Data Security and Chaos-based Data Security

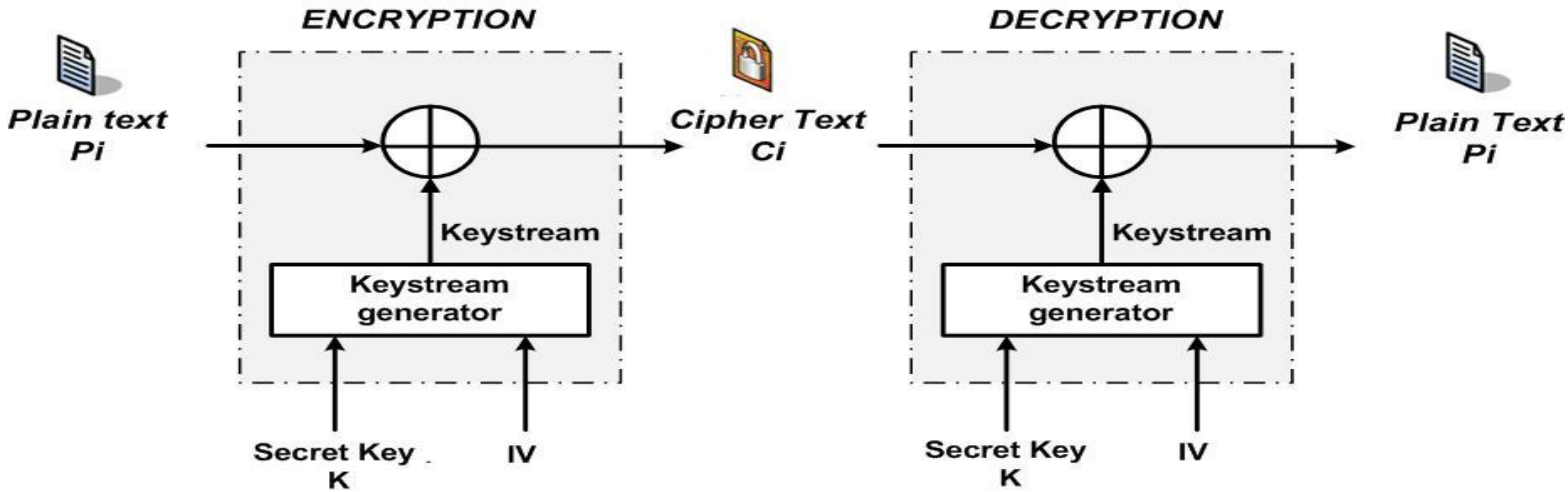
Part-2

Safwan El Assad

Polytech Nantes, school of engineering of the university of Nantes – France

IETR Laboratory, UMR CNRS 6164; VAADER team - site of Nantes

General scheme of a Stream Cipher



Pi	Xi	Ci
0	0	0
0	1	1
1	0	1
1	1	0

Encryption: $Ci = Pi \oplus Xi$ Decryption: $Pi = Ci \oplus Xi$

Encrypt $Pi = 0$*, depending on the keystream bit* $Xi = \begin{cases} 0 \\ 1 \end{cases}$ *gives* $Ci = \begin{cases} 0 \\ 1 \end{cases}$

If the keystream bit Xi *is perfectly random, i.e., it is unpredictable and has exactly 50% chance to have the value 0 or 1, then both Ci also occur with a 50% likelihood. Likewise when we encrypt* $Pi = 1$:

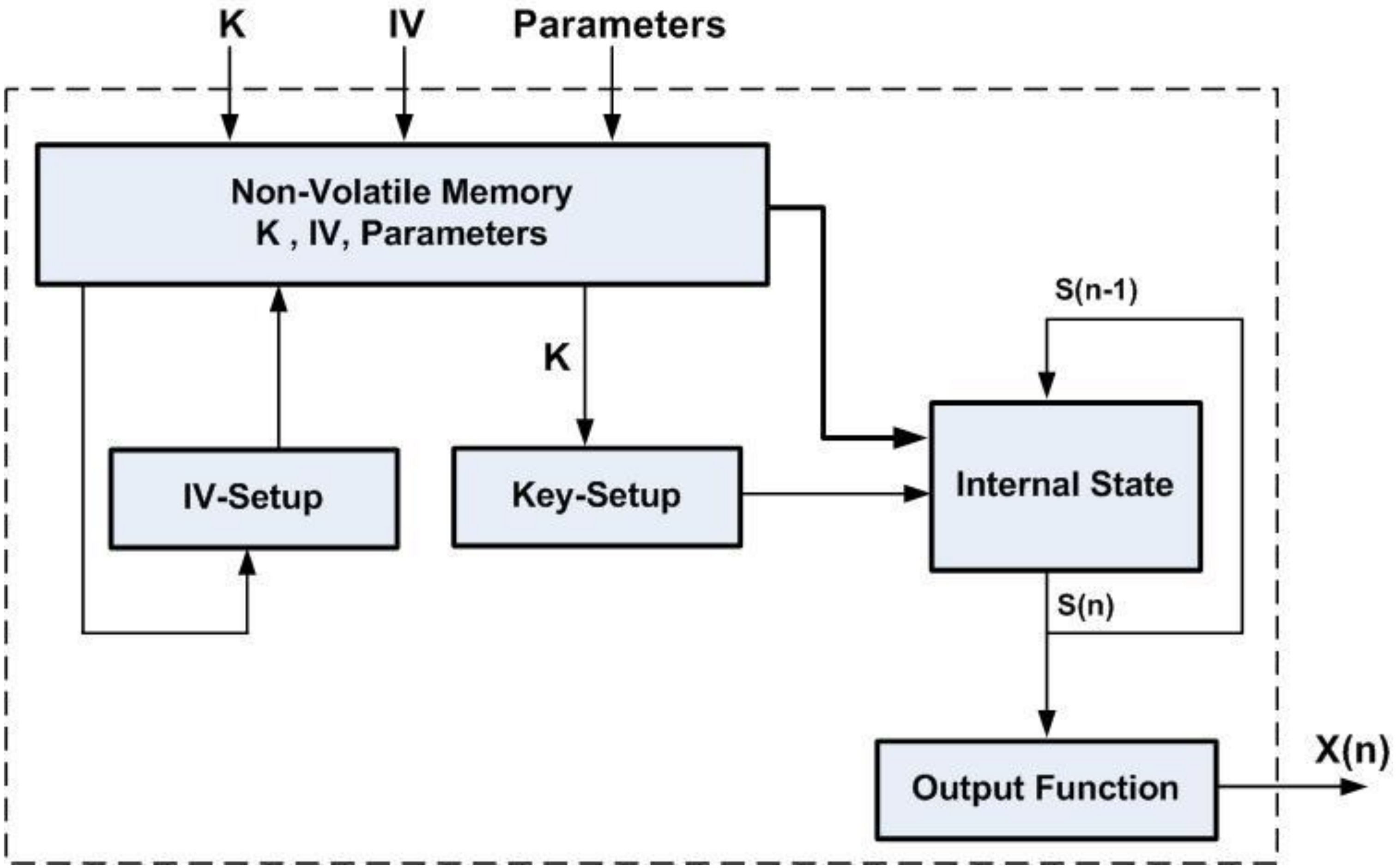
Encrypt $Pi = 1$*, depending on the keystream bit* $Xi = \begin{cases} 0 \\ 1 \end{cases}$ *gives* $Ci = \begin{cases} 1 \\ 0 \end{cases}$

The security of a stream cipher completely depends on the Keystream generator

How to avoid the effects of the finite precision N and to obtain randomness.

- Ultra-weak Coupling Technique & Chaotic mixing (Lozi, 2007 & 2012)
- Perturbation Technique (Tao, 2005, El Assad 2008)
- Recursive structure & Orbits Multiplexing (El Assad et. al., 2008 & 2011)
- Cascading Technique (Li et. al., 2001)

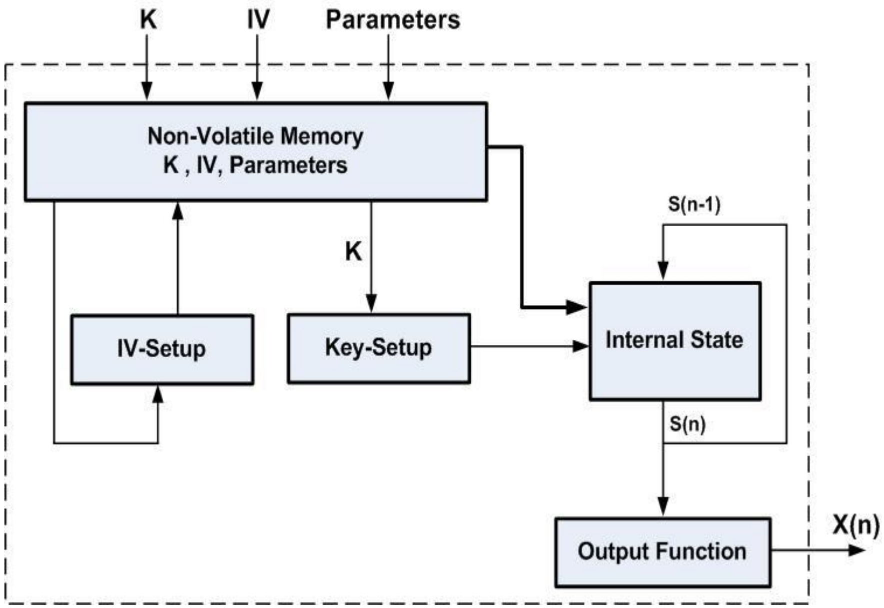
General structure of the proposed Pseudo Chaotic Number Generator (PCNG)



Keystream generator with internal feedback mode

The cryptographic complexity is in the internal state

Generation of the discrete chaotic samples: sequential calculus



$$S(n) = h\{Xs(n), Xp(n)\}$$

Step 1: Read the secret *K* (from a secured memory) and *IV* from the non-volatile memory

Step 2: Generation of the 1st sample: *n* = 1

$$X_{map}(0) = X_{map} + IV_{map}$$

$$S(1) = f\left[X_{map}(0), K\right], X(1) = g[S(1)]$$

or

$$Us = LSB_{32}(IV), Up = MSB_{32}(IV)$$

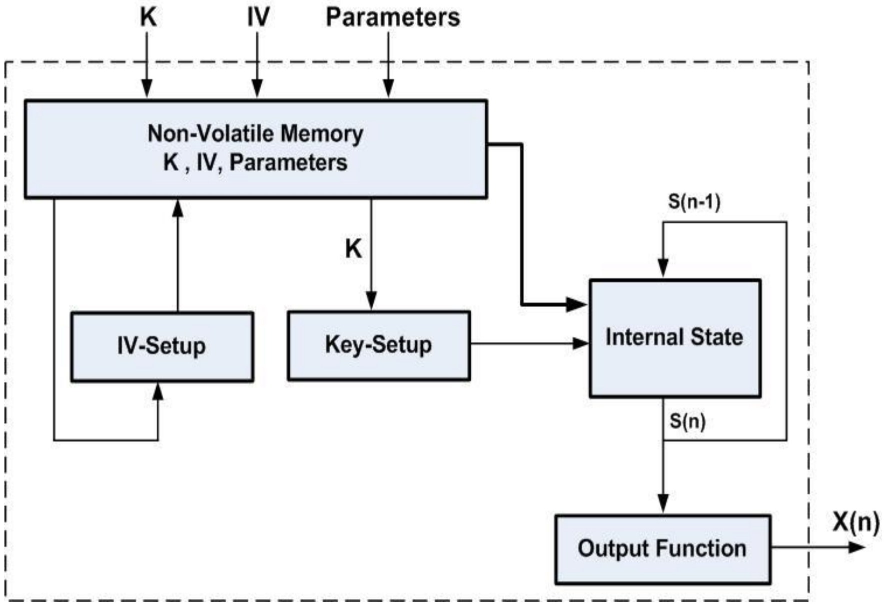
$$S(1) = f[IV, K], X(1) = g[S(1)]$$

Step 3: Generation of all samples: *n* = 2, …, *l_seq*

$$S(n) = f[S(n-1), K], X(n) = g[S(n)]$$

when (*n* = *l-seq*), then generation of a new *IV* using Linux generator "/dev/urandom", and the IV-Setup block, then save in the non-volatile memory and go to step 1 for a new execution of the program.

Generation of the discrete chaotic samples: parallel calculus

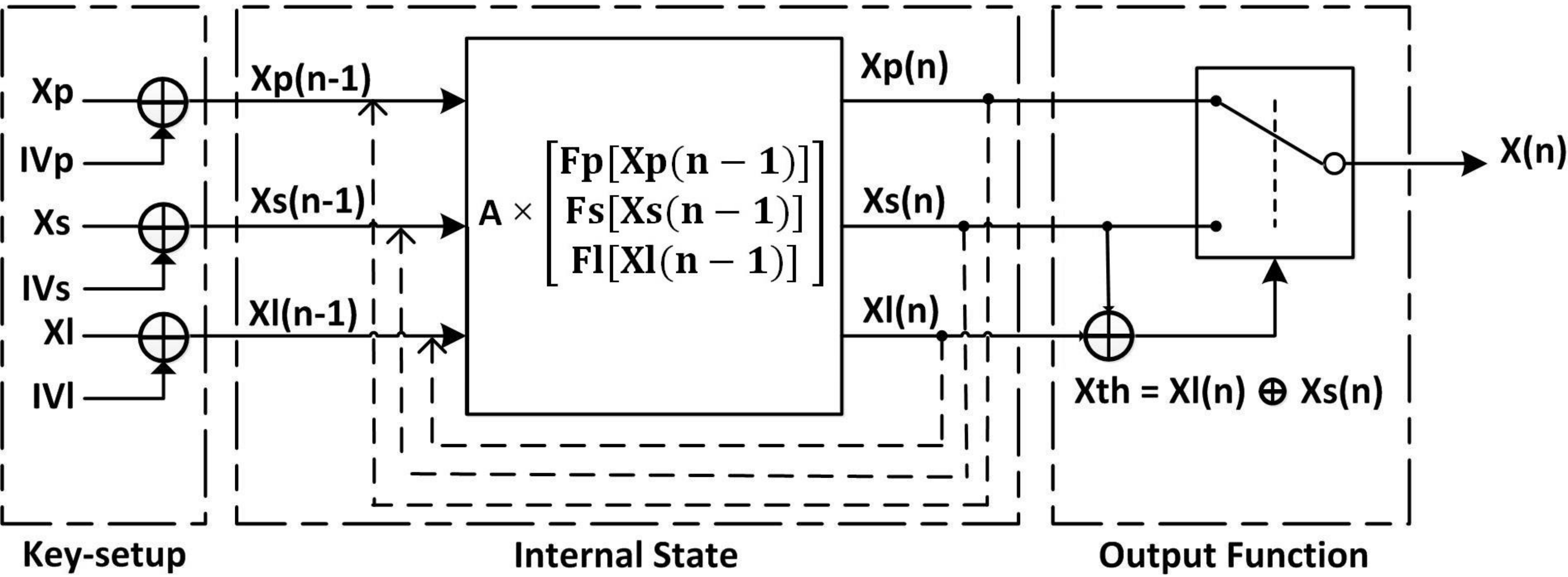


$$S(n) = h\{Xs(n), Xp(n)\}$$

Step 1: Read the secret *K* (from a secured memory) and *Iv* from the non-volatile memory. After that, calculus, using the K-Setup block, of *Nb-cores* (here *Nb-cores* = 4) secret keys, that differs each others by X_{map} and IV_{map} or by the parameters *K1s* and *K1p*. In the two cases, these parameters are obtained by a simple left circular shifting.

Step 2 & Step 3: Same calculus as previous by all cores in parallel, using the *P-thread* library. Each core calculates (*l-seq* / *Nb-cores*) samples.

Ultra-weak Coupling Technique and Chaotic mixing



$$\begin{bmatrix} X_p(n) \\ X_s(n) \\ X_l(n) \end{bmatrix} = \mathcal{A} \times \begin{bmatrix} F[X_p(n-1)] \\ F[X_s(n-1)] \\ F[X_l(n-1)] \end{bmatrix}; \mathcal{A} = \begin{bmatrix} (2^N - \varepsilon_{12} - \varepsilon_{13}) & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & (2^N - \varepsilon_{21} - \varepsilon_{23}) & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & (2^N - \varepsilon_{31} - \varepsilon_{32}) \end{bmatrix}$$

Where: $F[X_p(n-1)]$, $F[X_s(n-1)]$ and $F[X_l(n-1)]$ are the discrete chaotic maps PWLC, Skew Tent and Logistic respectively.

$$X(n) = \begin{cases} X_p(n), if\ 0 < Xth < T \\ X_s(n) \quad\quad otherwise \end{cases}$$

Ultra-weak Coupling Technique and Chaotic mixing

All the initial conditions, parameters and initialization vectors are chosen randomly from Linux generator: /dev/urandom.

