Cellular Automata Based Authentication (CAA)

Monalisa Mukherjee[1], Niloy Ganguly[2], and P. Pal Chaudhuri[1]

[1] Department of Computer Science & Technology, Bengal Engineering College (D.U), Botanic Garden, Howrah, West Bengal, India 711103,
{mona, ppc}@cs.becs.ac.in
[2] Computer Centre, IISWBM, Calcutta, West Bengal, India 700073,
n_ganguly@hotmail.com
monalisa_mukherjee@hotmail.com

Abstract. Current demands for secured communication have focussed intensive interest on 'Authentication'. There is a great demand for a high-speed low cost scheme for generation of Message Authentication Code (MAC). This paper introduces a new computational model built around a special class of Cellular Automata (CA) that can be employed for both message and image authentication. Cryptanalysis of the proposed scheme indicates that compared to other known schemes like MD5, SHA1 etc., the current scheme is more secure against all known attacks. High speed execution of the proposed scheme makes it ideally suitable for real time on-line applications. Further, the regular, modular, and cascadable structure of CA with local interconnections makes the scheme ideally suitable for VLSI implementation with throughput in the range of Gigabits per second.

1 Introduction

The human society is currently living in 'Cyber Age'. Phenomenal technological advances of this age have brought unprecedented benefits to the society. However, at the same time this has generated some unique social problems the human society has never encountered in the history of civilization. The issue of 'Cyber Crime' has become a major challenge for law-makers, government officials, social workers and technologists around the globe. *Secured communication in the networked society of cyber age is a pre-requisite for growth of human civilization of twenty-first century.*

Electronic transfer of all types of digital files demands authentication and verification of data source, protection of copyright and detection of intrusion. A strong trend in the development of the mechanisms for authentication of both message and image is based on cryptographic hash functions designed for MD5 by Rivest. However, hash functions are not originally designed for application in the field of authentication. The conventional MD5 based message authentication, as reported in [1], cannot withstand the cryptanalytic attacks.

This paper reports a simple, high speed, low cost authentication scheme for digital messages and images. It employs the computing model of a special class of Cellular Automata (CA) referred to as GF(2^p) CA. The theory of extension field of GF(2^p) has provided the foundation of this model.

S. Bandini, B. Chopard, and M. Tomassini (Eds.): ACRI 2002, LNCS 2493, pp. 259–269, 2002.

Table 1. The *CA* Rule Table

With XOR (linear *CA*)	With XNOR (complemented rule)
rule 60 : $q_i(t+1) = q_{i-1}(t) \oplus q_i(t)$	rule 195 : $q_i(t+1) = \overline{q_{i-1}(t) \oplus q_i(t)}$
rule 90 : $q_i(t+1) = q_{i-1}(t) \oplus q_{i+1}(t)$	rule 165 : $q_i(t+1) = \overline{q_{i-1}(t) \oplus q_{i+1}(t)}$
rule 102 : $q_i(t+1) = q_i(t) \oplus q_{i+1}(t)$	rule 153 : $q_i(t+1) = \overline{q_i(t) \oplus q_{i+1}(t)}$
rule 150 : $q_i(t+1) = q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)$	rule 105 : $q_i(t+1) = \overline{q_{i-1}(t) \oplus q_i(t) \oplus q_{i+1}(t)}$
rule 170 : $q_i(t+1) = q_{i+1}(t)$	rule 85 : $q_i(t+1) = \overline{q_{i+1}(t)}$
rule 204 : $q_i(t+1) = q_i(t)$	rule 51 : $q_i(t+1) = \overline{q_i(t)}$
rule 240 : $q_i(t+1) = q_{i-1}(t)$	rule 15 : $q_i(t+1) = \overline{q_{i-1}(t)}$

2 CA Preliminaries

A *Cellular Automata* (CA) consists of a number of cells arranged in a regular manner, where the state transitions of each cell depends on the states of its two neighbors and itself (Fig. 1). Each cell stores 0 or 1 in GF(2). The next state function (local transition function) of a cell is defined by one of the 256 (2^{2^3}) rules [4]. Some of the XOR and XNOR rules of GF(2) *CA* are noted in Table 1. A *CA* employing only XOR rules is referred to as Linear, while the ones using both XOR and XNOR are referred to as Additive *CA*. A *CA* with XNOR rules can be viewed as a *CA* with XOR rules and an inversion vector F to account for the XNOR logic function.

