell, and the corresponding 2D transition rule can be expressed as  $f_{ca} : (s_{i,j-1}^t, s_{i,j}^t, s_{i,j+1}^t, s_{i+1,j}^t) \rightarrow s_{i,j}^{t+4}$ .

There is a concrete example of the rule generation process in Fig. 10, and  $f_{ca} : (2031) \rightarrow 3$  is a 1-radius 2D rule. Table 3 shown some 2D rules generated by Kgen algorithm.

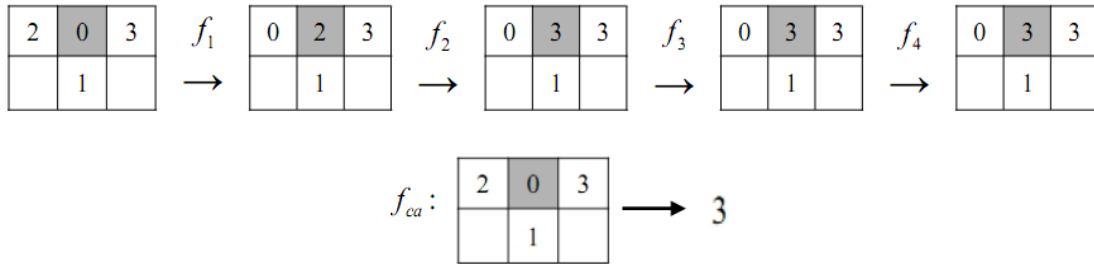


Fig. 10. The process of generating a 2D rule

Table 3. Part of the generated 2D rules

t	0133	1321	3200	2012	1320	3212	2113	1131
t+1	3	3	1	0	3	0	3	1
t	2202	2031	0320	3223	1120	1203	2013	0110
t+1	0	3	2	2	0	3	2	0

**Enc.** Randomly select a large prime number  $p = 37591$ , and a random number  $\alpha = 30930 \in Z_p$ . Given a message  $M = 29753 \in Z_p$ . Because the state of the CA in the

public and private key is  $S = \{0,1,2,3\}$ , so we first coding the number  $\alpha = 30930$  to  $\alpha' = 13203102$ .

- Arrange the  $\alpha'$  into a two-layer CA because of the radius of the T-shaped neighborhood  $r' = 1$ . The structure of the two-layer CA is shown in Fig. 11 (a), and its second layer is shown in Fig. 11 (b).

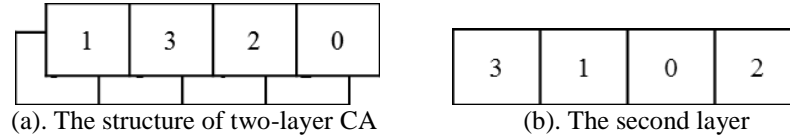


Fig. 11. Two-layer Cellular Automata

- Set  $F_{ca} : S \rightarrow S$ ,  $F_{ca} = f_{ca}^5$ , where  $f_{ca}$  is the transition rule of the public key  $CA^*$ . Take each cell of the two-layer CA shown in Figure 11 and its neighbors as the input of the rule  $F_{ca}$ , the all outputs make up the ciphertext of  $\alpha'$ , that is  $F_{ca}(\alpha) = 33102301$ . Then coding the ciphertext to decimal form  $F'_{ca}(\alpha) \bmod p = 48382 \bmod p = 10791$ .
- Compute  $\alpha + 1 = 30931$ , coding it to  $(\alpha + 1)' = 13203103$ , encrypt it into  $F'_{ca}(\alpha + 1) \bmod p = 17210$ .
- Set  $C_1 = F'_{ca}(\alpha) \bmod p = 10791$ , and  $C_2 = (F'_{ca}(\alpha + 1) + M) \bmod p = 9372$
- So the ciphertext of the message  $M$  is  $C = (C_1, C_2) = (10791, 9372)$ .

**Dec.** Given the ciphertext  $C = (C_1, C_2)$  and the private key  $sk = (CA_1, CA_2, CA_3, CA_4, k = 5)$ . Because the four transition rules of 1D CA in the  $sk$  is self-reversible, i.e.  $f_1^{-1} = f_1$ ,  $f_2^{-1} = f_2, \dots, f_4^{-1} = f_4$ , so we should not compute their reverse rules.