The initial values:

$$\begin{cases} \boldsymbol{X_p(0) = X_p + IV_p} \\ \boldsymbol{X_s(0) = X_s + IV_s} \\ \boldsymbol{X_l(0) = X_l + IV_l} \end{cases}$$

$$|\boldsymbol{K}| = \{|\boldsymbol{X_p}| + |\boldsymbol{X_s}| + |\boldsymbol{X_l}|\} + \{|\boldsymbol{P_p}| + |\boldsymbol{P_s}|\} + \boldsymbol{6} \times |\boldsymbol{\varepsilon_{ij}}| = \boldsymbol{189\ bits}$$

Where:

$$|\boldsymbol{X_p}| = |\boldsymbol{X_s}| = |\boldsymbol{X_l}| = |\boldsymbol{P_s}| = \boldsymbol{32\ bits};\ \ |\boldsymbol{P_p}| = \boldsymbol{31\ bits}; |\boldsymbol{\varepsilon_{ij}}| = \boldsymbol{5\ bits}$$

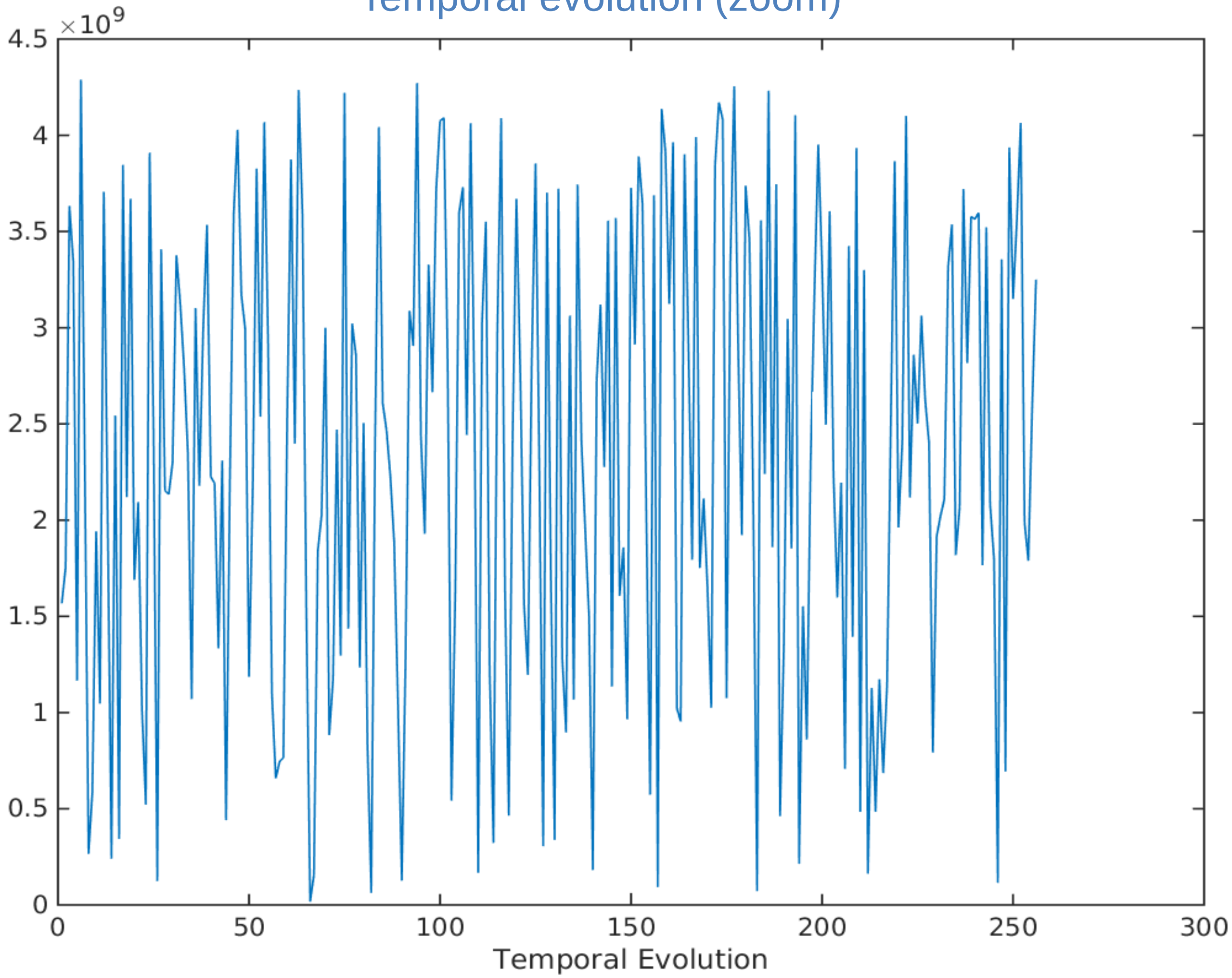
The key space is 2^{189}, it is large enough to make the brute-force attack infeasible

Ultra-weak Coupling Technique and Chaotic mixing

Robustness of the system against statistical attacks

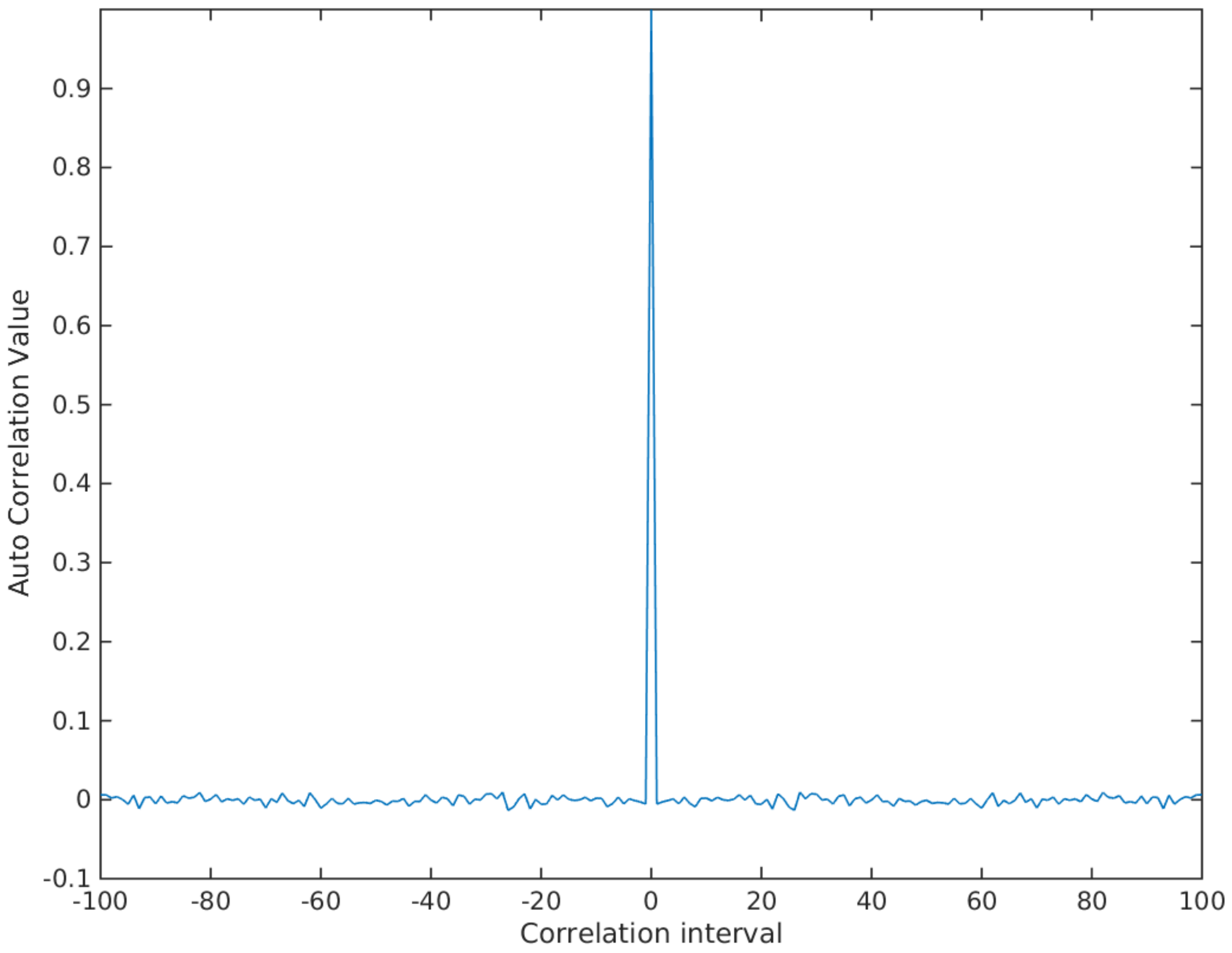
Passing statistical tests: Delta-like auto-correlation, nearly zero cross correlation, Pseudo-random mapping, Nist test, Uniformity of Histograms, Chi2 test

Temporal evolution (zoom)



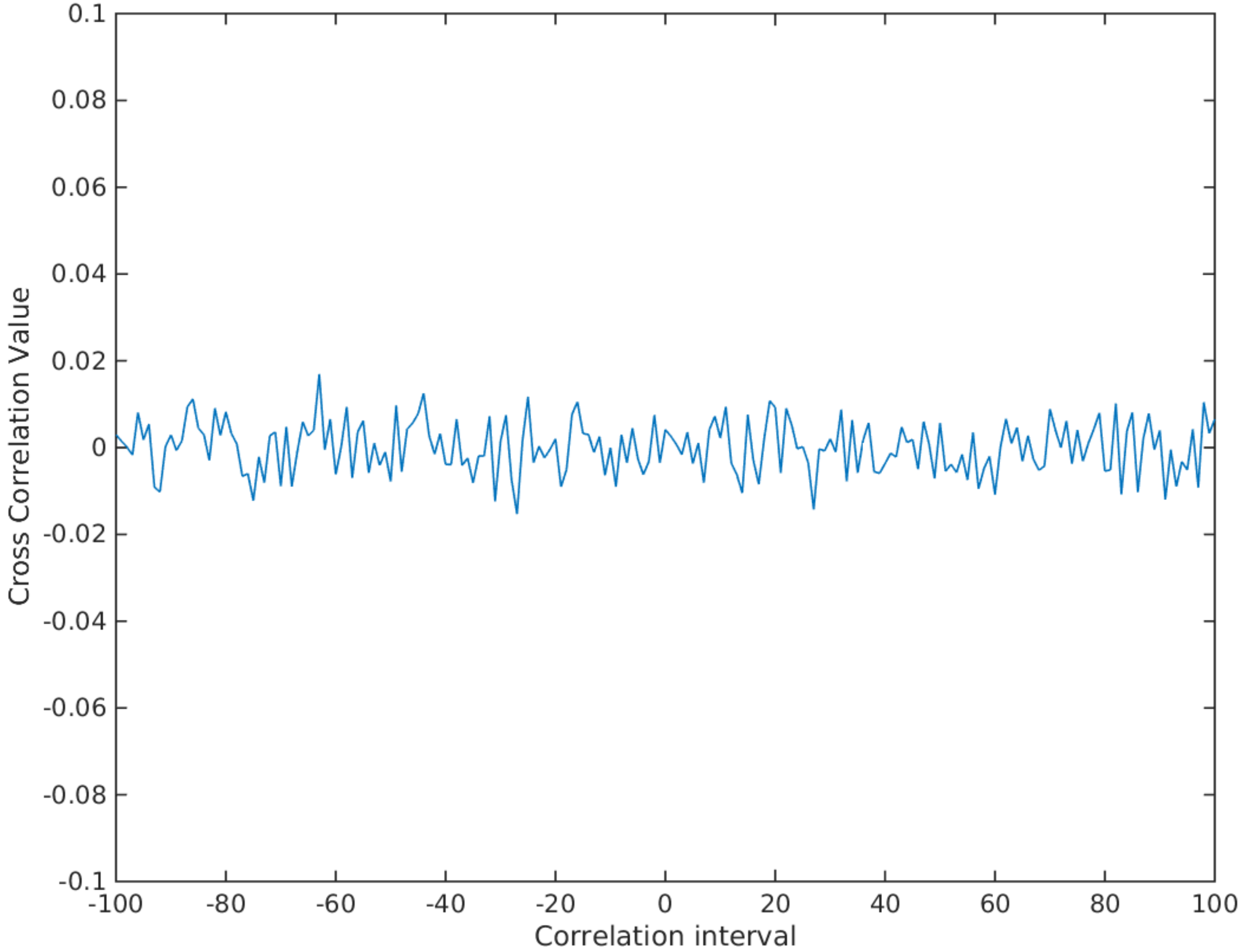
Ultra-weak Coupling Technique and Chaotic mixing: statistical performance

Auto-correlation (zoom)



Ultra-weak Coupling Technique and Chaotic mixing: statistical performance

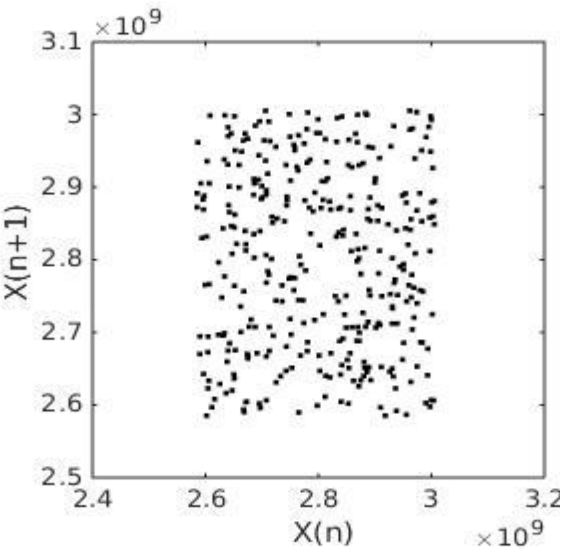
Cross-correlation (zoom)

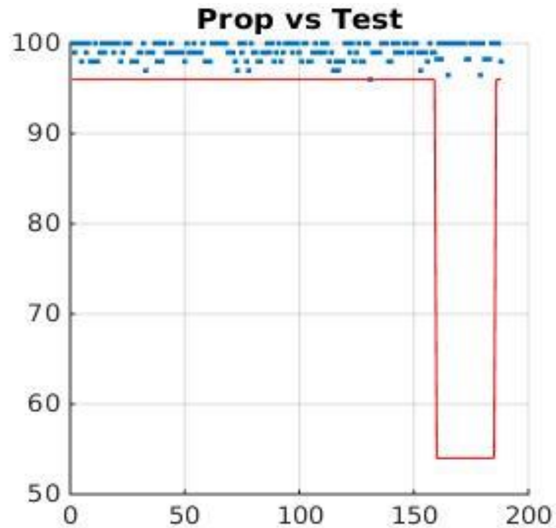


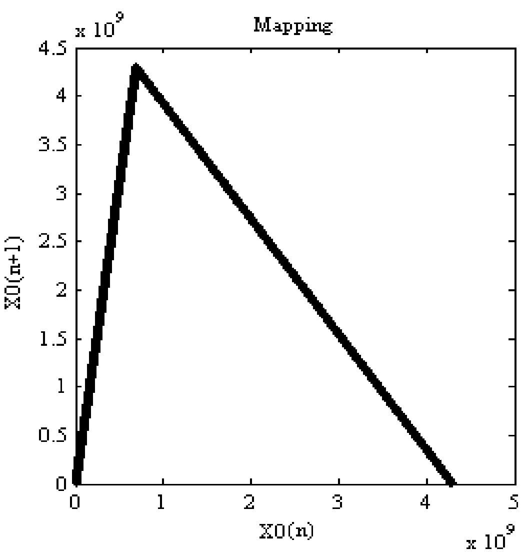
Ultra-weak Coupling Technique and Chaotic mixing: statistical performance

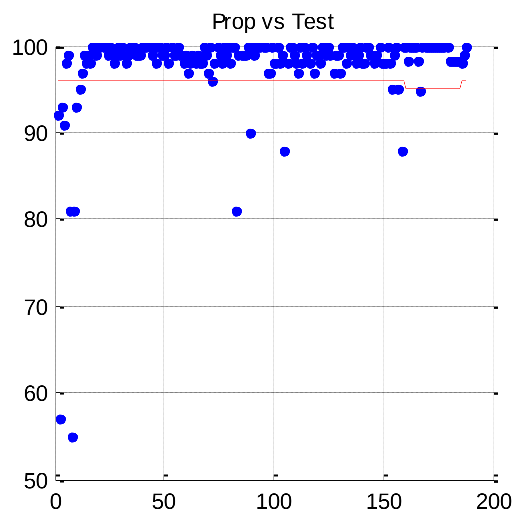
Mapping

Nist test









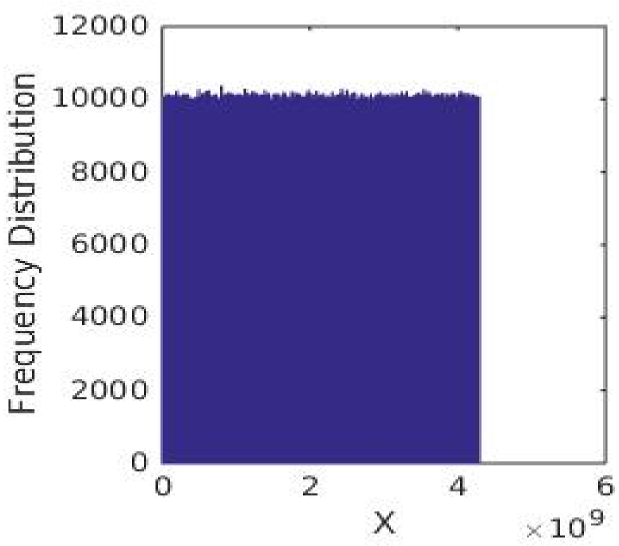
Skew Tent Map

Proposed system

Test	P-value	Prop
Frequency	0.946	100
Block-frequency	0.883	99
Cumulative-sums (2)	0.376	100
Runs	0.616	98
Longest-run	0.898	100
Rank	0.290	99
FFT	0.534	100
Non-periodic-templates (148)	0.483	99.06
Overlapping-templates	0.063	100
Universal	0.172	99
Approximate Entropy	0.419	99
Random-excursions (8)	0.335	99.12
Random-excursions-variant (18)	0.436	99.32
Serial (2)	0.478	100
Linear-complexity	0.249	98

Ultra-weak Coupling Technique and Chaotic mixing: statistical performance

Histogram



Uniformity $\Leftrightarrow \chi^2_{ex} < \chi^2_{th}(N_c - 1, \alpha)$

χ^2_{ex}	**1012.82**
χ^2_{th}	**1073.64** For $\alpha = 0.05$ and $N_c = 1000$

$$\chi^2_{ex} = \sum_{i=0}^{Nc-1} \frac{(O_i - E_i)^2}{E_i}$$

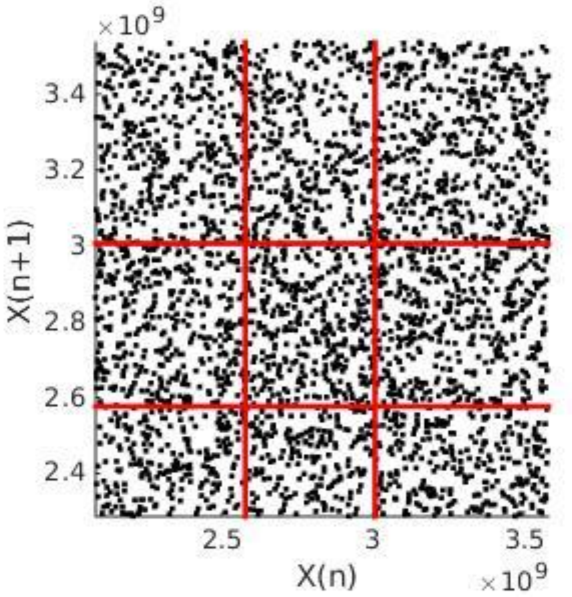
$N_c = 1000$: number of classes (sub − intervals)

O_i: number of observed (calculated) samples in the ith class E_i

$E_i = 10^7/N_c$: expected number of samples of a uniform distribution

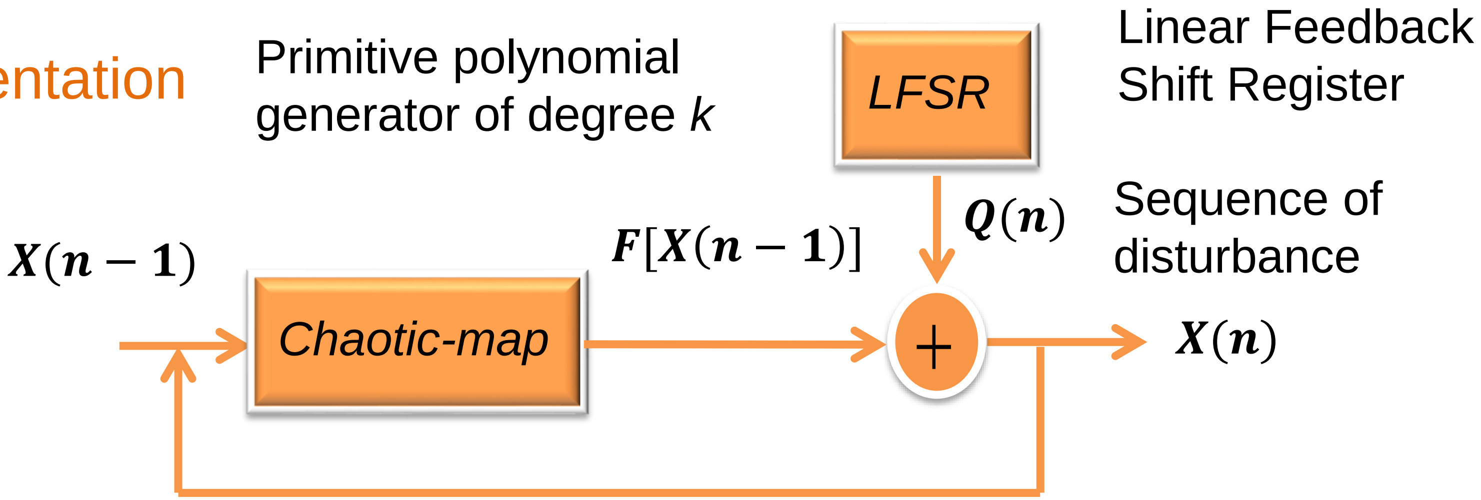
Ultra-weak Coupling Technique and Chaotic mixing: statistical performance