Such a CA we have marked as GF(2) CA. In order to enhance the computing power of such a three neighborhood structure, GF(2^p) CA [8] has been proposed.

2.1 GF(2^p) CA

The Fig. 1 depicts the general structure of an n-cell GF(2^p) CA. Each cell of such a CA having p number of memory elements can store an element $\{0, 1, 2, ..., 2^p - 1\}$ in GF(2^p). In $GF(2^p)$ [5], there exists an element α that generates all the non-zero elements, $\alpha, \alpha^2, ...\alpha^{2^p-1}$, of the field. α is termed as the *generator*. α can be represented by a $p \times p$ matrix having its elements as $\{0, 1\} \in GF(2)$. The *matrix representation* of element α^j $(j = 2, 3, \cdots, (2^p - 1))$ for $p = 2$ is shown in Fig. 2.

The connections among the cells of the CA are weighed in the sense that to arrive at the next state $q_i(t+1)$ of i^{th} cell, the present states of $(i-1)^{th}$, i^{th} and $(i+1)^{th}$ are multiplied respectively with w_{i-1}, w_i and w_{i+1} and then added. The *addition* and *multiplication* follows the rule of addition and multiplication defined in GF(2^p). So, under three neighborhood restriction, the next state of the i^{th} cell is given by -
$q_i(t+1) = \phi((w_{i-1}, q_{i-1}), (w_i, q_i), (w_{i+1}, q_{i+1}))$. ϕ denotes the local transition function of the i^{th} cell and w_{i-1}, w_i & $w_{i+1} \in$ GF(2^p) specify the weights of interconnection. A three neighborhood n cell GF(2^p) CA is equivalent to np cell $3p$ neighborhood GF(2) CA. The structure of GF(2^p) CA provides hierarchy and abstraction that can be exploited in many applications [8].

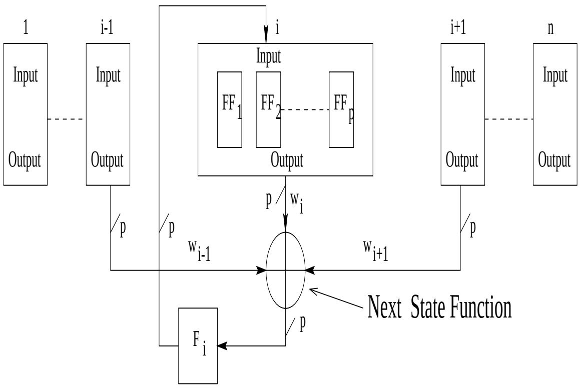

Fig. 1. General structure of a GF(2^p) CA (For p=1, it's a conventional GF(2) CA)

An n cell GF(2^p) CA can be characterized by the $n \times n$ characteristic matrix T, where

$$T_{ij} = \begin{cases} w_{ij}, & \text{if the next state of the } i^{th} \text{ cell} \\ & \text{depends on the present state of the} \\ & j^{th} \text{ cell by a weighed } w_{ij} \in GF(2^p) \\ 0, & \text{otherwise} \end{cases}$$

F = an n symbol inversion vector with each of its element in GF(2^p).

The state of a GF(2^p) CA at time t is an $n-$symbol string, where a symbol $\in$GF(2^p) is the content of a CA cell. If s_t represents the state of the automata at the t^{th} instant of time, then the next state, at the $(t+1)^{th}$ time, is given by

$s_{(t+1)} = T * s_t + F$, and

$s_{(t+n)} = T^n * s_t + (I + T + T^2 + \cdots + T^{n-1}) * F$.

The '$*$' and '$+$' operators are the operators of the Galois Field GF(2^p). If the F vector of GF(2^p) CA is an all zero vector, the CA is termed as linear CA, else it is an Additive CA.

In the CA state transition graph, if all the states lie in some cycles, it is called a group CA. For a group CA, det[T]$\neq$ 0. If the characteristic matrix T is singular, that is det[T] = 0, then the CA is a non-group CA. The T matrix of the example non-group GF(2^2) CA of Fig. 2 has the elements in GF(2^2). Its state transition graph has a single component of an inverted tree with a root (a node with self-loop) referred to as 'Attractor'. Consequently, such a CA is marked as Single Attractor CA ($SACA$).