- For  $C_1 = 10791$ , change it to quaternary form 02220213 and arrange it into a two-layer CA.
- Use the transition rules  $f_1, f_2, f_3, f_4$  in  $sk$  to cyclic evolve every cell successively for 5 times. The final configuration of the two-layer CA is the plaintext of  $C_1$ , that is the random number  $\alpha$ .
- Compute and encrypt  $\alpha + 1$  into  $F'_{ca}(\alpha + 1) \bmod p = 17210$ .
- Compute  $M = (C_2 - F'_{ca}(\alpha + 1)) \bmod p = 29753$ , that is the plaintext.

For the security parameter  $k \in \mathbf{N}$ , we give an experiment on the difference between  $F'_{ca}(\alpha)$  and  $F'_{ca}(\alpha + 1)$  when choosing different values. We define a parameter  $\delta \in [0, 1]$  to denote the proportion of the  $f_{ca}^k(\alpha + 1)$  different from the  $f_{ca}^k(\alpha)$ . Based on the Reversibility problem of CA, only more than half bits in  $f_{ca}^k(\alpha + 1)$  are changed, i.e.  $\delta \geq 1/2$ , the semantic security of our proposed scheme is achieved. More detailed, the result is shown in Table 4.



**Table 4.** The comparison of  $f_{ca}^k(\alpha)$  and  $f_{ca}^k(\alpha+1)$  with different  $k$ 

$k$	$f_{ca}^k(\alpha)$	$f_{ca}^k(\alpha+1)$	$F'_{ca}(\alpha) \bmod p$	$F'_{ca}(\alpha+1) \bmod p$	$\delta$
1	33102301	33122302	25050	25563	2/8
2	30023000	30002030	12262	11701	3/8
3	03310330	30030323	15676	12388	6/8
4	20003003	20200210	32963	34852	5/8
5	23303332	10030322	10791	17210	6/8

We can see that the bigger  $k$  is, the better diffusion property we can achieve, which profits to effectively prevent the adversary from solving the D-DCA problem. So in the practical application, the value of  $k$  should be as big as possible on the premise of no obvious reduction of encryption efficiency.

## 7. Performance Analysis

The numeric example has shown the feasibility of the proposed encryption scheme PKE. In this section, we will exhibit its strengths by giving a comparison between the proposed PKE and the algorithm RSA.

Since the radius and state sets of the CAs used in the proposed PKE are not appointed, we can achieve different size key space by varying the radius and state number. Moreover, different key space means more or less calculations and time cost. In our proposed PKE, we set the 2D CA in **Kgen** algorithm has  $n$ -state and T-shaped neighborhood with  $r'$ -radius, so there may be  $n^{n^{3r'+1}}$  possible rules generated as the public key, i.e., the key space is  $n^{n^{3r'+1}}$ . **Table 5** shows the key space size and the time of generating the public key when  $n$  and  $r'$  are chosen different values and compared with RSA. It's obviously observed from the table that the key space increases quickly as  $n$  and  $r'$  become large, and the time cost of generating the public key of the proposed PEK is less than that of RSA.

**Table 5.** The key space and timing analysis between PKE and RSA

	State number	Radius		Key space	Time (ms)
PKE	$n = 2$	$r = 1$	$r' = 1$	$2^{16}$	--
			$r' = 2$	$2^{128}$	15.6
			$r' = 3$	$2^{1024}$	202.8
	$n = 4$	$r = \frac{1}{2}$	$r' = 1$	$4^{256} = 2^{512}$	31
			$r' = 2$	$4^{2^{15}} = 2^{32768}$	4477
RSA				$2^{1024}$	4708

As we know that an RSA algorithm is secure if and only if the public key  $n^* = pq$  is large enough to secure against factorization, where  $p$  and  $q$  are set as the private key. In general,  $p$  and  $q$  are at least chosen as 512 bits large primes so that the corresponding product  $n^*$  will be 1024 bits. Factoring a number with this length that is far beyond the capability of

existing factorization algorithms. Therefore, we consider RSA-1024 to compare with the proposed PKE.

Because the key space of RSA-1024 algorithm is  $2^{1024}$ , as well as the plaintext space and ciphertext space, here we set  $n = 2$  and  $r' = 3$ , such that  $n^{3r'+1} = 2^{1024}$ . Now, we randomly choose 100 plaintexts from the plaintext space and encrypt them to get the ciphertexts, and then decrypt these ciphertexts. All the encryption and decryption are executed by the RSA-1024 and our proposed PKE on an Intel Core 2 Duo 2.0 GHZ, in C++ platform. The average execution time of the 100 encryption and 100 decryption processes are calculated separately and the results are tabulated in [Table 6](#). It is observed that the time taken by our proposed PKE is less than RSA-1024, which obviously demonstrates the efficiency of our proposed scheme.