Approximated probability distribution function



Perturbation Technique

Implementation



$$X(n) = x_{N-1}(n)x_{N-2}(n)\cdots x_i(n)\cdots x_1(n)x_0(n) \qquad x_i(n) \in A_b = [0, 1]$$

Perturbation every Δ iterations Δ : Average orbit of the chaotic-map without perturbation

$$If \quad n = l \times \Delta \qquad l = 1, 2, \cdots$$

$$x_i(n) = \begin{cases} F[x_i(n-1)] & k \le i \le N-1 \\ F[x_i(n-1)] \oplus q_i(n) & 0 \le i \le k-1 \end{cases}$$

$$Else$$

$$No\ perturbation: X(n) = F[X(n-1)]$$

Lower length of the orbit: $o_{min} = \Delta \times (2^k - 1)$

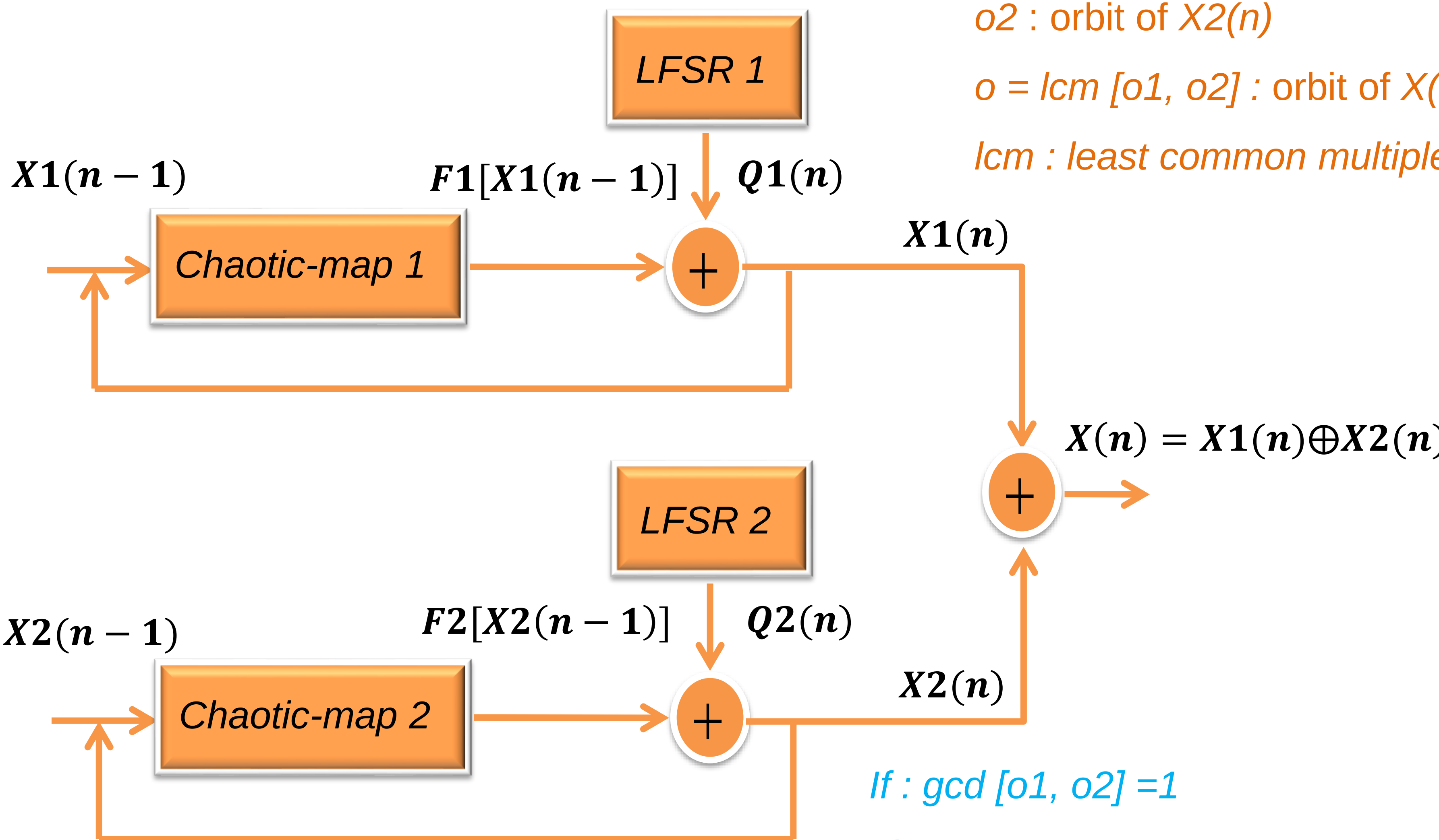
Cascading Technique

o1 : orbit of *X1(n)*

o2 : orbit of *X2(n)*

o = lcm [o1, o2] : orbit of *X(n)*

lcm : least common multiple

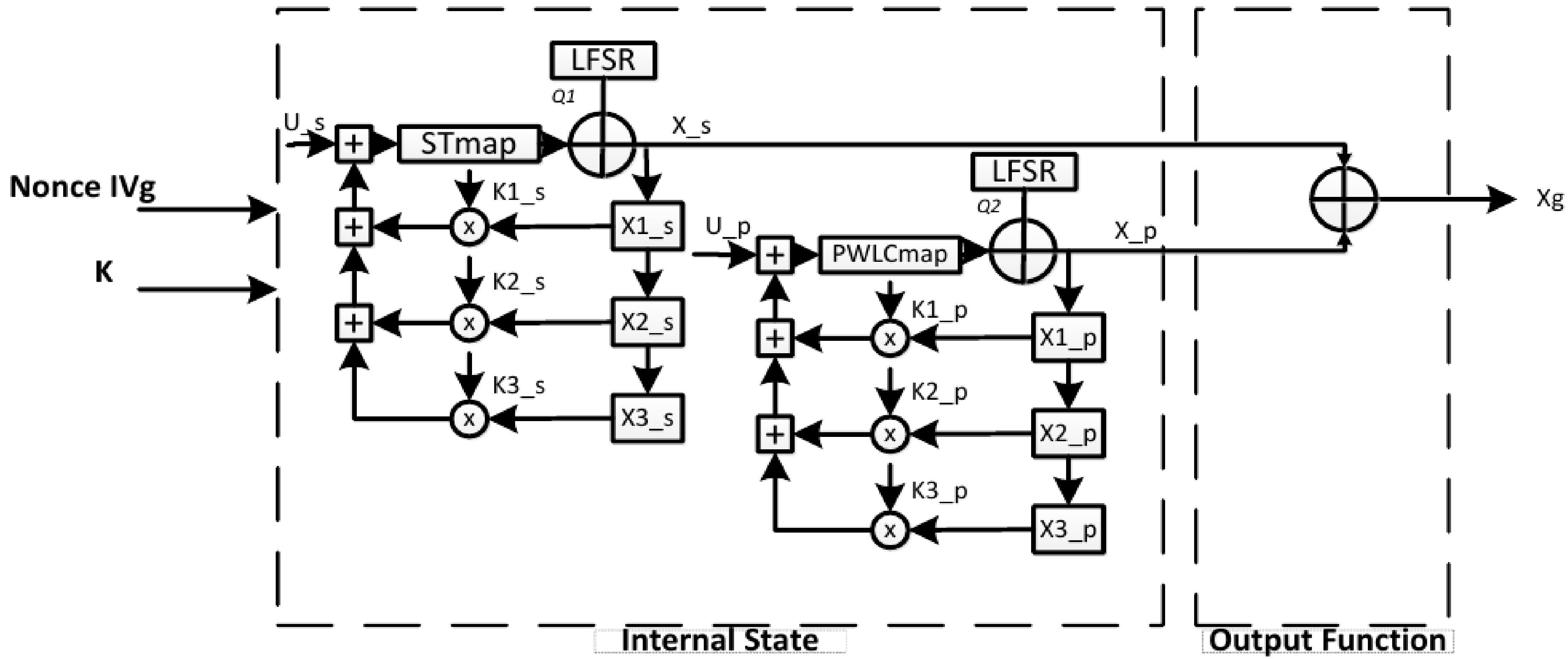


If : gcd [o1, o2] =1

Then : *o = o1 x o2*

gcd : greatest common divisor

Basic chaotic generator: Patent 2011



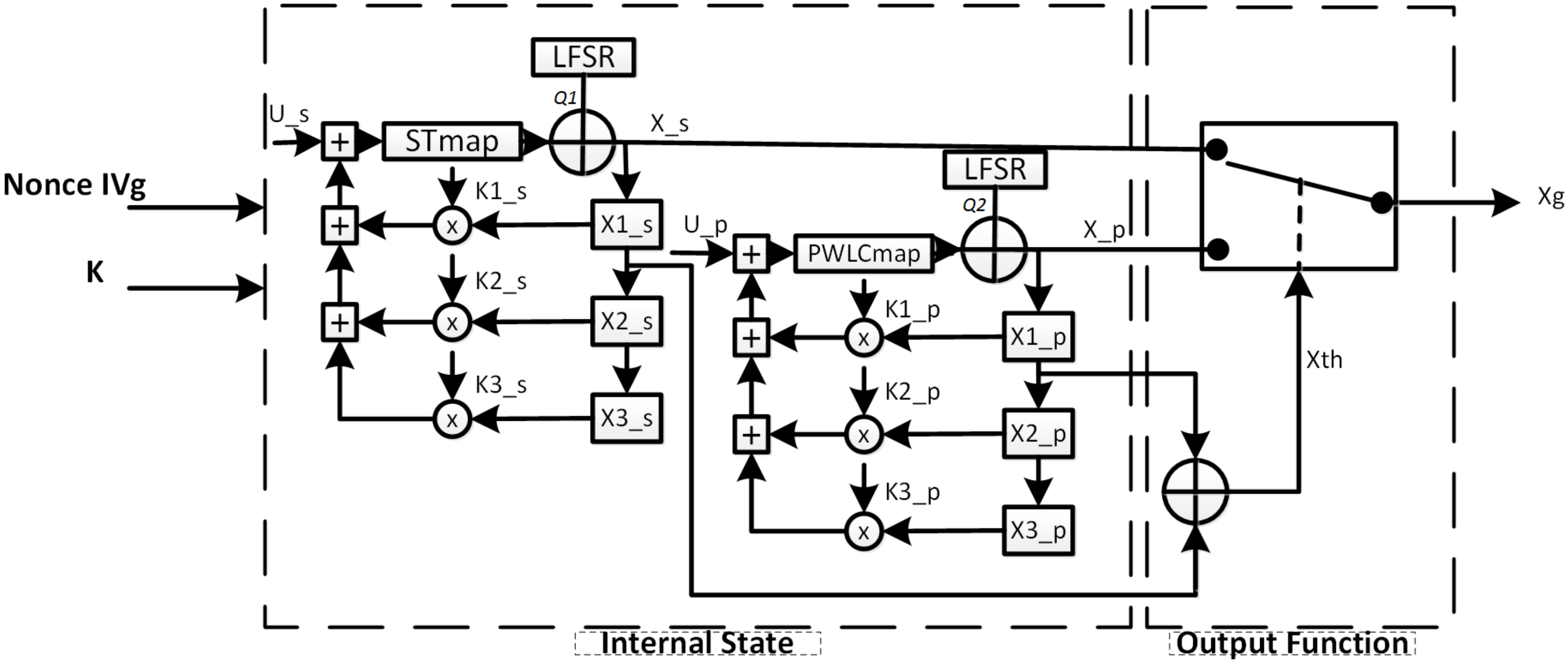
$$Xs(n) = Skew\_Tent\left\{mod\left[\left(Us + Xs(0) + \sum_{i=1}^{j} Kis \times Xs(n-i)\right), 2^N\right], Ps\right\} \oplus Q1(n)$$

$1 \leq j \leq 3$

$$Xp(n) = PWLC\left\{mod\left[\left(Up + Xp(0) + \sum_{i=1}^{j} Kip \times Xp(n-i)\right), 2^N\right], Pp\right\} \oplus Q2(n)$$

$$X_g(n) = Xs(n) \oplus Xp(n)$$

Basic chaotic generator: Patent 2011 PhD Student: Mohammad Abu Taha



$$Xs(n) = Skew\_Tent\left\{mod\left[\left(Us + Xs(0) + \sum_{i=1}^{j} Kis \times Xs(n-i)\right), 2^N\right], Ps\right\} \oplus Q1(n)$$

$$Xp(n) = PWLC\left\{mod\left[\left(Up + Xp(0) + \sum_{i=1}^{j} Kip \times Xp(n-i)\right), 2^N\right], Pp\right\} \oplus Q2(n)$$

$1 \leq j \leq 3$

$$Xg(n) = \begin{cases} Xp(n)\ if\, X_{Th}(n) \leq 2^{N-1} \\ Xs(n) \qquad\qquad othewise \end{cases} \qquad X_{Th}(n) = Xs(n-1) \oplus Xp(n-1)$$

Basic chaotic generator : Advantages

- **Generic scheme**
- **Long orbit of *Xg(n)*:** $o_{min} = lcm[\Delta_s \times (2^{k1} - 1), \Delta_p \times (2^{k2} - 1)]$

With: $N = 32,\ k1 = 21, k2 = 23\ and\ \Delta_{nom} \cong 2^{\frac{N}{2} \times 3} = 2^{48} \Longrightarrow 2^{71} \leq o_{min} \leq 2^{140}$

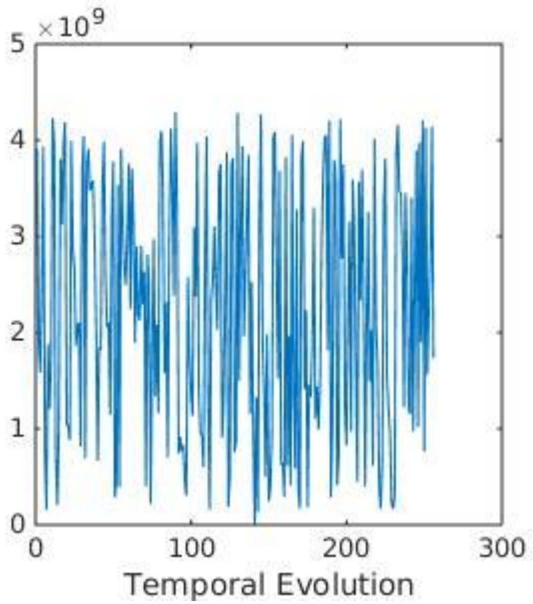
- **Large secret key space:** Brute-Force Attack infeasible

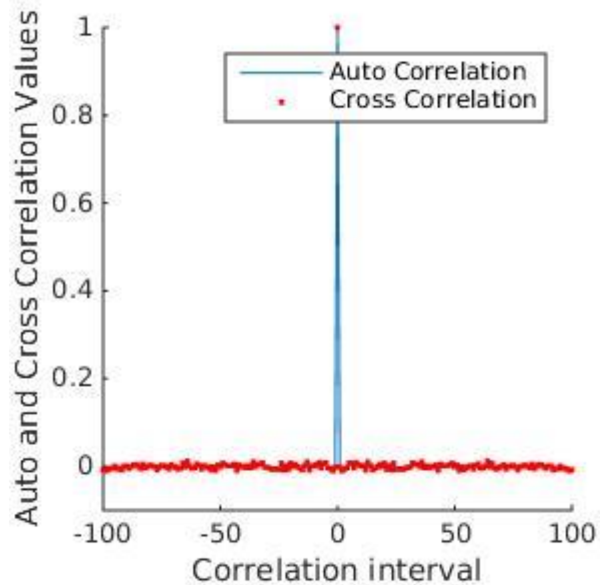
Delay d	Key size (bits) of the Skew-tent recursive cell [Nic + Np] x N + k1	Key size (bits) of the PWLCM recursive cell [Nic + (Np-1)] x N + N - 1 + k2	Key size (bits)
3	4N + 4N + k1 = 256 + 21 = 277	4N + 3N + (N-1) + k2 = 255 + 23 = 278	555
2	3N + 3N + k1 = 192 + 21 = 213	3N + 2N + (N-1) + k2 = 191 + 23 = 214	427
1	2N +2N + k1 = 128 + 21 = 149	2N + N + (N-1) + k2 = 127 + 23 = 150	299

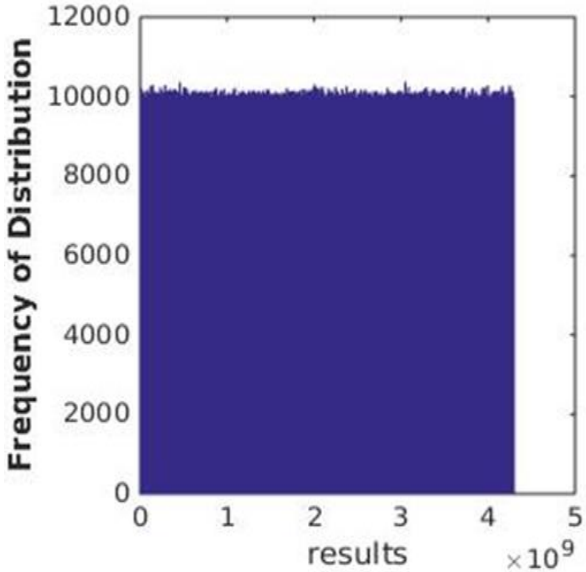
Speed of a Brute-Force Attack: (Nb of keys to be tested and the speed of each test)

With **key size = 256 bits**, there are **2^{256} possible keys.** Assuming a computer can try a **million keys a second**, it will take **[2^{256} / (10^6 x 3600 x 24 x 356)] > 3 x10^{63} years old**, a very long time, because **the universe is only 10^{10} years old**.