Definition 1 *Dependency matrix D - if all the non-zero weights in T are replaced by 1 then it is referred to as the dependency matrix of the CA in GF(2^p).*

For $p = 1$, dependency matrix is the characteristic matrix T of GF(2) CA (Fig. 2).

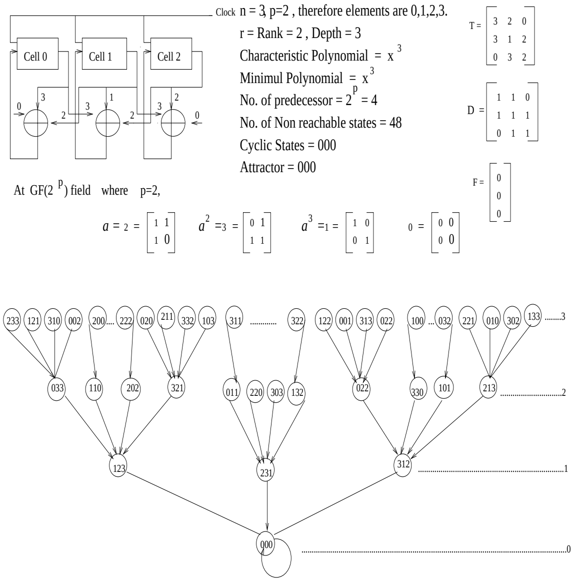

Fig. 2. State Transition Diagram of 3-cell GF(2^2) $SACA$

2.2 Single Attractor Cellular Automata ($SACA$)

The CA belonging to this class and its complemented counterpart referred to as Dual $SACA$ display some unique features that have been exploited in the proposed authentication scheme. The T matrix of an n cell GF(2^p) $SACA$ is an $n \times n$ matrix with its elements in GF(2^p). The rank, characteristic polynomial and minimal polynomial of the T matrix are :
rank(T) $= n - 1$, rank($T \oplus$ I) $= n$, I being the $n \times n$ identity matrix.
Characteristic polynomial $= \alpha x^n$, Mimimal polynomial $= \alpha x^n$, where $\alpha \in$ GF(2^p).
A few theorems are next introduced without proof. The proof is analogous to GF(2) TPSA (Two Predecessor Single Attractor) CA noted in [2].

Theorem 1 : If the rank of the characteristic matrix T of an n cell GF(2^p) non-group CA is $n - 1$, then each reachable state has 2^p predecessors.

Theorem 2 : Depth of an n cell $SACA$ is equal to n

The inversion vector F in the example $SACA$ of Fig. 2 is an all 0's vector. A non-zero F leads to its dual counterpart.

Dual SACA

A dual $SACA$ also referred to as $\overline{SACA}$ results from an introduction of non-zero inversion vector F with the characteristic matrix T of the $SACA$. $\overline{SACA}$ has identical state transition behavior as that of $SACA$ with change of relative

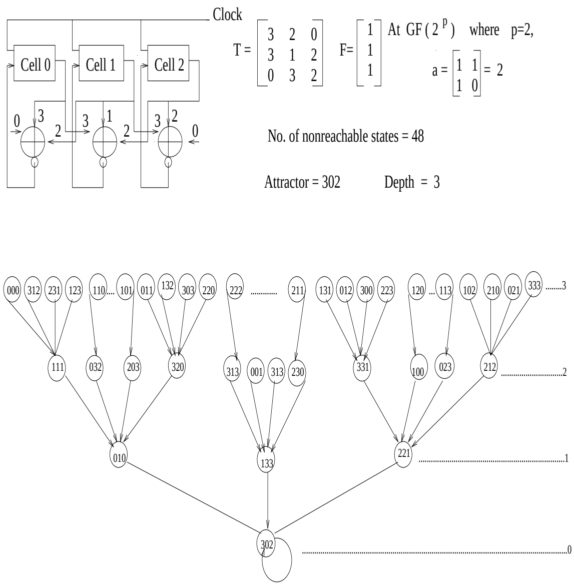

Fig. 3. Structure and state transition graph of a 3 cell GF(2^2) Dual $SACA$

position of states. All the reachable states in a $SACA$ becomes non-reachable in $\overline{SACA}$ [2]. The example CA of Fig. 3 is a dual counterpart of the $SACA$ of Fig. 2. The following Theorem characterizes a $SACA$ and $\overline{SACA}$.