**Table 6.** Average execution time for RSA-1024 and PKE

	RSA-1024	PKE
Encryption	2.71ms	2.24ms
Decryption	4.19ms	3.8ms

## 8. Conclusion

In this paper, we have proposed an efficient public key encryption scheme based on layered cellular automata. We use four one-dimensional (1D) RCAs to construct a two-dimensional (2D) CA, as the reversibility of 2D CA is undecidable, we set the transition rules of the constructed 2D CA as the public key, the 1D RCAs as the private key. And we have formally shown the proposed encryption scheme is semantically secure against chosen-plaintext attacks (IND-CPA) in the standard model, based on the difficult Decisional Dependent-CA assumption. Moreover, the proposed scheme is developed with a numerical example, and analysis of key space and time efficiency are also carried out along with RSA-1024, and the results demonstrate that proposed scheme is more efficient than RSA-1024.

## References

- [1] Announcing the Data Encryption Standard (DES). *Federal Information Processing Standards Publication 197*. October 25, 1999. [Article \(CrossRef Link\)](#)
- [2] Announcing the Advanced Encryption Standard (AES). *Federal Information Processing Standards Publication 197*. November 26, 2001. [Article \(CrossRef Link\)](#)
- [3] R. Rivest, A. Shamir, and L. Adleman. "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," *Communications of the ACM*, 21 (2): 120–126, February 1978. [Article \(CrossRef Link\)](#)
- [4] T. ElGamal. "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," *IEEE Transactions on Information Theory*, Vol. 31, no. 4, pp. 469-472, 1985. [Article \(CrossRef Link\)](#)
- [5] N. Ganguly, et. al. "A survey on cellular automata," 2003. [Article \(CrossRef Link\)](#)
- [6] Y. Zheng, H. Imai, "A Cellular Automaton Based Fast One-way Hash Function Suitable for Hardware Implementation," *Public Key Cryptography, Lecture Notes in Computer Science*, 1998. [Article \(CrossRef Link\)](#)