Basic chaotic generator: Correlation (zoom), Histogram, Chi2







	Delay = 1	Delay = 2	Delay = 3
χ^2_{ex}	**1022.96**	**990.13**	**860.35**
χ^2_{th}	**1073.64** For $\alpha = 0.05$ and $N_c = 1000$		

Basic chaotic generator: Nist test

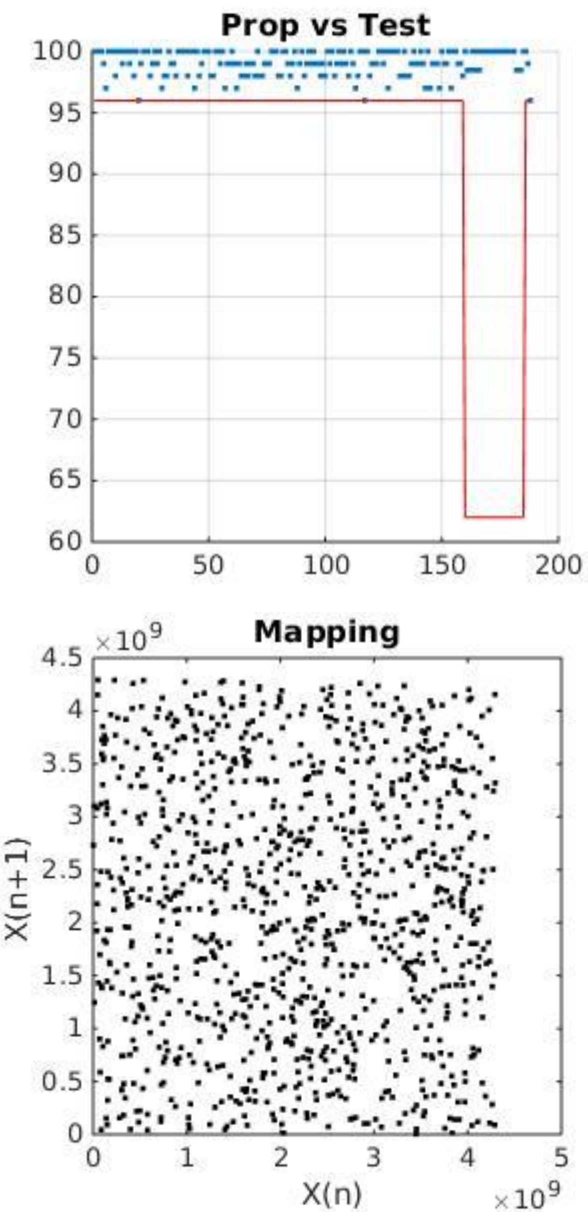
	Delay = 1		Delay = 2		Delay = 3	
Test	**P-value**	**Prop**	**P-value**	**Prop**	**P-value**	**Prop**
Frequency	0.081	100	0.616	99	0.699	100
Block-frequency	0.616	100	0.475	100	0.237	98
Cumulative-sums (2)	0.790	100	0.527	98.5	0.373	100
Runs	0.494	99	0.868	99	0.130	100
Longest-run	0.350	97	0.367	99	0.534	100
Rank	0.658	100	0.699	98	0.924	100
FFT	0.213	100	0.575	100	0.834	99
Non-periodic-templates (148)	0.514	99.01	0.531	99	0.498	98.94
Overlapping-templates	0.575	99	0.72	100	0.004	99
Universal	0.898	99	0.851	100	0.596	97
Approximate Entropy	0.437	98	0.699	96	0.834	100
Random-excursions (8)	0.418	99.24	0.489	98.83	0.399	99.77
Random-excursions-variant (18)	0.364	99.75	0.323	98.44	0.552	99.79
Serial (2)	0.395	99.5	0.535	99.5	0.269	99
Linear-complexity	0.081	96	0.304	96	0.991	100

The minimum pass rate for each statistical test with the exception of the random excursion (variant) test is approximately = 96.00 for a sample size = 100 binary sequences.

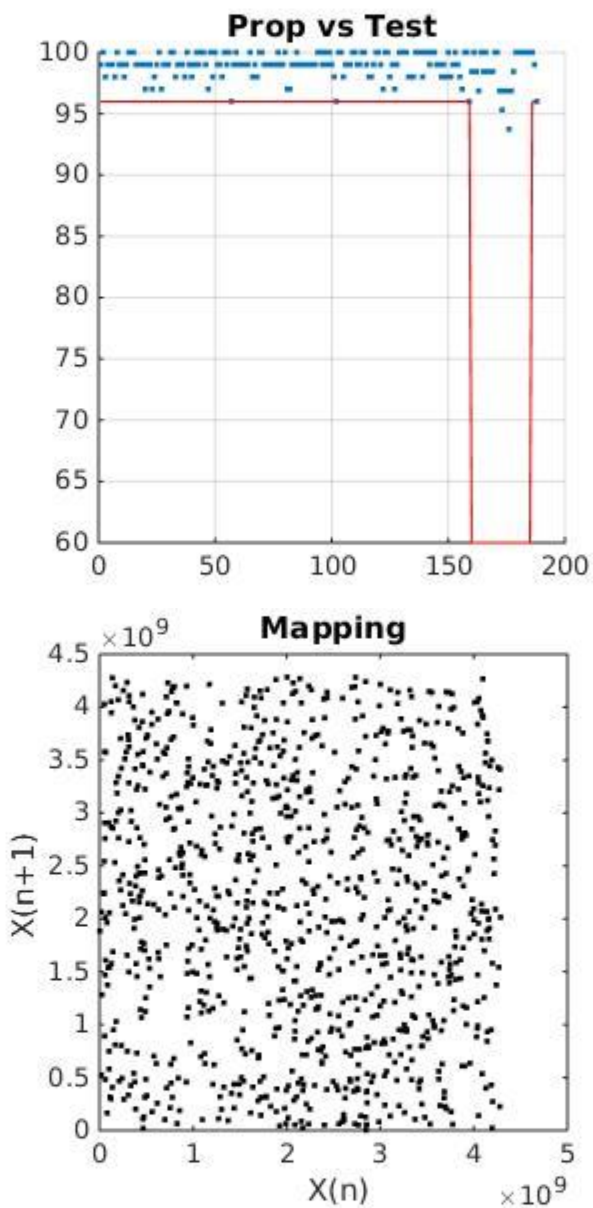
The minimum pass rate for the random excursion (variant) test is approximately= 62.00 for a sample size = 66 binary sequences

Basic chaotic generator: Nist test and Mapping (zoom)

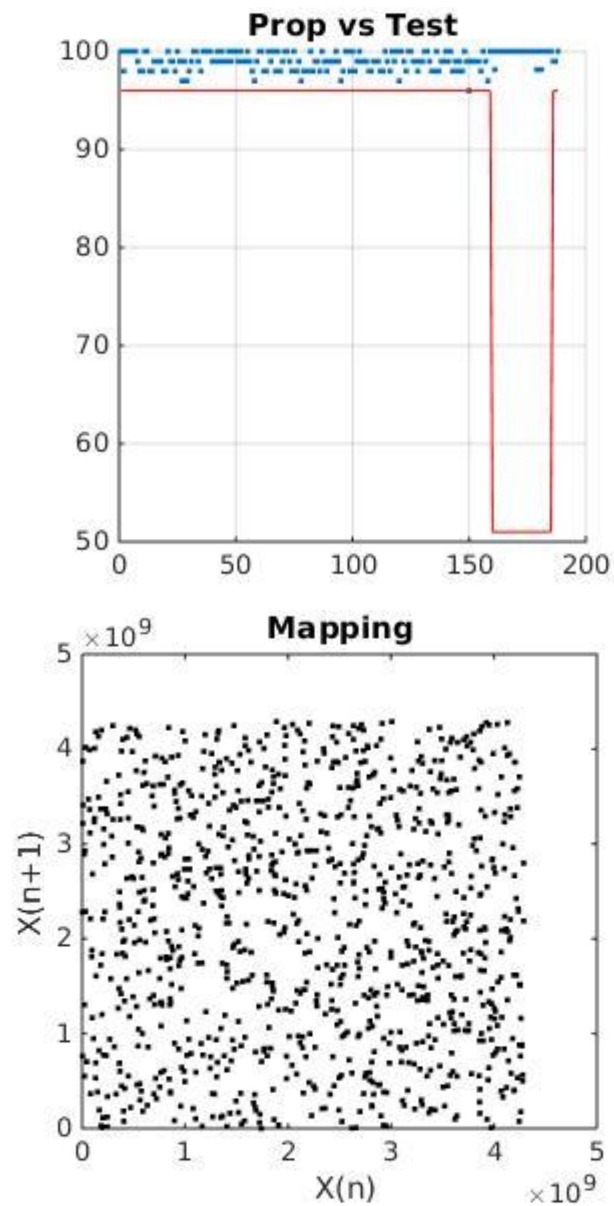
Delay = 1



Delay = 2



Delay = 3



Structure of the chaotic generator

Generator of chaotic Sequences

and corresponding generating

system WO Patent

WO/2011/121,218 A1, Oct 6, 2011

PCT Extension:

United States

US-8781116 B2, July 15, 2014.

Europe

EP-2553567 B1, Sept 3, 2014.

China :

CN-103124955 B, April 20, 2016.

2958057

Truly unbreakable cipher: One-Time Pad

3/4

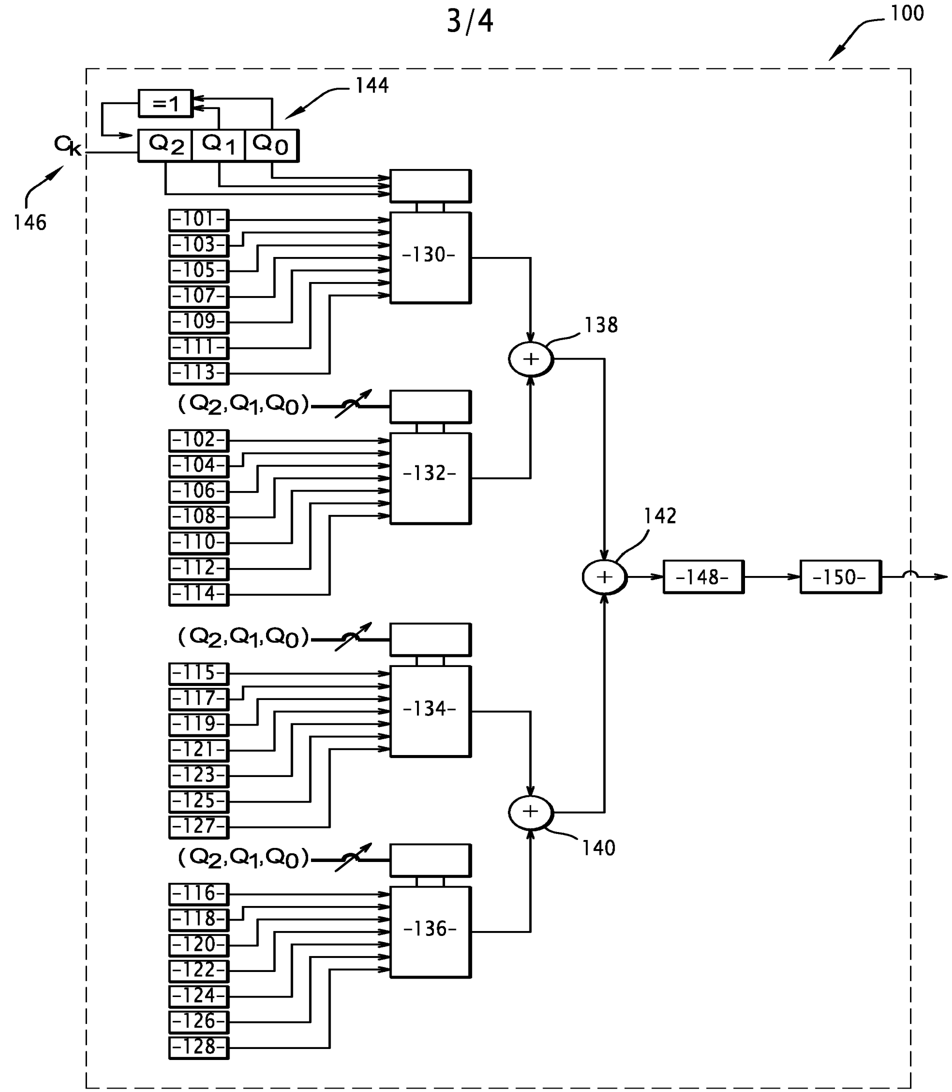


FIG.3

For each state *j = 1,2,…,7 of the LFSR*

Point 142 : $$o_{j\min}{}_{j=1,2,\cdots,7} = lcm\left[o_{j\min 1}, o_{j\min 2}\right]$$

Point 138 : $$o_{j\min 1}{}_{j=1,2,\cdots,7} = lcm\left\{\left[2^{k_{(2j-1)}}-1\right]\times\Delta_{k_{(2j-1)}}, \left[2^{k_{(2j)}}-1\right]\times\Delta_{k_{(2j)}}\right\}$$

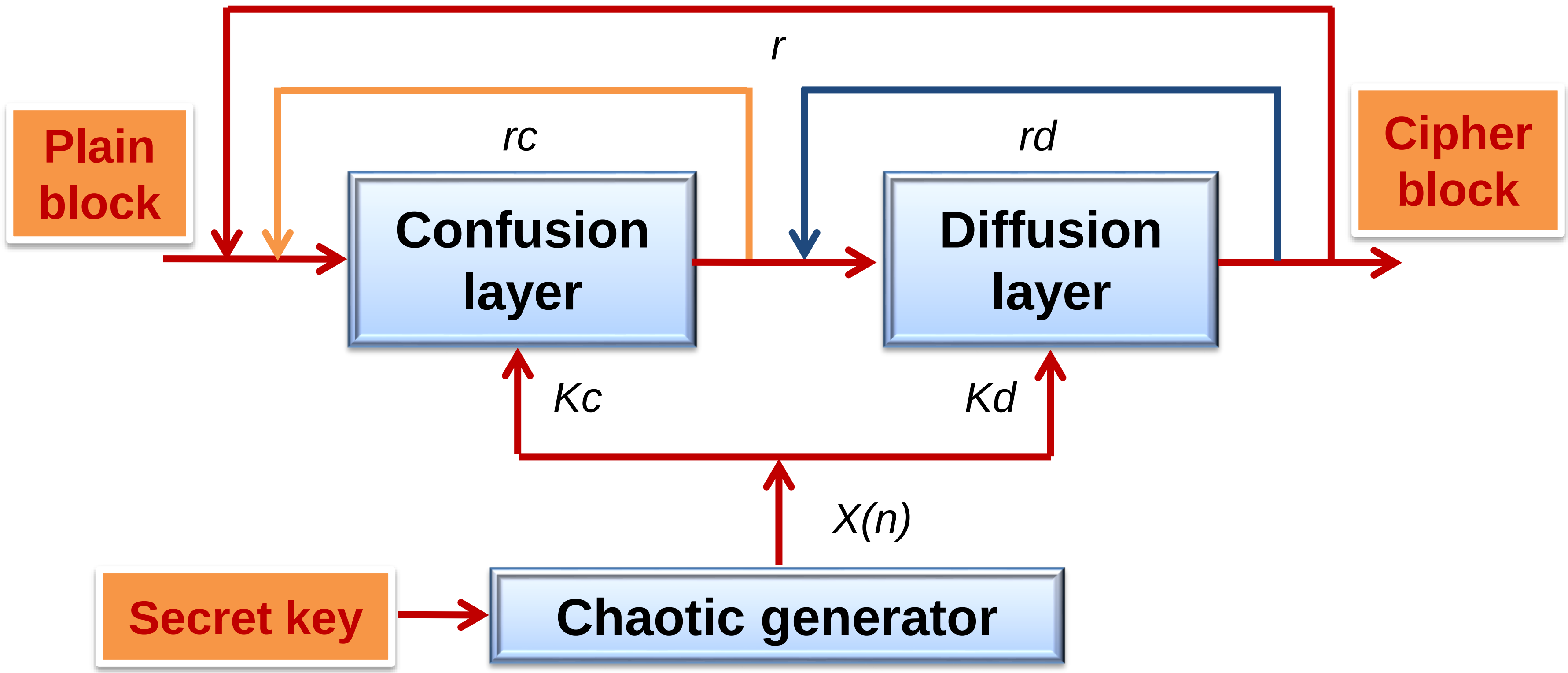
Point 140 :

$$o_{j\min 2}{}_{j=1,2,\cdots,7} = lcm\left\{\left[2^{k_{(14+2j-1)}}-1\right]\times\Delta_{k_{(14+2j-1)}}, \left[2^{k_{(14+2j)}}-1\right]\times\Delta_{k_{(14+2j)}}\right\}$$

$$T_{Ck} = Min\left(o_{j\min}{}_{j=1,2,\cdots,7}\right)$$

$$o_{\min} = 7\times T_{Ck}\left[1-p\%\right]$$

General structure of chaos-based cryptosystems: Encryption side



Shannon [1949]

Confusion : measures how a change in the secret key affects the ciphered massage

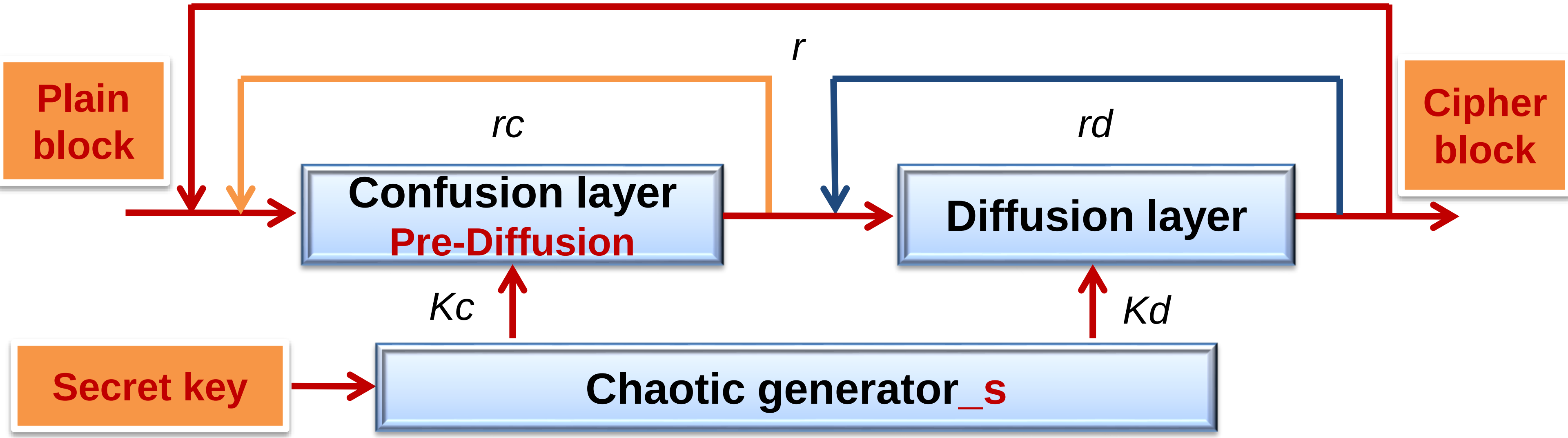
Diffusion : assesses how a change in the plain message affects the ciphered one

Fridrich [1998]:

Most popular structure adopted in many chaos-based cryptosystems

Chaos-based cryptosystems: two types

- 1er type : Separate layers of confusion and diffusion



Both layers required image-scanning to obtain ciphered image

Confusion layer:

- **Pixel 2D-Permutation (Cat map; Standard map; Baker map)**

The image pixels are relocated without changed their values, an operation of **Substitution.**

- **Pixel 1-D Substitution (Finite state Skew tent map: a non linear function)**

The image pixel values are substituted without or with Key-dependent on each round

Diffusion Layer:

1-D diffusion (Discrete Logistic map, Discrete Skew tent map)

Logistic map as diffusion layer

$$\begin{cases} c(i) = v(i) \oplus q\left\{ f\left[c(i-1)\right], L \right\} \\ c(-1) = Kd, \quad L = 8 \end{cases}$$

$$\begin{cases} f\left[c(i-1)\right] = 4 \times c(i-1) \times \left[1 - c(i-1)\right] \\ q\left[b, L\right] = \left\lfloor b \times 2^{L} \right\rfloor, \quad b = 0.b_1 b_2 \cdots b_L, \;\; bj \; is \; 0 \; or \; 1 \end{cases}$$

v_i is the value of the *i*th pixel of the permuted image

c_{i-1} and c_i are the values of the (*i-1*)th and *i*th pixels of the diffused image

Kd is the diffusion key

Chaos-based cryptosystems: 1er type

Pre-Diffusion included in the confusion layer:

XOR or Add: after relocated

$$v(i) = Mod\left\{\left[v(i) \oplus v(i-1)\right], Q)\right\}$$

Add-and-Shift: before relocated

$$\begin{cases} v(i) = Cyc\left[Mod\left[(p(i)+v(i-1)), Q\right], LSB_3(v(i-1))\right] \\ v(-1) = Kc \in \left[1, (Q-1)\right], \quad Q = 2^8 = 256 \end{cases}$$

p(i) is the current value of the plain image, *v*(*i*-1) is the value of the (*i*-1)th pixel after permutation, *Cyc* [*s*, *z*] performs the *z*-bit right cyclic shift on the binary sequence *s*, and *v*(*i*) is the resultant pixel value in the permuted image.