Theorem 3 : If the complement vector F of a GF(2^p) $SACA$ with characteristic matrix T is such that $T^n.F = 0$, and $T^{n-1}.F \neq 0$, then this complemented CA is a dual $SACA$

Detailed characterization of a $SACA$ and its dual are reported in [7]

2.3 Synthesis of $SACA$ and Its Dual

The algorithmic steps for synthesis of an n cell GF(2^p) $SACA$ and its dual are noted below with illustrating example. The **Steps 1** and **2** ensures that the resulting CA is a $SACA$ - the proof is omitted for shortage of space.

Step 1. Generate the dependency matrix D of size $n \times n$ whose 1^{st} cell has no dependency on its neighbors (left, self and right) and the rest of the cells having dependency on its left neighbor only. For a 3 cell GF(2^2) $SACA$, D is: $\begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$

Step 2. Construct characteristic matrix T of the $SACA$ from D by performing elementary row/column operations such that each of the cells has dependency on left, self and right neighbors.

For a 3 cell GF(2^2) $SACA$, T is: $\begin{pmatrix} 3 & 2 & 0 \\ 3 & 1 & 2 \\ 0 & 3 & 2 \end{pmatrix}$

Step 3. Construct $\overline{SACA}$ by implementing the result of Theorem 3.

3 Cellular Automata Based Authentication (CAA) Scheme for Message/Image

The schemes for message and image authentication are noted along with proof of robustness against the attacks. The GF(2) CA based authentication scheme proposed in [9] is insecure against attacks based on Differential Cryptanalysis. The proposed scheme overcomes the problem.

3.1 *SACA* as One-Way Hash Function Generator

The proposed scheme employs *keyed one-way hash* function based authentication using GF(2^p) $SACA$ and its dual $\overline{SACA}$. The one-way hash function maps a secret key and an arbitrary length input message data to a fixed length hash output referred to as message digest .

3.2 CAA for Digital Message

Let, A has a message M to send to B and they share a common secret key K. A calculates message digest $C_K(M)$ from M, and K employing one way $SACA$ based hash function. Message M and digest $C_K(M)$ are transmitted to B where B performs the same function on the received message to generate a new digest $C_K(M^{'})$. The message gets authenticated if $C_K(M)$ and $C_K(M^{'})$ matches.

Algorithm 1 Generate_Message_Digest

Input*: Message M of length* $|M|$ *bits; Private key* $\mathcal{P}$*:* $n \times p$ *bits :*
- n cell GF(2^p*)* $SACA$ *and its dual* $\overline{SACA}$
Output*: Message Digest:* $n \times p$ *bits*
Step 1: *Group Message M into k blocks* $\{ M_1, M_2, \ldots M_k \}$ *each of length n symbol (*$S_1, S_2, \ldots, S_n$*) in GF(*2^p*)*
Let $\mathcal{P}_1 = \mathcal{P}$ *(Private Key)*
For(i=1 to k)
{
Step 2: *Form a tridiagonal matrix* CA_{M_i} *whose n diagonal elements are n-symbols of* M_i*; off diagonal values are 1 and the remaining all values are zero*
Step 3: *Run each of the CAs for one step:*
(a) *Run* CA_{M_i} *with* $\mathcal{P}_i$ *as seed to obtain* ${\mathcal{P}_i}^1$
(b) *Run* $SACA$ *with* ${\mathcal{P}_i}^1$ *as seed to obtain* ${\mathcal{P}_i}^2$
(c) *Run* $\overline{SACA}$ *with* ${\mathcal{P}_i}^2$ *as seed to obtain* ${\mathcal{P}_i}^3$

Step 4: *Let* $\mathcal{P}_{i+1} = \mathcal{P}_i{}^3$
}
Step 5: *Output* $\mathcal{P}_{k+1}$ *as the Message Digest*

3.3 Robustness of CAA for Digital Message

Robustness of the proposed scheme is analyzed against probable attacks.
Attack 1: Brute Force Attack
Birthday attack, Collision attack belong to this category of attacks. An authentication scheme can be made robust against such attacks by increasing message digest/key length. The CAA scheme can easily employ Variable Length key of any size since it employs simple, regular, modular, cascadable structure of CA. So, CAA can be efficiently designed against such attacks.
Attack 2: The Extension Attack or the Padding Attack
This type of attack is not possible for the proposed scheme as it employs a keyed hash function where the key is not a part of the original message.