- [7] E. Franti, C. Slav and T. Balan, "Design of Cellular Automata Hardware for Cryptographic Application," in *Proc. of CAS2004 Int. Semiconductor Conference*, 2:463-466, 2004. [Article \(CrossRef Link\)](#)
- [8] S.Wolfram, "Random Sequence Generation by Cellular Automata," *Advance in Applied Mathematics*, 1986. [Article \(CrossRef Link\)](#)
- [9] M. Tomassini, M. Sipper, "On the Generation of High-quality Random Numbers by Two-dimensional Cellular Automata," *IEEE Trans. On computers*, Vol. 49, No.10, pp. 1140-1151, 2000. [Article \(CrossRef Link\)](#)
- [10] M. Seredynski, P. Bouvry, "Block Encryption Using Reversible Cellular Automata," *Proceedings of ACRI 2004, LNCS3305*, pp. 785-792, 2004. [Article \(CrossRef Link\)](#)
- [11] P. Anghelescu, S. EmilSofron, "Block Encryption Using Hybrid Additive Cellular Automata," in *Proc. of Seventh International Conference on Hybrid Intelligent Systems*, pp. 132-137, 2007. [Article \(CrossRef Link\)](#)
- [12] X. Xia, Y. Li, Z. Xia and R.Wang, "Data Encryption Based on Multi-Granularity Reversible Cellular Automata," in *Proc. of International Conference on Computational Intelligence and Security*, pp. 192-196, 2009. [Article \(CrossRef Link\)](#)
- [13] R. Ayanzadeh, K. Hassani, Y. Moghaddas, H. Gheiby, S. Setayeshi, "Multi-layer Cellular Automata for Generating Normal Random Numbers," in *Proc. of Numbers Proceedings of ICEE 2010*, 2010. [Article \(CrossRef Link\)](#)
- [14] P. Guan, "Cellular Automaton public key cryptosystem," *Complex System*, Vol. 1, 1987. [Article \(CrossRef Link\)](#)
- [15] P. Joshi, D. Mukhopadhyay, D. RoyChowdhury, "Design and analysis of a robust and efficient block cipher using cellular automata," *Proceeding of Advanced Information Networking and Applications*, 2006. [Article \(CrossRef Link\)](#)
- [16] J. Kari, "Cryptosystems based on reversible cellular automata," 1992. [Article \(CrossRef Link\)](#)
- [17] A. Clarridge, K. Salomaa, "A cryptosystem based on the composition of reversible cellular automata," *LATA2009, LNCS5457*, pp. 314-325, 2009. [Article \(CrossRef Link\)](#)
- [18] B. Zhu, L. Zhou, "Public key cryptosystem based on cellular automata," *Journal of Nanjing University of Science and Technology*, 2007.
- [19] B. Schneier, "Applied cryptography," Wiley, New York, 1996. [Article \(CrossRef Link\)](#)
- [20] R. Lu, X. Lin, X. Liang, X. Shen, "An efficient and provably secure public key encryption scheme based on coding theory," *Security and Communication Networks*, pp. 1440-1447, 2011. [Article \(CrossRef Link\)](#)
- [21] D. Das, A. Ray, "A parallel encryption algorithm for block ciphers based on reversible programmable cellular automata," *Journal of Computer Science and Engineering*, pp. 82-90, 2010. [Article \(CrossRef Link\)](#)
- [22] J. Kari, "Reversibility and surjectivity problems of cellular automata," *Journal of Computer and System Science*, pp. 149-182, 1994. [Article \(CrossRef Link\)](#)
- [23] Y. Wu, L. Hao and J. Chen, "Block cipher based on T-shaped cellular automata," *Journal on Communication*, Vol. 30, 2009.
- [24] M. Seredynski, P. Bouvry, "Block cipher based on reversible cellular automata," *New Generation Computing*, pp. 245-258, 2005. [Article \(CrossRef Link\)](#)
- [25] X. Wang, D. Luan. "A novel image encryption algorithm using chaos and reversible cellular automata," *Commun Nonlinear SCI Number Simulat*, 2013. [Article \(CrossRef Link\)](#)
- [26] A. Kumaravel, O. Meetei, "An Application of Non-uniform Cellular Automata for Efficient Cryptography," *Indian Journal of Science and Technology*, pp. 4560-4566, May 2013. [Article \(CrossRef Link\)](#)
- [27] M. Kishore, S. Kanthi, "A novel encryption system using layered cellular automata," *Proceedings of the World Congress on Engineering 2011*, 2011. [Article \(CrossRef Link\)](#)
- [28] C. Rao, S. Attada, "Implementation of object oriented encryption system using layered cellular automata," *International Journal of Engineering Science and Technology*, pp. 5786-5795, July 2011. [Article \(CrossRef Link\)](#)
- [29] S. Wolfram, "A New Kind of Science," 2002. [Article \(CrossRef Link\)](#)



**Xing Zhang** received the B.S. degree from Xuchang University, China, in 2010. From 2010 to now, she is working her Ph.D. degree in Computer Application from Nanjing University of Science and Technology (NUST), Jiangsu, China. During the period from November 2013 to May 2014, she was also a visiting Ph.D. student at School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Her research interests include information security and cryptography, and the encryption scheme based on cellular automata.



**Rongxing Lu** received the Ph.D degree in computer science from Shanghai Jiao Tong University, Shanghai, China in 2006 and the Ph.D. degree (awarded Canada Governor General Gold Medal) in electrical and computer engineering from the University of Waterloo, Waterloo, Ontario, Canada, in 2012. Since May 2013, he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, as an Assistant Professor. His research interests include computer, network and communication security, applied cryptography, security and privacy analysis for vehicular network, eHealthcare system, and smart grid communications. He won the IEEE Communications Society (ComSoc) Asia Pacific (AP) Outstanding Young Researcher Award in 2013.



**Hong Zhang** is a professor in the Department of Computer Science, Nanjing University of Science and Technology. His current interests are in the areas of theory and technology of information security, data mining and network fault diagnosis.



**Chungen Xu** received the M.S. degree from East China Normal University, Shanghai, China, in 1996 and the Ph.D degree from Nanjing University of Science and Technology in 2003. He is a professor in the Department of Applied Mathematics, School of Sciences, Nanjing University of Science and Technology. His current interests are in the areas of computer and network security, cryptography and coding.