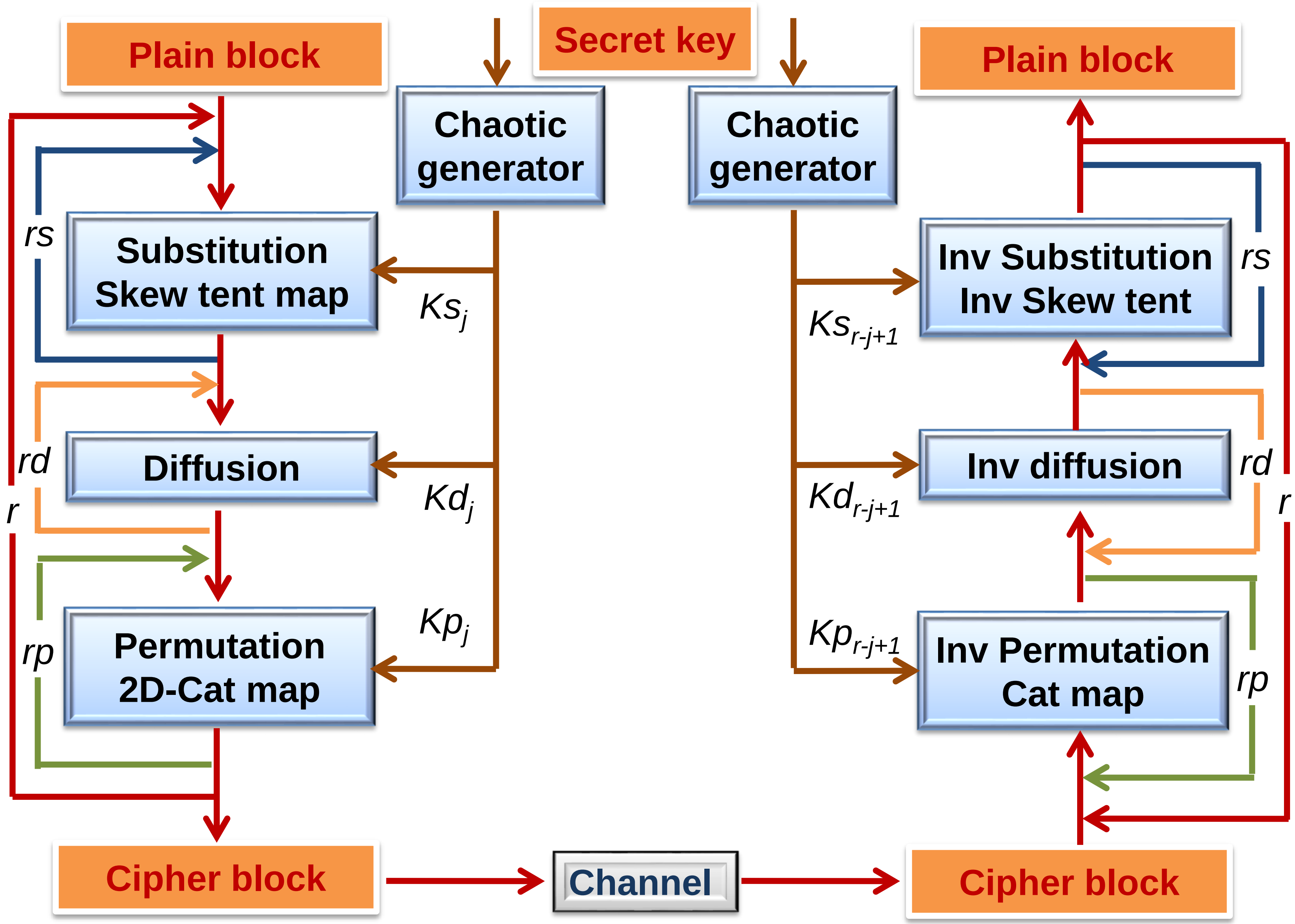
Chaotic generator_s of dynamic keys (encryption keys):

Logistic, Skew tent, PWLCM, Lorenz, basic generator, combined maps

[Fridrich , 1998], [Chen et al., 2004], [Lian et al., 2005a], [Lian et al., 2005b], [Wong et al., 2008], Masuda et al., 2006], [Farajallah et al., 2013], [Wang et al., 2009], [El Assad et al., 2008], [Caragata et al., 2014], El Assad et al., 2014].

Proposed chaos-based cryptosystem (GreenCom 2013), [Farajallah et al.]

- Cryptosystem based on variable control keys



Equations of the Skew tent map and inverse Skew tent maps

Finite state Skew tent map as substitution layer :

Robust nonlinear layer, resists to the chosen cipher text attack

$$Y = S_a(X) = \begin{cases} \left\lceil \frac{Q}{a} X \right\rceil & 0 \le X \le a \\ \left\lfloor \frac{Q}{Q-a}(Q-X) \right\rfloor + 1 & a < X < Q \end{cases}$$

Structure of the dynamic key *Ks*

$$Ks = \left[Ks_1 \| Ks_2 \| \cdots Ks_r \right]$$

$$Ks_j = \left[a_{j,1} \| a_{j,2} \| \cdots a_{j,rs} \right], \ j = 1, \cdots, r$$

$$1 \le a_{j,i} < Q = 2^8$$

Inverse Skew tent map

$$X = S_a^{-1}(Y) = \begin{cases} \xi 1 \ \ if \ \ \theta(Y) = Y \ and \ \frac{\xi 1}{a} > \frac{Q - \xi 2}{Q-a} \\ \xi 2 \ \ if \ \ \theta(Y) = Y \ and \ \frac{\xi 1}{a} \le \frac{Q - \xi 2}{Q-a} \\ \xi 1 \ \ if \ \ \theta(Y) = Y + 1 \end{cases}$$

$$\xi 1 = \left\lfloor \frac{a}{Q} Y \right\rfloor, \xi 2 = \left\lceil \left\lceil \frac{a}{Q} - 1 \right\rceil Y + Q \right\rceil$$

$$\xi 3 = \left\lceil \frac{a}{Q} Y \right\rceil, \theta(Y) = Y + \xi 1 - \xi 3 + 1$$

One to one mapping: Implemented by lookup tables

Key generator : A simplified version of the basic chaotic generator of our Patent

Chaos-based cryptosystems: 1er type

2-D Cat map as permutation layer

$$\begin{bmatrix} i_n \\ j_n \end{bmatrix} = Mod\left(\begin{bmatrix} 1 & u \\ v & 1+uv \end{bmatrix} \times \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} ri+rj \\ rj \end{bmatrix}, \begin{bmatrix} M \\ M \end{bmatrix} \right) \quad 0 \le u, v, ri, rj \le M-1 = 2^q - 1$$

Structure of the dynamic key *Kp*

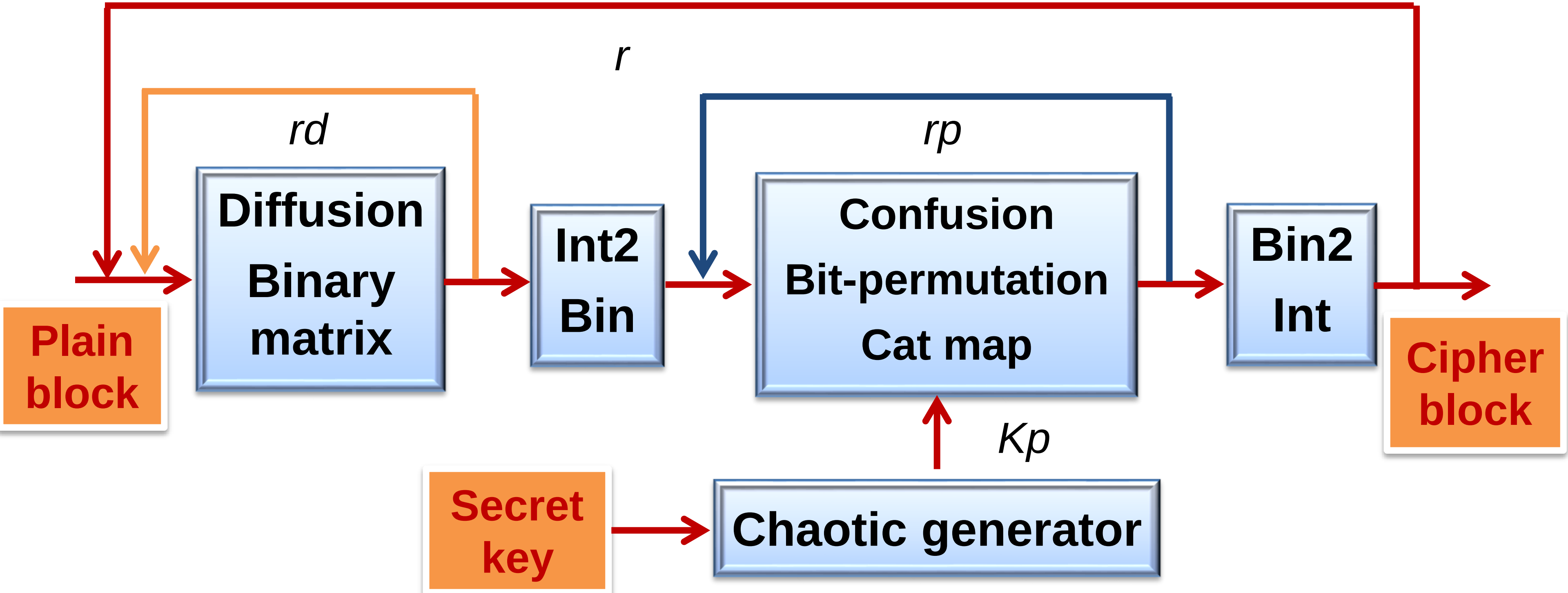
$$Kp = \left[kp_1 \| kp_2 \| \cdots \| kp_r \right]$$

$$kp_l = \left[u_l, v_l, ri_l, rj_l \right] \quad l = 1, \cdots, rp$$

Where i, j and i_n, j_n are the original and permuted pixel positions of the *M X M square* matrix, with $M = 2^q$.

The Cat map is bijective, so each point in the square matrix is transformed to another point uniquely.

Chaos-based cryptosystems: 1er type



$$\begin{bmatrix} O_{d_0} \\ O_{d_1} \\ \vdots \\ \vdots \\ O_{d_{31}} \end{bmatrix} = [\boldsymbol{DM}] \odot \begin{bmatrix} O_0 \\ O_1 \\ \vdots \\ \vdots \\ O_{31} \end{bmatrix}$$

Int2BIN: Nonlinear converter

2D cat : Efficient formulation for C implementation

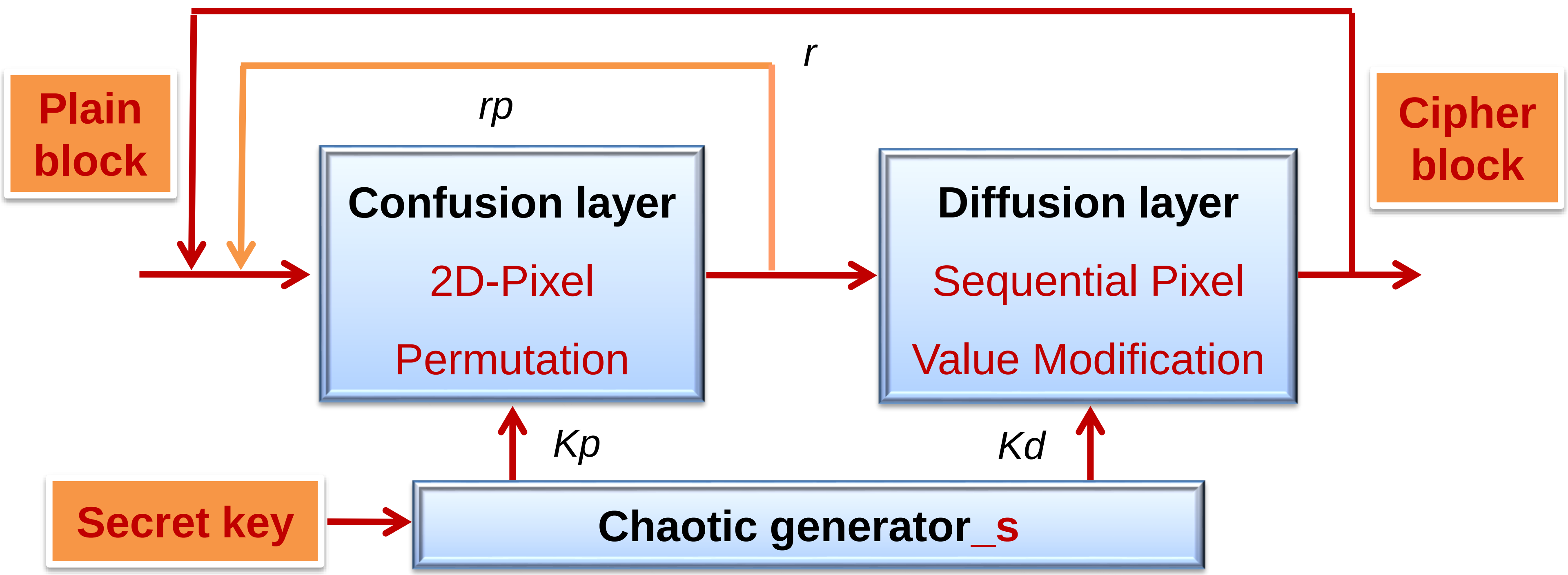
When a bit permutation layer is applied on a block , it performs, on one scan, a substitution and a diffusion operations on the bytes

[El Assad & Farajallah, 2016., in Signal Processing: Image Communication]

Koo et. al., 2006

Chaos-based cryptosystems: 2nd type

- **2nd type : Combined layers of confusion and diffusion**

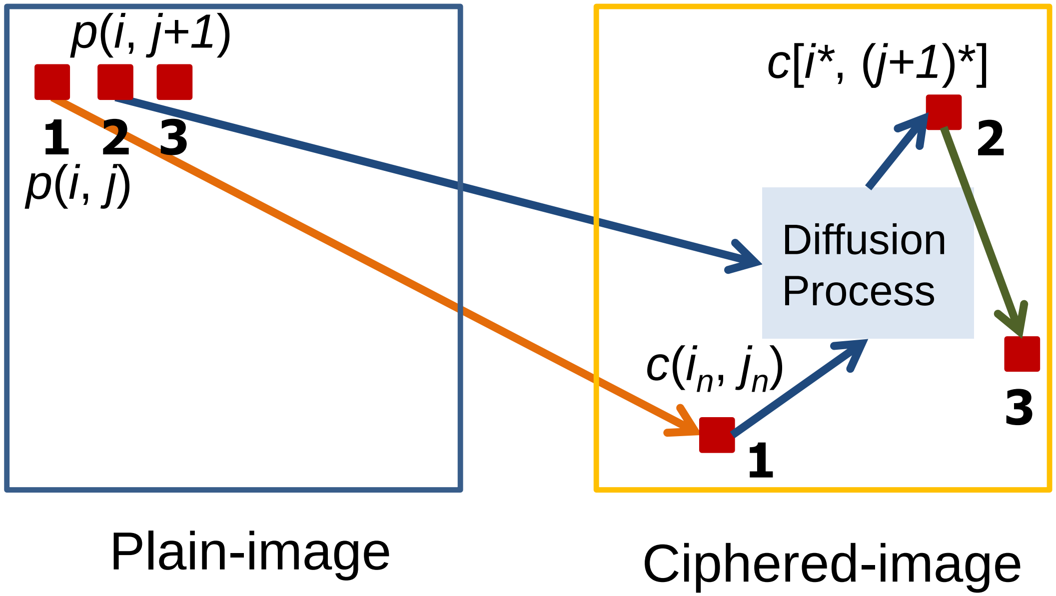


The confusion and diffusion processes are performed simultaneously in a single scan of plain-image pixels. More Speed

[Wong et al., 2009], [Wang et al., 2011], [Zhang et al., 2013], [Farajallah et al., 2016]

Chaos-based cryptosystems: 2nd type

- **2nd type : The diffusion process at the pixel level is governed by the confusion one**



$$\begin{cases} \left[i_n,\ j_n\right] = Cat\left[(i,\ j, u, v, ri, rj), M\right] \\ c(i_n,\ j_n) = p(i,\ j) \oplus q\left[f(z), L\right] \\ z = c(i_n,\ j_n) \end{cases} \qquad (1)$$

$$\begin{cases} z_{-1} = Kd \\ f(z) = \mu\ z \times (1 - z) \\ q\left[b, L\right] = \left\lfloor b \times 2^L \right\rfloor \end{cases} \qquad (2)$$

Chaos-based cryptosystems: 2nd type

Advantages:

- **The sensitivity to any modifications in the plain-image is increased. Indeed, equation (1) shows that $c(i_n, j_n)$ is influenced by both the diffusion key *Kd* and the previously ciphered pixel value z.**
- **The confusion effect can't be removed using a homogeneous plain-image:**

$$HI \xrightarrow{Kp1} C1$$

$$HI \xrightarrow{Kp2} C2 \neq C1$$

In separate confusion – diffusion architecture : $c(i) = v(i) \oplus q\left\{ f\left[c(i-1)\right], L\right\}$

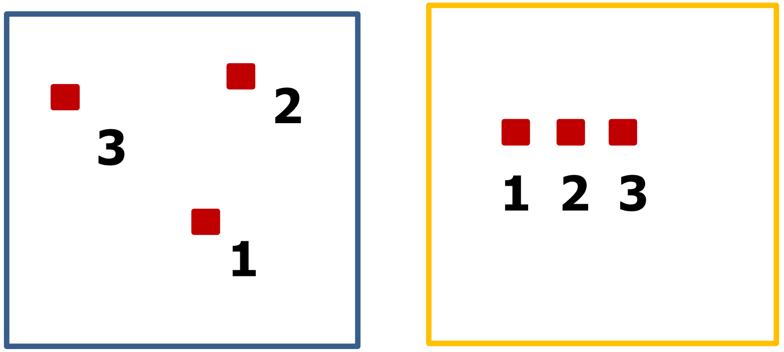
$$HI \xrightarrow{Kp1} c1(i) = v \oplus q\left\{ f\left[c1(i-1)\right], L\right\} \rightarrow C1$$

$$HI \xrightarrow{Kp2} c2(i) = v \oplus q\left\{ f\left[c2(i-1)\right], L\right\} \rightarrow C2 = C1$$

Chaos-based cryptosystems: 2nd type

- **Reverse use of 2-D chaotic map: 1er algo of [Zhang et al., 2013]**

$$\begin{cases} \left[i_n,\ j_n\right] = Cat\left[(i,\ j, u, v), M\right] \\ c(i,\ j) = p(i_n,\ j_n) \oplus f(z) \qquad (3) \\ z = c(i,\ j) \end{cases}$$