A detailed description of robustness of CAA against Attack 1 and 2 is reported in [7].
Attack 3: Next the robustness of CAA is tested in respect of the strength of the $SACA$ based hash function employed for the scheme. The attacks are employing much more subtlety than mere brute force attack.

Cryptanalytic attacks attempt to guess whether the function is such that two messages or keys, close to each other in terms of bit distance, produce the outputs which are also close to each other. If it is so, then the code can be broken in much lesser time than exhaustive search. The following two results show that our scheme is protected against such attack.
Result 3(a): Let M be an arbitrary message while $M^{'}$ is another message derived out of M by flipping a randomly chosen bit of M. The corresponding message digests are $C_K(M)$ and $C_K(M^{'})$. From Table 2 this is clear that the difference (performing XOR between $C_K(M)$ and $C_K(M^{'})$) has on the average the same number of zeros and ones.

Table 2 shows that there are equal number of zeros and ones in the output difference which indicates that flipping one bit of a randomly chosen message results in a completely different message digest.
Result 3(b): Let M be an arbitrary message and K be a secret key while $K^{'}$ is another secret key derived out of K by flipping a randomly chosen bit of K. The corresponding message digests are $C_K(M)$ and $C_{K'}(M)$. If the difference between $C_K(M)$ and $C_{K'}(M)$ has almost same number of zeros and ones, then it can be concluded that flipping a bit of Key results in a completely different message digest(Table 2).

In both the attacks, the result becomes better as the value of p increases.
Attack 4: Next CAA is analyzed from the viewpoint of another very important attack called differential attack [10]. The attack analyzes the plaintext pairs along with their corresponding message digest pairs to identify the correlations that would enable identification of the secret key.

Table 2. Results of Result 3(a) and 3(b) on CAA and MD5

Input size of file in bytes	Result 3(a)					Result 3(b)				
	No. of ones for CAA $(C_K(M) \oplus C_K(M^{'})$				No. of ones for MD5 128 bit	No. of ones for CAA $(C_K(M) \oplus C_{K'}(M)$				No. of ones for MD5 128 bit
	key-length 128		key-length 256			key-length 1 28		key-length 256		
	p=4	p=8	p=8	p=16		p=4	p=8	p=8	p=16	
3239	34	70	128	122	69	54	63	134	130	64
3239	56	67	124	132	69	70	66	140	132	69
3239	45	66	122	138	70	52	64	136	126	66
65780	55	76	114	138	64	64	66	130	142	70
65780	57	65	140	128	65	45	64	104	134	68
65780	59	65	118	140	67	66	63	118	120	62
259120	38	62	134	136	70	46	69	122	126	67
259120	51	64	130	130	65	55	64	132	128	66
259120	55	66	132	132	67	48	70	140	128	76

For example, let the length of the plain text and message digest are of 8 bit and the fixed bit difference D taken as 3. For a pair of plaintexts X=11001011, $X^{'}$=10011001, corresponding message digests are (say)
MD=00110101, $MD^{'}$=10000110, i.e, with difference $D^{'}$=5. The value of $D^{'}$ is calculated for all plaintext pairs with D=3. Then from the distribution of $D^{'}$, we can calculate the standard deviation (σ). In general, a one-way hash function is said to be protected from differential cryptanalytic attack if σ is lower than 10 % [12].

We have performed differential cryptanalysis on our scheme with 50 different files having 5 different size. For each file, we take 5 different fixed input differences. Table 3 depicts results of differential cryptanalysis on CAA. From Table 3 (Column 2 to 7) this is clear that as p increases (p is the dimension of Galois field GF(2^p)) σ decreases. The experimental results at Table 3 establish that *CAA* can defend differential attack in a better way than MD5 (Column 6).

Execution time

Comparative results for GF(2) CA based authentication algorithm [9], MD5 and CAA at GF(2^p) in respect of *CPU time* are displayed in the Table 3 (Column 9 to 13). These experimental results establish the higher speed of execution of CAA scheme based on GF(2^p) *SACA*. Higher value of p leads to reduction of computation time because rather than handling $np \times np$ matrix with GF(2) elements we deal with $n \times n$ matrix with GF(2^p) elements. In software the speed is almost one and half times more than MD5 at p=16. The throughput of the Hardwired implementation of scheme is of the order of tens of Gigabits/sec.