Plain-image Ciphered-image

$$\begin{cases} f(z) = Mod\left\{\left[\mu\, z \times (1-z) \times 1000\right], (256)\right\} \\ z(-1) = Kd \end{cases} \quad (4)$$

2 Logistic maps are used as key generator: *Kp*, *Kd*

- **Partial Cryptanalysis of the 1er algorithm of Zhang by removing the diffusion effect using equation (5)**

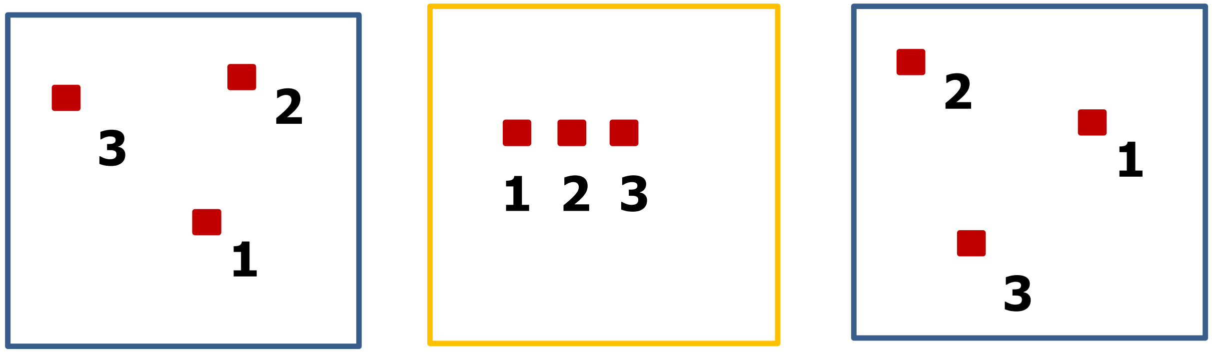
$$\begin{cases} c(k) = p(k_n) \oplus f\left[z(k)\right] = p(k_n) \oplus f\left[c(k-1)\right] \\ p(k_n) = c(k) \oplus f\left[c(k-1)\right] \\ k = i \times M + j \qquad k_n = i_n \times M + j_n \end{cases} \quad (5)$$

P is a permuted version of the original plain-image

[Farajallah et al., 2015, to appear in IJBC Journal]

Chaos-based cryptosystems: 2nd type

- 2nd algorithm of [Zhang et al., 2013]



Plain-image Mid-image Ciphered-image

Confusion process

$$\begin{cases} \left[i_{n1},\ j_{n1}\right] = Cat\left[(i,\ j, u1, v1), M\right] \\ mid(i,\ j) = p(i_{n1},\ j_{n1}) \oplus rand1(z1) \\ \left[i_{n2},\ j_{n2}\right] = Cat\left[(i,\ j, u1, v1), M\right] \\ c(i_{n2}, j_{n2}) = mid(i,\ j) \\ z1 = c(i_{n2}, j_{n2}); \quad z1(-1) = Kp1 \end{cases} \quad (6)$$

Diffusion process

$$ciph\_c(i) = c(i_{n2},\ j_{n2})$$

$$\begin{cases} ciph\_d(i) = ciph\_c(i) \oplus rand2(z2) \\ z2 = ciph\_d(i) \\ z2(-1) = Kd2 \end{cases} \quad (7)$$

*rand*1, *rand*2 are a random arrays with 256 distinct elements generated by 2 Logistic maps: $rand(z) = Mod\left\{\left[\mu\, z \times (1-z) \times 1000\right], (256)\right\}$ (8)

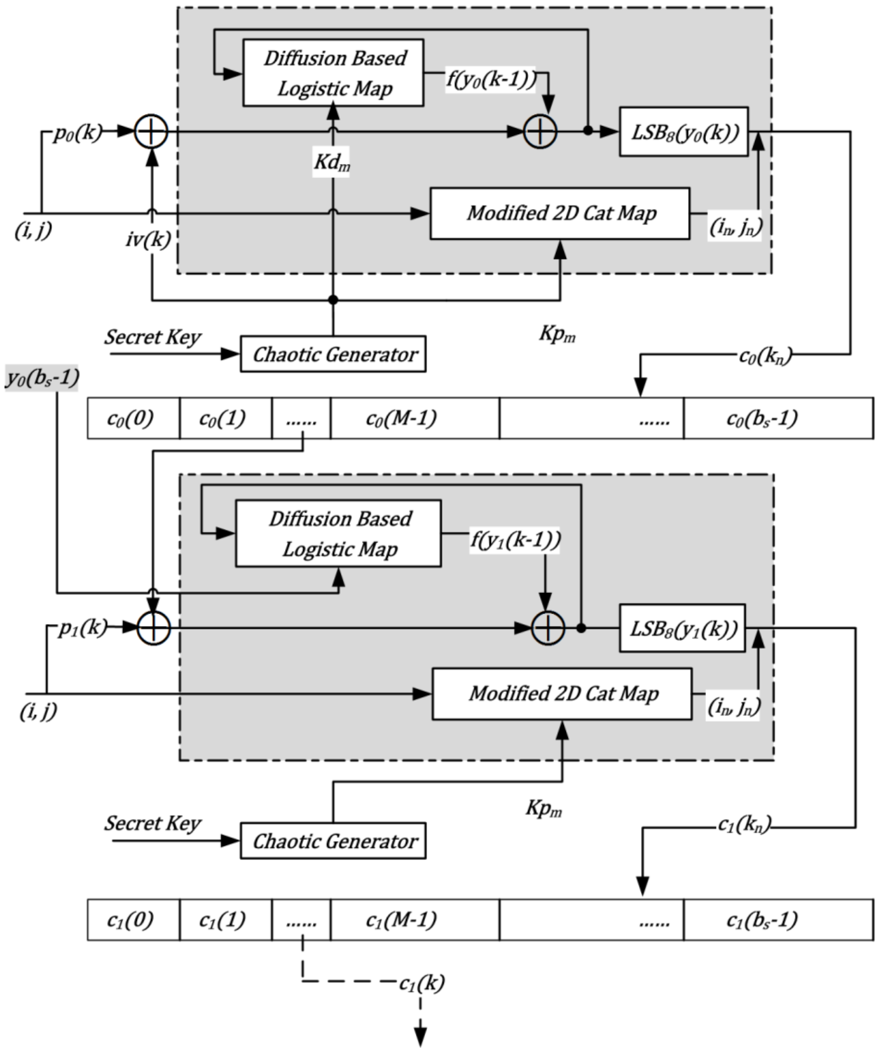
Chaos-based cryptosystems: 2nd type

- **Proposed algorithm**

$$y_l(k) = p_l(k) \oplus s_{l-1}(k) \oplus f(y_l(k-1))$$

$$c_l(k_n) = LSB_8[y_l(k)]$$

$$s_{l-1}(k) = \begin{cases} iv(k) & \text{if } l = 0 \\ c_{l-1}(k) & \text{if } l > 0 \end{cases}$$



Diffusion process :

V1: Discrete Logistic map with N = 32 bits

V2: Discrete Skew tent map with N = 32 bits

V3 : Look up table with N = 8 bits of the Skew tent map

[Farajallah et al., 2016, in IJBC Journal]

Performance in terms of time consuming

Average Encryption / Decryption time

Encryption Throughput

Number of needed Cycles per Bytes

$$ET = \frac{Image\ Size\ (Byte)}{Average\ Encyption\ Time\ (second)} \qquad NCpB = \frac{CPU\ Speef\ (Hertz)}{ET(Byte/s)}$$

Average is done by encrypting the test image at least 100 times with different secret keys each time

C language, PC: 3.1 GHz processor Intel Core TM i3-2100 CPU, 4GB RAM Windows 7, 32-bit operating system.

Performance in terms of time consuming

Lena image of size 256 X 256 X 3

Crypto3-V1 : Discrete Logistic map-32 bit (as diffusion)

Crypto3-V2: Discrete Skew tent map-32 bit (as diffusion)

Crypto3-V3: Look up table-8 bit of the Skew tent map (as diffusion)

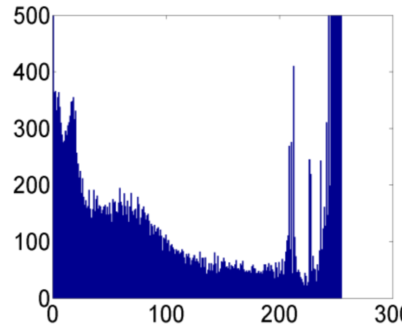
Cryptosystem	Enc / Dec times (ms)	ET (MBps)	Cycles per Byte
Crypto 1	9.9 / 32.4	18.9	157
Crypto 2	8.38 / 8.48	22.3	132
Crypto 3-V1	2.1 / 2.6	93.9	32
Crypto 3-V2	4.15 / 4.79	45.3	65
Crypto 3-V3	1.3 / 1.4	140.7	21
Zhang et al	7.5 / 8.25	25	122
Wang et al	7.79 / 8.39	24.1	208
Wong et al	15.59 / 16.77	7.2	417
AES	1.75 / 1.8	122	24

Performance in terms of security analysis

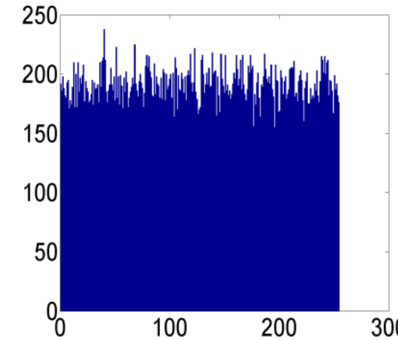
- **Statistical analysis: Histogram and correlation (Confusion property)**

Cameraman

Ciphered



Histograms :Plan



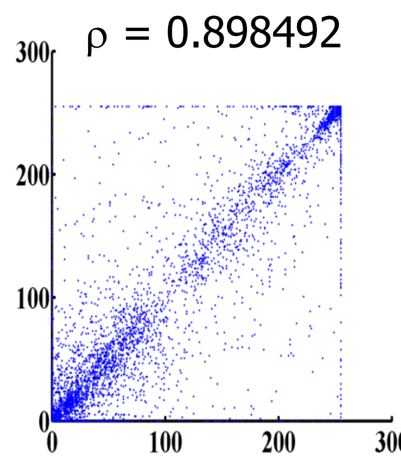
Ciphered

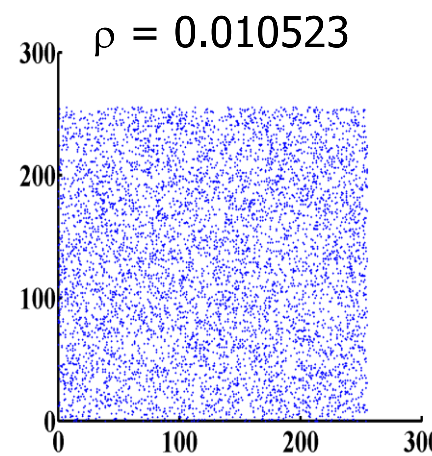
Theoretical Chi-square is 293 in case of alpha=0.05 and number of intervals = 256.

Chi-square Exp value = 255.12

N=8000 pairs (x, y) of two adjacent pixels randomly selected in vertical, horizontal, and diagonal directions from the original and encrypted images.

$$\rho_{xy} = \frac{\sum_{i=1}^{N}[(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2} \times \sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$





Correlation of adjacent horizontal pixels of plain and ciphered images

Performance in terms of security analysis

Cryptanalytic Attacks: ordered, for an attacker, from the hardest type to the easiest:

1) **Ciphertext only:** the attacker has the ciphertext of several messages.
2) **Known plaintext attack:** the attacker has access to the ciphertext of several messages and their corresponding plaintext.
3) **Chosen plaintext attack:** the attacker has obtained temporary access to the encryption machinery, and then he can choose a specific plaintext to encrypt and obtain the corresponding ciphertext.
4) **Chosen ciphertext attack**: the attacker has obtained temporary access to the decryption machinery, and then he can choose a specific ciphertext to decrypt and obtain the corresponding plaintext.

If a cryptosystem is able to resist chosen plaintext attack, then it is also resistant to all the other attacks. It is computationally secure

Performance in terms of security analysis

- **Plaintext sensitivity attack: Diffusion property**

To resist the chosen plaintext attack and the differential attack, the cryptosystem should be highly sensitive to one bit change in the plaintext.

We evaluate the plaintext sensitivity as follows:

For each of the 1000 random secret keys, we compute the Hamming distance, versus the number of rounds *r*, between two cipher-text images *C1* and *C2*, resulted from two chosen plaintext images *I1* and *I2,* with:

I1 = [0, 0, …,0] and *I2* = [0, 0, …1_i,…,0], differ only by one bit (chosen randomly).

$$HD(C1, C2) = \frac{1}{|Ib|} \sum_{k=1}^{|Ib|} C1(k) \oplus C2(k)$$

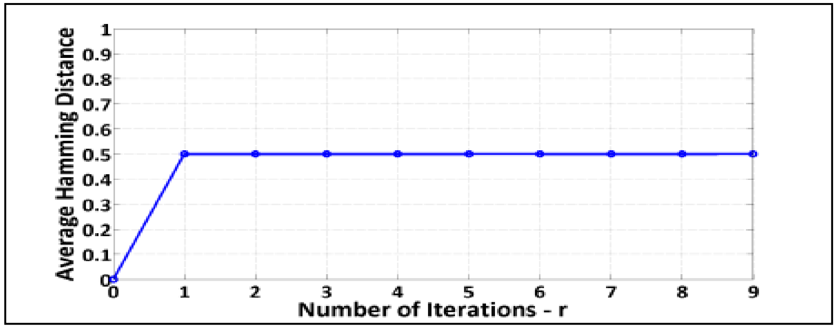
with $|Ib| = L \times C \times P \times 8$ size of the image in bits

If the Hamming distance is close to 50% (probability of bit changes close to 1/2), then the previous attacks would become ineffective.

This test gives also the minimum number of rounds *r*, needed to overcome the plaintext sensitivity attack.

Performance in terms of security analysis

▪ **Plaintext sensitivity attack: Diffusion property**



Average Hamming distance (over 1000 keys) versus the number of rounds *r*.
With *r* =1, the effect avalanche is reached.

For two random images the expected values of *NPCR* and *UACI* are:

***E(NPCR)* = 99.609 %**

***E(UACI)* = 33.463 %**

NPCR and _UACI_ criteria

Number of pixel change rate (_NPCR_)

$$NPCR = \frac{\sum_{p=1}^{P}\sum_{i=1}^{L}\sum_{j=1}^{C} D(i,j,p)}{L \times C \times P} \times 100\%$$

$$D(i,j,p) = \begin{cases} 0 \; if \;\; C1(i,j,p) = C2(i,j,p) \\ 1 \; if \;\; C1(i,j,p) \neq C2(i,j,p) \end{cases}$$

Unified average changing intensity (UACI)

$$UACI = \frac{1}{M \times N \times P} \times \sum_{p=1}^{P}\sum_{i=1}^{L}\sum_{j=1}^{C} \frac{\left|C1(i,j,p) - C2(i,j,p)\right|}{255} \times 100\%$$

Performance in terms of security analysis

Propose d Crypto	Image	Size	HD	NPCR	UACI
V1	Lena	512x512	0.500014	99.610	33.464
V2	Lena	512x512	0.499995	99.608	33.463
V3	Lena	512x512	0.499994	99.607	44.465

- **Key sensitivity test**

A good encryption scheme should be sensitive to the secret key in process of both encryption and decryption.

$$I1 \xrightarrow{Key} C1 \qquad C1 \xrightarrow{Key} I1$$

$$I1 \xrightarrow{Key\ with\ 1\ bit\ changes} C2 \neq C1 \qquad C1 \xrightarrow{Key\ with\ 1\ bit\ changes} I2 \neq I1$$

To quantify the effectiveness of any algorithm, researchers use the *NPCR* and *UACI* criteria

Enhancement of two spatial steganography algorithms by using a chaotic system: comparative analysis

D. Battikh, S. El Assad
*B. Bakhache, O. Deforges, *M. Khalil

Polytech Nantes, school of engineering of the university of Nantes – France
IETR Laboratory, UMR CNRS 6164; Image team - site of Nantes

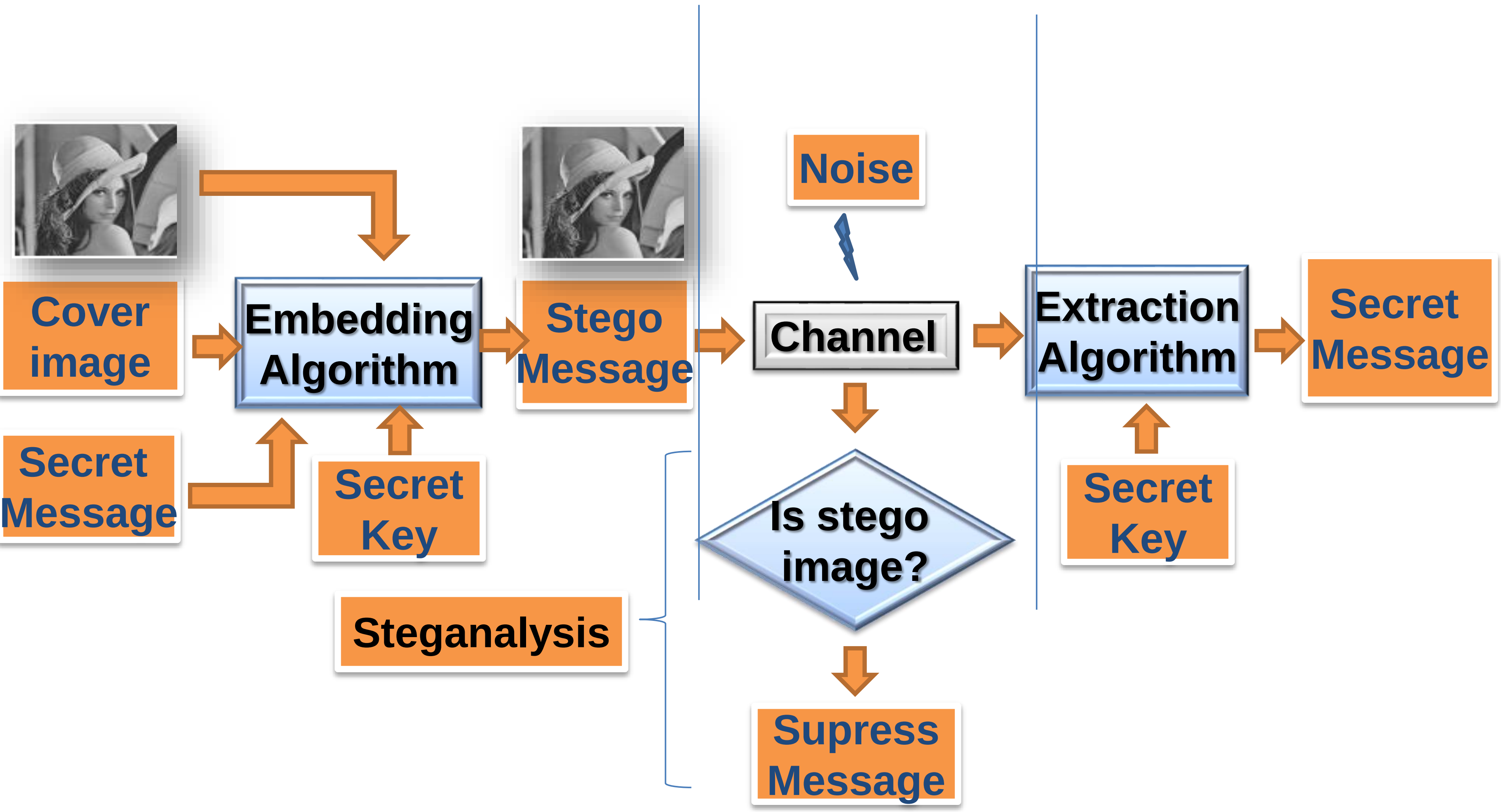
*Lebanese International University of Beirut, Lebanon

The 8th International Conference for Internet Technology and Secured Transactions, ICITST-2013, December, 9 - 12, 2013, London, UK

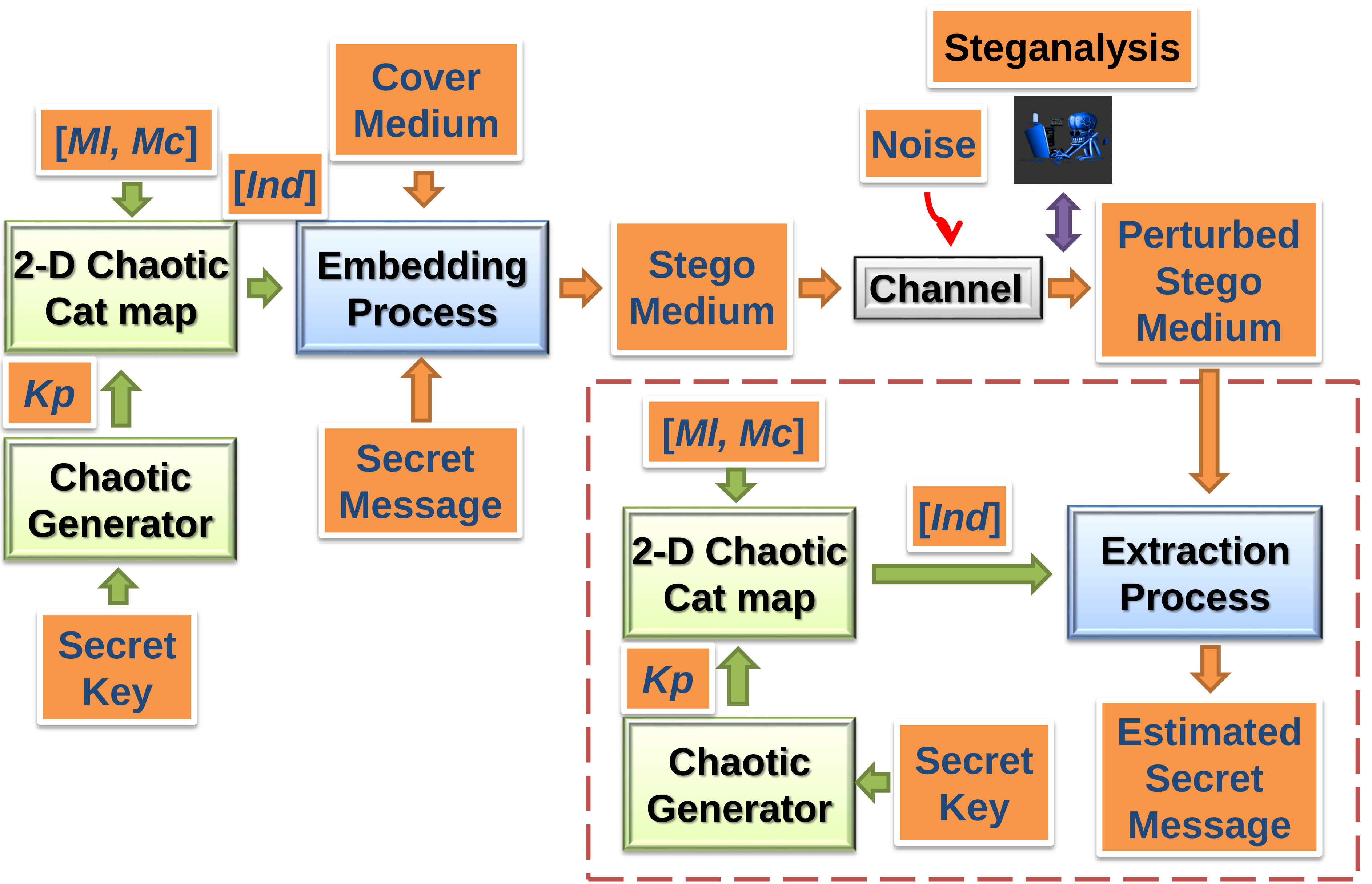
Outline

- **Principle of data hiding in spatial LSB domain**
- **Structure of the proposed chaos-based steganography systems**
- **Enhanced Adaptive data hiding in Edge areas of images with spatial Low Significant Bit domain systems : EAE-LSB**
- **Enhanced Edge Adaptive Image Steganography Based on LSB Matching Revisited : EEA-LSBMR**
- **Experimental Results.**
- **Conclusion and perspectives.**

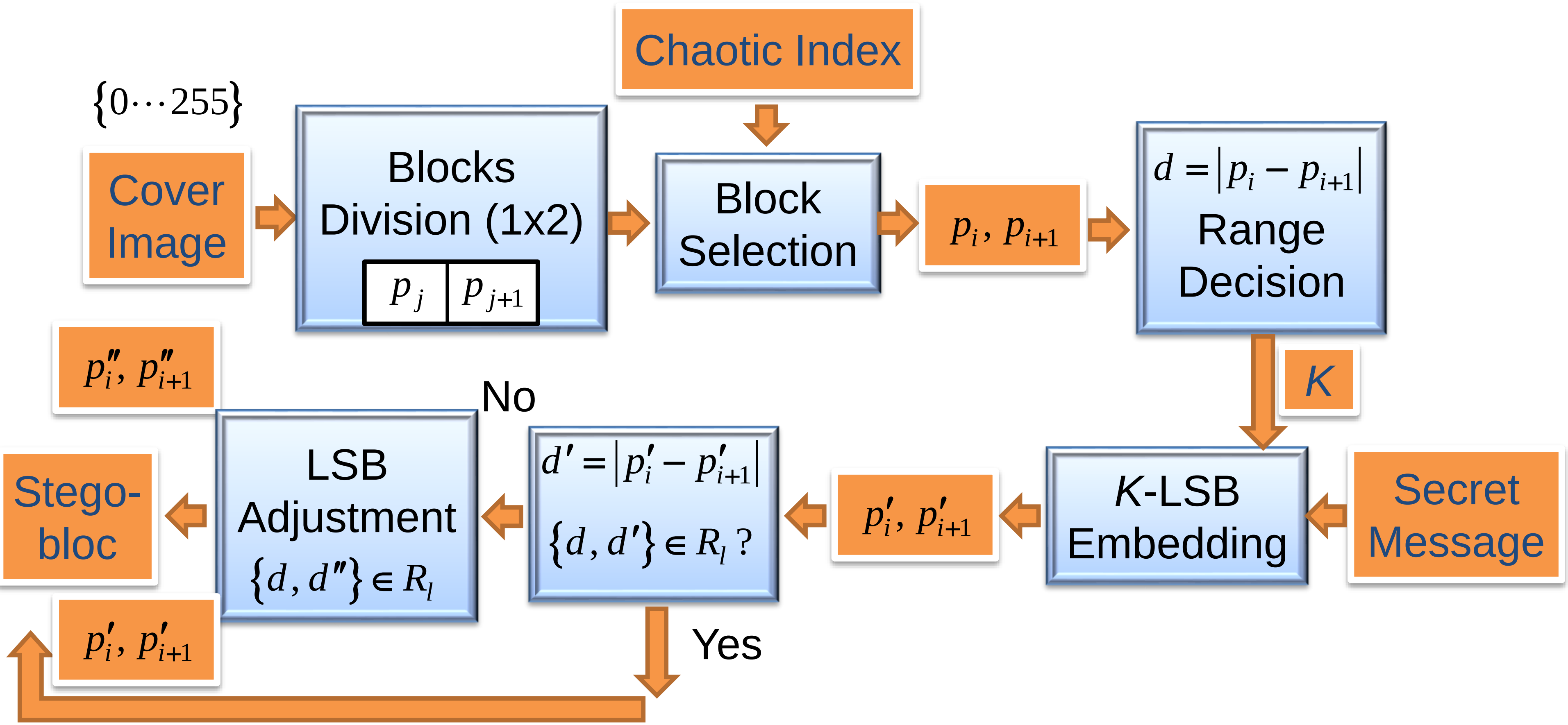
Principle of data hiding in spatial LSB domain



Structure of the proposed chaos-based steganography systems



EAE-LSB : Adaptive Embedding process



Low Level $K = 3$	Middle Level $K = 4$	High Level $K = 5$
R_1 = [0, 15]	R_2 = [16, 31]	R_3 = [32, 255]

0 ← Range → 255

EAE-LSB : Adaptive Embedding process

- Divide image in 2-pixel overlapped blocks
- Chose block (p_i, p_{i+1}) using chaotic index (*Ind*)
- Compute block difference $d = |p_i - p_{i+1}|$, find its corresponding range R_l and identify K : R_1 =[0, 15] => K=3; R_2 =[16, 31] => K=4; R_3 =[32, 255] => K=5
- Hide 2K bits message in every block using K-LSB insertion => (p'_i, p'_{i+1})
- Compute block difference $d' = |p'_i - p'_{i+1}|$, and test if { d, d' } are in the same range R_l.
- If yes, than Stego-block (p'_i, p'_{i+1}) is carrying the secret message.
- Else, apply the LSB adjustment process => Stego-block (p''_i, p''_{i+1})

EAE-LSB : Adaptive Embedding process: LSB adjustment process

$d \in R_l,\ d' \in R_t,\quad l \neq t$

Input : (p'_i, p'_{i+1}), (p_i, p_{i+1}) ; Output : (p''_i, p''_{i+1})

If ($d < d'$)

if ($p'_i >= p'_{i+1}$)

(p''_i, p''_{i+1}) =Best_Choice_Of $\{(p'_i, p'_{i+1}+2^K), (p'_i - 2^K, p'_{i+1})\}$

else

(p''_i, p''_{i+1}) =Best_Choice_Of $\{(p'_i, p'_{i+1} - 2^K), (p'_i + 2^K, p'_{i+1})\}$

Else ($d > d'$)

if ($p'_i >= p'_{i+1}$)

(p''_i, p''_{i+1}) =Best_Choice_Of $\{(p'_i, p'_{i+1} - 2^K), (p'_i + 2^K, p'_{i+1})\}$

else

(p''_i, p''_{i+1}) =Best_Choice_Of $\{(p'_i, p'_{i+1}+2^K), (p'_i - 2^K, p'_{i+1})\}$

End

Best_Choice_Of = MSE $\{(p_i, p_{i+1}), (p''_i, p''_{i+1})\}$

$$\mathrm{MSE} = \{(p_i - p''_i)^2 + (p_{i+1} - p''_{i+1})^2\}$$

Example of Embedding process

Secret bits : 1011 0110

p_i d p_{i+1}

1001 0101 149 173 1010 1101

$d = |p_i - p_{i+1}| = |149 - 173| = 24$

$d \in R_2 \Rightarrow K=4$

p'_i d' p'_{i+1}

1001 **1011** 155 166 1010 **0110**

$d' = |p'_i - p'_{i+1}| = |155 - 166| = 11$

$d > d' \in R_1$

$\Downarrow$

Adjustment process

$Case: p'_i \le p'_{i+1} = 166$

(p''_i, p''_{i+1}) =Best_Choice_Of $\{(p'_i, p'_{i+1}+2^K), (p'_i - 2^K, p'_{i+1})\}$

(p''_i, p''_{i+1}) =Best_Choice_Of {(155,182), (139,166)}

$MSE = \{(p_i - p''_i)^2 + (p_{i+1} - p''_{i+1})^2\}$

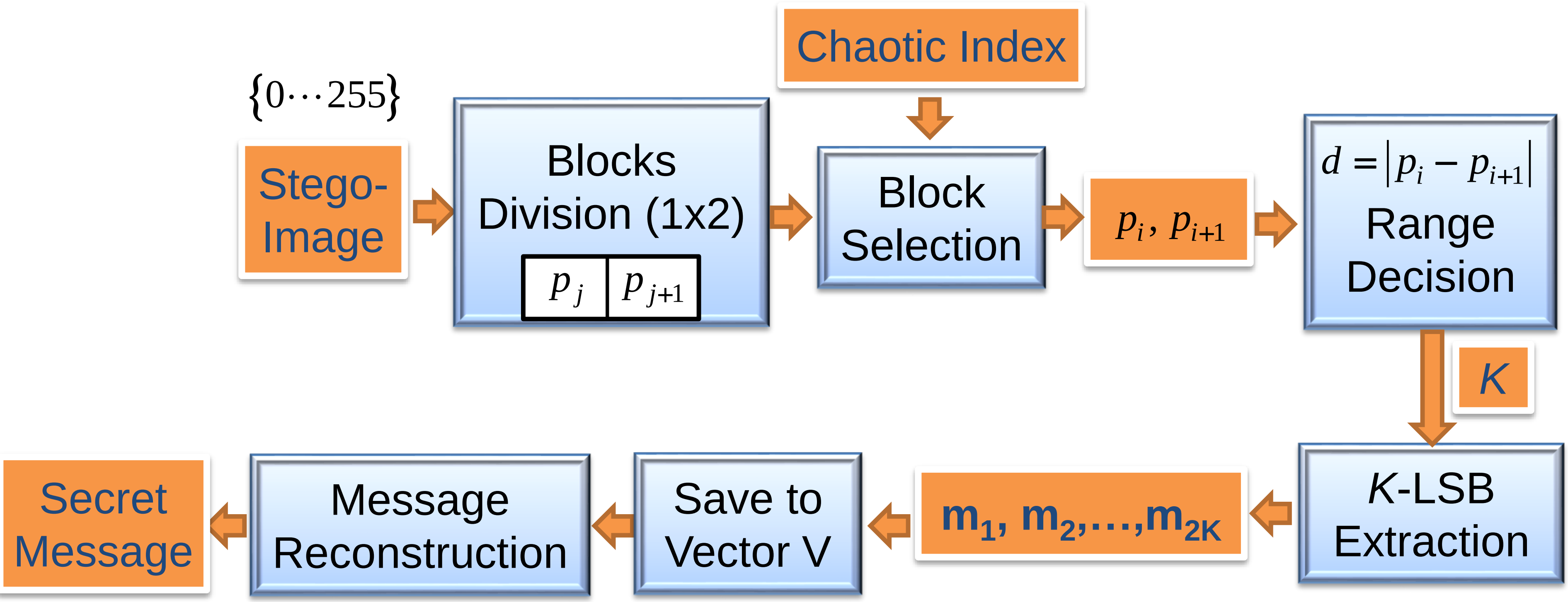
$MSE_1 : (149 - 155)^2 + (173 - 182)^2 = 117$

$MSE_2 : (149 - 139)^2 + (173 - 166)^2 = 149$

Best_Choice

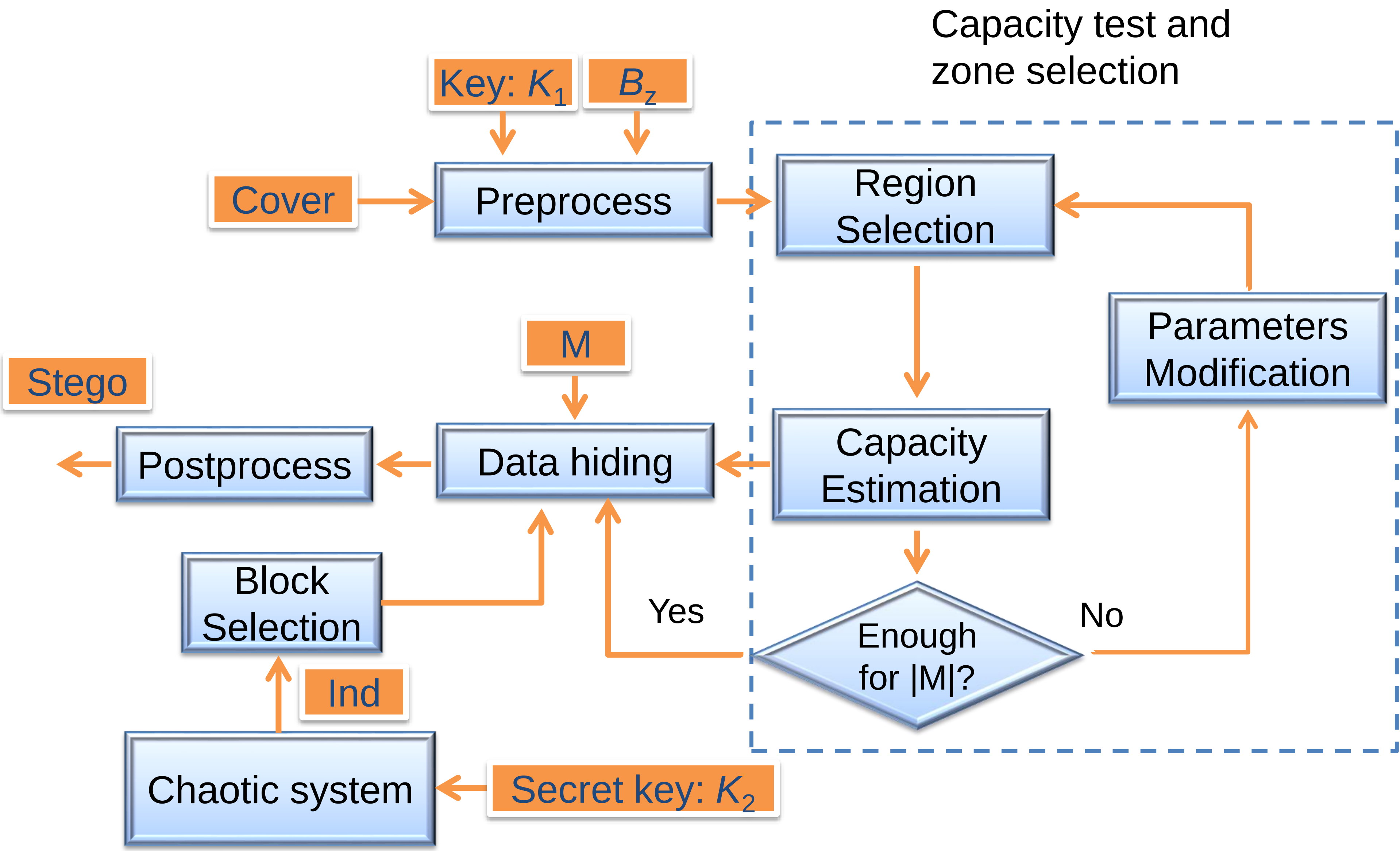
$p''_i = 155$ $p''_{i+1} = 182$

EAE-LSB : Extraction process



- Divide stegoimage in overlapped 2-pixels blocks
- Select block chaotically (p_i, p_{i+1}) as for insertion
- Compute block difference d and identify *K*-LSBs for the corresponding range
- Extract *K* LSB secret bits from p_i, *K* LSB secret bits from $p_{i+1}$and add to message vector V
- Reconstruct secret message from 2K bits sequence groups of V

EEA-LSBMR : Adaptive Embedding process



EEA-LSBMR : Adaptive Embedding process: 4 steps

1- Preprocess:

- Divide the cover image into non-overlapping blocks of B_z x B_z pixels (Bz = 4, 8, 12).
- Rotate each block by a random degree in the range of (0°, 90°, 180°, 270°) according to a secret key K_1
- Rearrange the resulting image as a row vector *V* by raster scanning, and then divide *V* into no overlapping 2-pixel blocks : (p_i, p_{i+1}).