3.4 CAA for Watermarking

Digital watermarking research has generally focused upon two classes of watermarks, fragile and robust. Fragile watermarks is ideal for image authentication applications [13,14]. In this watermarking it allows a user with an appropriate

Table 3. Differential cryptanalysis for CAA and Comparative speed of CAA and MD5 (in WindowsNT 4.00-1381,IBM)

Input size of file in bytes	Avg.Std.Devn of XOR Distribution for $SACA$ (%)							CPU Time in Seconds				
	key-length 128				key-length 256			MD5 method	p=1 n=128	p=2 n=64	p=4 n=32	p=8 n=16
	p=1	p=2	p=4	p=8	p=8	p=16						
1608	9.110	8.950	7.881	5.899	5.660	4.883		0.0549	0.055	0.050	0.040	0.040
35860	14.821	12.111	8.458	6.134	6.123	5.123		0.165	0.147	0.105	0.105	0.087
65780	8.989	7.813	6.657	5.034	5.002	4.986		0.193	0.166	0.129	0.110	0.091
142164	6.824	6.771	5.998	4.823	4.989	5.024		0.2198	0.2053	0.1650	0.118	0.081
259120	14.100	11.783	10.213	7.982	6.102	4.033		0.299	0.271	0.267	0.210	0.200
852984	13.015	12.443	7.893	4.342	3.032	4.003		0.330	0.294	0.252	0.205	0.205

secret key to verify the authenticity, integrity and ownership of an image. If the user performs the watermark extraction with an incorrect key or an image which is not watermarked, the user obtains an Image that resembles noise.

Recent systems apply sophisticated embedding mechanisms, including the use of cryptographic hash functions to detect changes to a watermarked image. This section reports a watermarking scheme that employs CAA based hash functions.

Let the original grey-scale image be X. A bi-level watermark ‘A’ will be inserted in it and again will be extracted from it for authentication. X and A are divided into some equal blocks of size $n \times n$ and say, each block of X is termed as X_r and A as A_r.

Insertion

Let, Image block $X_r = \begin{pmatrix} 255 & 128 \\ 108 & 11 \end{pmatrix}$ or, $\begin{pmatrix} 11111111 & 10000000 \\ 01101100 & 00001011 \end{pmatrix}$ and

Watermark block $A_r = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$.

- Set all LSBs of X_r to 0, $X_r{}' = \begin{pmatrix} 1111111\mathbf{0} & 1000000\mathbf{0} \\ 0110110\mathbf{0} & 0000101\mathbf{0} \end{pmatrix}$ is obtained.
- **Hash** $X_r{}'$ **using CAA** and the hash output $H_r = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$.
- Perform pixel by pixel ex-or operation between H_r and A_r, ($H_r \oplus A_r = C_r$) and obtain $C_r = \begin{pmatrix} 0 & 1 \\ 0 & 1 \end{pmatrix}$.
- Insert values of C_r into all LSBs of $X_r{}'$. The resulting watermarked block $X_r{}^w = \begin{pmatrix} 1111111\mathbf{0} & 1000000\mathbf{1} \\ 0110110\mathbf{0} & 0000101\mathbf{1} \end{pmatrix}$ or, $\begin{pmatrix} 254 & 129 \\ 108 & 11 \end{pmatrix}$.

Extraction

- Let, $Y_r = \begin{pmatrix} 254 & 129 \\ 108 & 11 \end{pmatrix}$ or, $\begin{pmatrix} 11111110 & 10000001 \\ 01101100 & 00001011 \end{pmatrix}$ be the watermarked image block.

– Extract all LSBs from Y_r, $C_r = \begin{pmatrix} 0\,1 \\ 0\,1 \end{pmatrix}$ is obtained.
– Set all LSBs of Y_r to 0, and obtain $Y_r{}' = \begin{pmatrix} 1111111\mathbf{0}\ 1000000\mathbf{0} \\ 0110110\mathbf{0}\ 0000101\mathbf{0} \end{pmatrix}$.
– Hash $Y_r{}'$ by CAA and obtain $H_r = \begin{pmatrix} 1\,1 \\ 1\,0 \end{pmatrix}$.
– Perform pixel by pixel ex-or operation between H_r and C_r to obtain water mark image block $A_r = \begin{pmatrix} 1\,0 \\ 1\,1 \end{pmatrix}$.

Analysis and Comparative Study

The inherent advantages of the proposed scheme can be summarised as follows:
(a) The greatest advantage of our scheme is the flexibility of adjusting key size without any overhead. This is possible due to modular structure of Cellular Automata.
(b) The Table 4 shows the result where in each of the image, watermark has been inserted according to proposed scheme (column 4 to 7) gives better PSNR values than MD5 (column 8). Moreover as extension field parameter p increases in CAA, PSNR value improves (Table 4). The robustness of CAA based one-way hash function, as noted in Section 3.3, has resulted in the superior quality watermarked image.

Table 4. Comparison of PSNR values using CAA for different p and MD5

Image name	Data in Bytes	Block size	PSNR Values in dB unit				
			Wong-Memon method				
			p=1	p=2	p=4	p=8	MD5
Sachin	522835	14 x 30	51.126629	51.201994	51.29048	51.541979	51.013072
SkylineArch	964451	72 x 60	52.013388	52.216852	52.427391	52.811862	51.034981
Lena	1064071	60 x 90	53.23367	53.295081	53.463457	53.788033	51.243123
Concord	1485604	80 x 84	53.830272	53.884655	54.020056	54.526984	51.317890
Rabbit	964451	80 x 72	52.177280	52.307440	52.443773	52.725227	51.103782

(c) The most effective attack on image authentication is Holliman-Menon attack or Vector Quantization attack [6]. CAA based watermarking is tuned to counterfeit this attack as a built-in function whereas all other hash functions (including MD5) defend the attack externally which effectively decreases the insertion/extraction speed of watermarking.

4 Conclusion

This paper reports GF(2^p) Cellular Automata (CA) based Authentication (CAA) scheme. The scheme has been employed to insert fragile watermark in images. Security of CAA against known attacks and its execution speed are emphatically better than those of MD5. Future prospective of the CAA lies in building robust watermarking scheme using the proposed CA based one-way hash function and extend the scheme to develop a digital signature scheme for e-commerce application.

References

1. B. Schineier *Applied Cryptography* tJohn Wiley and Sons, 2001.
2. P.Pal Chaudhuri, D.Roy Choudhury, S. Nandi and S. Chattopadhyay *Additive Cellular Automata Theory and Applications.* IEEE Computer Society Press, California, USA, 1997.
3. S. Wolfram *Cryptography with Cellular Automata* Proceedings of Crypto'85, pp.429-432
4. S. Wolfram *Cellular Automata and Complexity* Addition-Wesly Publishing Company, 1994
5. T. R. N. Rao and E. Fujiwara *Error-control Coding for Computer Systems* Prentice-Hall, Englewood Cliffs, N.J, 1989
6. M. Holliman and N. Memon *Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes* IEEE trans. on Image Processing, volume-9, No-3, March,2000
7. M. Mukherjee, B.K.S, N. Ganguly and P. Pal Chaudhuri *GF(2^p) Cellular Automata As A Message Digest Generator* 9 th International Conference on Advance Computing and Communications, December-2001
8. K Paul, D. Roy Chowdhury, P. Pal Chaudhuri *Theory of Extended Linear Machine*, to be published in IEEE, Transaction on Computers.
9. P. Dasgupta, S. Chattopadhyay and I. Sengupta *An ASIC for Cellular Automata based message Authentication* 12 th Int. Conf. on VLSI Design, 1999
10. D. Wagner *Differential cryptanalysis of KHF* 5th Int. Workshop on Fast Software Encryption, 1998
11. B. Preneel and P. van Oorschot *MD-x MAC and building fast MACs from hash functions* CRYPTO 95 Proceedings, Vol.963, D. Coppersmith ed., Springer-Verlag, 1995.
12. E. Biham and A. Shamir *Differential Cryptanalysis of DES-like Cryptosystems* Journal of Cryptology, 4(1991), 3-72
13. J. Fridrich *Security of Fragile Authentication Watermarks with Localization* Proc. of SPIE Photonic West, Electro Imaging 2002, Security and Watermarking of Multimedia Contents, January-2002
14. P. W. Wong and N. Memon *Secret and Public Key Image Watermarking Schemes for Image Authentication and ownership verification* IEEE trans. on Image Processing, volume-10, No-10, October,2001