2- Capacity test and zone selection:

- For each $t \in$ {1, 2,………, 31}, calculate the set of pixel pairs such as:

$$EU(t) = \left\{(p_i, p_{i+1}) / \left|p_i - p_{i+1}\right| \geq t, \forall (p_i, p_{i+1}) \in V\right\}$$

- Then, calculate the threshold *T* by : $T = \arg\max_t \left\{2 \times \left|EU(t)\right| \geq \left|M\right|\right\}$

where : |*EU(t)*| denotes the total number elements in the set of *EU(t)* and |M| the size of the secret message.

EEA-LSBMR : Adaptive Embedding process: 4 steps

3- Data hiding:

- Calculate the set of: $EU(T)=\left\{(p_i,p_{i+1})/\left|p_i-p_{i+1}\right|\geq T,\forall(p_i,p_{i+1})\in V\right\}$
- Select in a chaotic manner a block of the above set and perform data hiding according to the following 4 cases:

$$
\begin{aligned}
&Case\ 1:\ LSB(p_i)=m_i\ \&\ \ f(p_i,p_{i+1})=m_{i+1}\rightarrow(p'_i,p'_{i+1})=(p_i,p_{i+1})\\
&Case\ 2:\ LSB(p_i)=m_i\ \&\ \ f(p_i,p_{i+1})\neq m_{i+1}\rightarrow(p'_i,p'_{i+1})=(p_i,p_{i+1}+r)\\
&Case\ 3:\ LSB(p_i)\neq m_i\ \&\ \ f(p_i-1,p_{i+1})=m_{i+1}\rightarrow(p'_i,p'_{i+1})=(p_i-1,p_{i+1})\\
&Case\ 4:\ LSB(p_i)\neq m_i\ \&\ \ f(p_i-1,p_{i+1})\neq m_{i+1}\rightarrow(p'_i,p'_{i+1})=(p_i+1,p_{i+1})
\end{aligned}
$$

where m_i and m_{i+1} denote 2 secret bits to embed

r is a random value in [-1, +1] and $f(a,b)=LSB\left(\left\lfloor\frac{a}{2}\right\rfloor+b\right)$

(p'_i,p'_{i+1}) : Pixel pair after data hiding

EEA-LSBMR : Adaptive Embedding process: 4 steps

3- Data hiding:

$$if\ \ (p'_i, p'_{i+1}) \notin [0, 255]\ or\ |p'_i - p'_{i+1}| < T \Rightarrow readjustment$$

- Readjustment: $$(p''_i, p''_{i+1}) = \arg\ \min_{(e_1, e_2)}\{|e_1 - p_i| + |e_2 - p_{i+1}|\}$$

with:

$$\begin{cases} e_1 = p'_i + 4q_1 \\ e_2 = p'_{i+1} + 2q_2 \end{cases} \quad q_1, q_2 \in Z \qquad (1)$$

$$|e_1 - e_2| \geq T,\ \ 0 \leq e_1, e_2 \leq 255, \quad 0 \leq T \leq 31$$

Finally :

$$LSB(p''_i) = m_i \quad \& \quad f(p''_i, p''_{i+1}) = m_{i+1}$$

$$with \quad 0 \leq p''_i, p''_{i+1} \leq 255,\ |p''_i - p''_{i+1}| \geq T$$

4- Post process:

- The resulting image is divided into non overlapping B_z x B_z blocks. The blocks are then rotated by a random degree in the range of (0°, 90°, 180°, 270°) according to a secret key K_1.
- (T, B_z) are embedded in the stego image into a preset region.

EEA-LSBMR : Example of Embedding process

Let suppose:

$$\begin{cases}(p_i, p_{i+1}) = (62, 81) \\ (m_i, m_{i+1}) = (1, 0) \\ T = 19\end{cases} \Rightarrow$$ We verify that:

$$\left|p_i - p_{i+1}\right| = 19 \geq T$$

$$and \quad \begin{cases} LSB(62) = 0 \neq m_i \\ LSB\left\{\left\lfloor\left(\frac{62-1}{2}\right)\right\rfloor + 81\right\} = 1 \neq m_{i+1}\end{cases}$$

Therefore, we invoke case 4: $(p'_i, p'_{i+1}) = (p_i + 1, p_{i+1}) = (63, 81)$

Then the new absolute difference is:

$$\left|p'_i - p'_{i+1}\right| = \left|63 - 81\right| = 18 < T \Rightarrow r\mathrm{e}adjustment\ according\ to\ (1)\ and\ \ finally\ get:$$

$$q_1 = 0, q_2 = 1$$

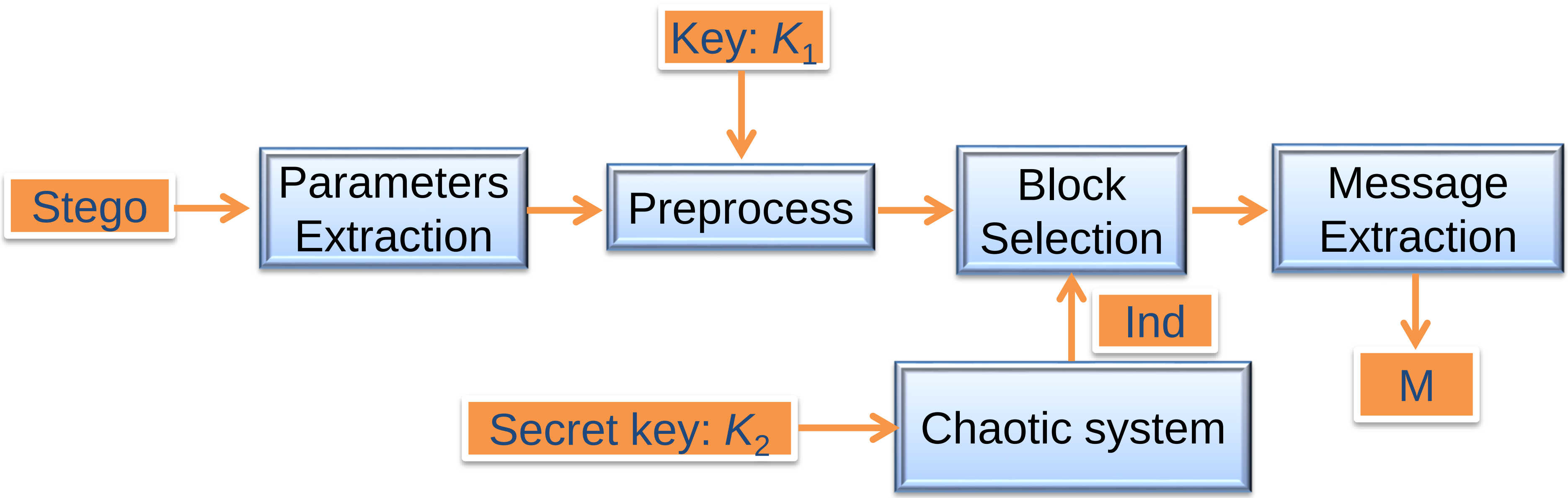
$$p''_i = p'_i + 4q_1 = 63 + 4 \times 0 = 63$$

$$p''_{i+1} = p'_{i+1} + 2q_2 = 81 + 2 \times 1 = 83$$

In such case, we have: $\left|p''_i - p''_{i+1}\right|\left|63 - 83\right| = 20 \geq T\ and$

$$and \quad \begin{cases} LSB(63) = m_i = 1 \\ LSB\left\{\left\lfloor\left(\frac{63-1}{2}\right)\right\rfloor + 83\right\} = m_{i+1} = 0\end{cases}$$

EEA-LSBMR : Extraction process



- Extract (T, B_z)
- Divide the stego image into blocks of B_z x B_z pixels and rotate the blocks by random degrees based on the secret key K_1.
- Rearrange the resulting image as a row vector Vs by raster scanning and divide Vs into no overlapping 2-pixel blocks : (p_i, p_{i+1}).

EEA-LSBMR : Extraction process

- Chose blocks (p_i, p_{i+1}) whose absolute differences are greater than or equal to T according to the chaotic system.
- Extract the two secret bits from each qualified bloc as follows :

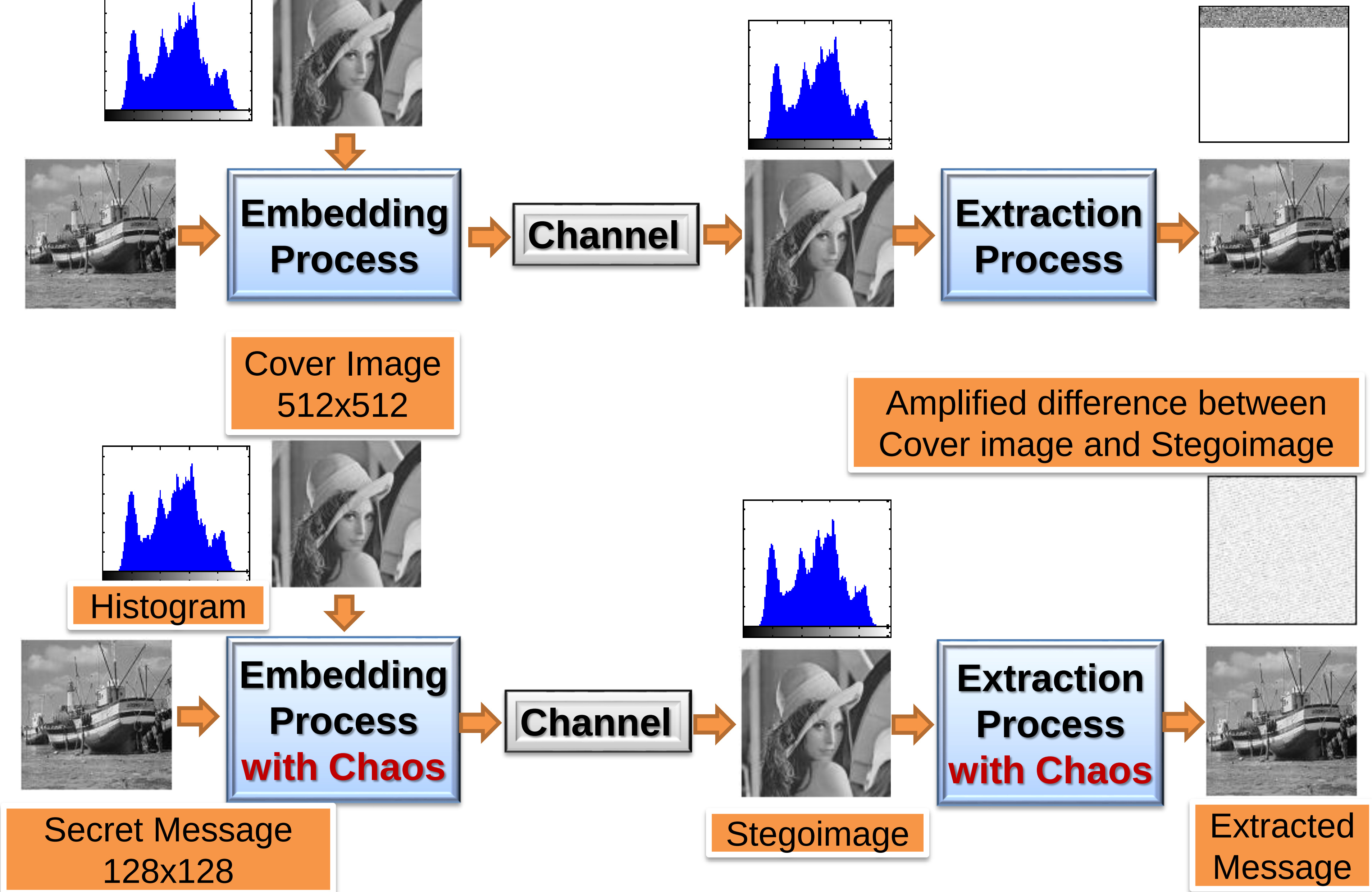
$$m_i = LSB(p_i), \quad m_{i+1} = LSB\left(\left\lfloor \frac{p_i}{2} \right\rfloor + p_{i+1}\right)$$

- For instance, if (p_i, p_{i+1}) = (63, 83), with T=19, so, we get the secret bits:

$$m_i = LSB(63) = 1, \quad m_{i+1} = LSB\left(\left\lfloor \frac{63}{2} \right\rfloor + 83\right) = 0$$

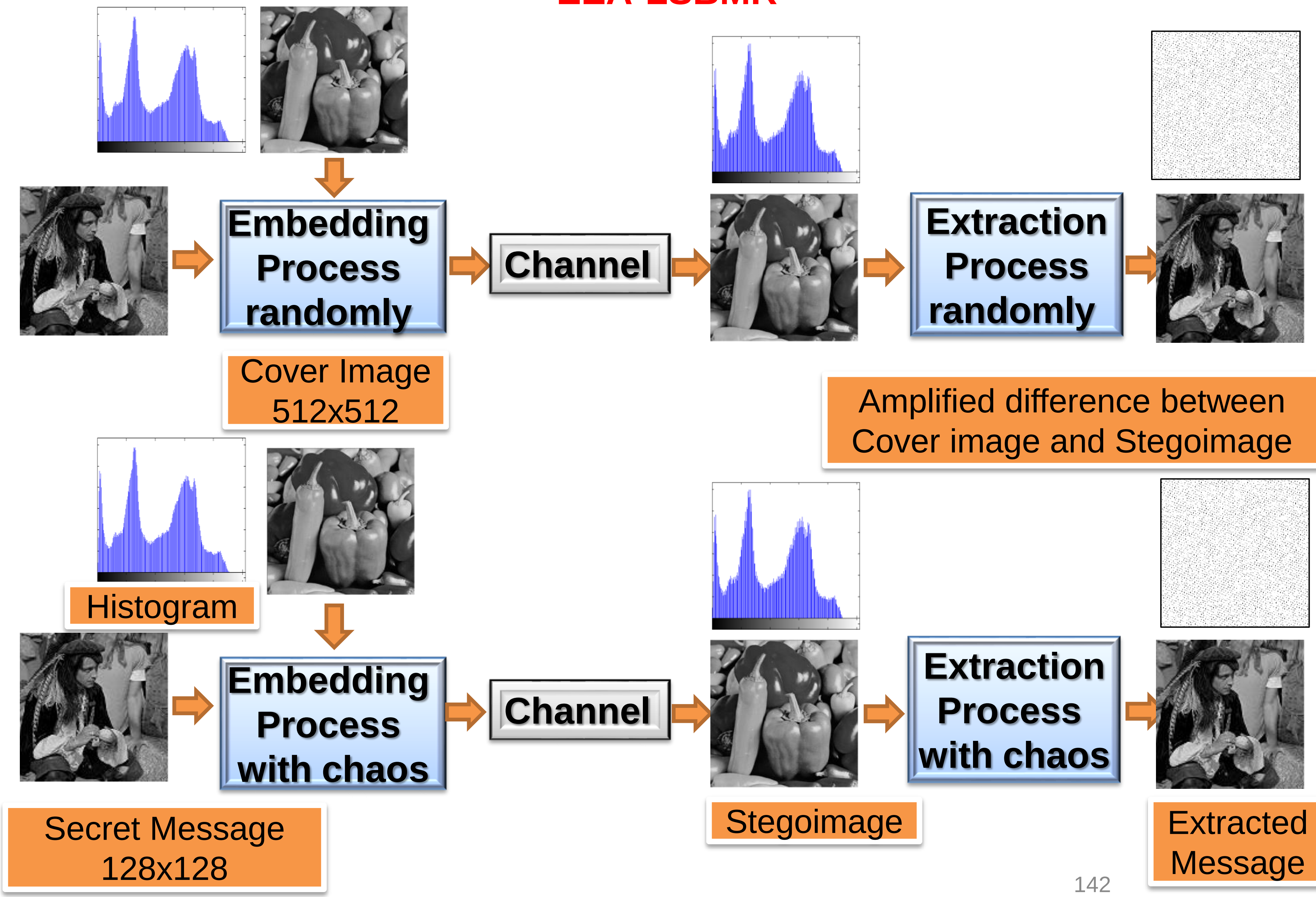
Experimental results : Embedding-Extraction without and with chaos

EAE-LSB



Experimental results : Embedding-Extraction without and with chaos

EEA-LSBMR



Experimental Results

- Same good performances in terms of secret message capacity and image quality

$$PSNR = 10 \times \log_{10}\left(\frac{M\text{ax}\, I^2(i,j)}{\frac{1}{M \times N}\left(\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left[I(i,j) - I_s(i,j)\right]^2\right)}\right)$$

Cover C	Message M	PSNR EAE-LSB	PSNR EEA-LSBMR
Lena (512x512)	32x32	60.03	70.35
	64x64	54.42	64.41
	100x100	50.33	60.51
	128x128	48.32	58.35
	256x256	42.49	--
Baboon (512x512)	32x32	57.55	70.52
	64x64	51.27	64.46
	100x100	47.19	60.59
	128x128	45.20	58.41
	256x256	39.40	--
Peppers (512x512)	32x32	59.43	69.71
	64x64	54.52	63.78
	100x100	50.25	59.86
	128x128	48.04	57,70
	256x256	42.42	--

Conclusion and perspectives

Conclusion :

- We demonstrated the contribution of the chaos in the information hiding and security.
- We designed an enhancement (message security) of two spatial steganographic algorithms: EAE-LSB and EEA-LSBMR, that have low distortions and high insertion capacity

Perspectives :

- Study the robustness of the above chaos-based steganography algorithms against steganalysis
- Design secure chaos-based steganography systems in frequency domain

References

1. Abbas Cheddad, Joan Condell, Kevin Curran, Paul Mc Kevitt, Digital image steganography: Survey and analysis of current methods, Signal Processing, Volume 90, Issue 3, March 2010, Pages 727-752
2. C.-H. Yang, C.-Y. Weng, and S.-J. Wang, “Adaptive data hiding in edge areas of images with spatial LSB domain systems”, IEEE Trans. on information forensics and security, vol. 3, no. 3, september 2008.
3. Luo, W., Huang, F. et Huang, J. (2010). Edge Adaptive Image Steganography Based on LSB Matching Revisited. IEEE transactions on information forensics and security, Vol. 5, NO. 2.
4. T.S. Parker, L.O. Chua, “Practical Numerical Algorithms for Chaotic Systems,” Springer-Verlag, 1989.
5. A. Baranovsky, D. Dames, “Design of One-Dimensional chaotic maps with prescribed properties,” International Journal of Bifurcationand chaos , vol. 5, pp. 1585-1598, Feb. 1995.
6. S. El Assad, H. Noura, I.Taralova, "Design and Analyses of Efficient Chaotic Generators for Cryptosystems," wcecs, pp.3-12, Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008, 2008

References

7. S. El Assad (85%), H. Noura (15%), Generator of chaotic Sequences and corresponding generating system WO Patent WO/2011/121,218, 2011.
8. S. El Assad, "Chaos Based Information Hiding and Security," in 7th International Conference for Internet Technology and Secured Transactions, IEEE, London, United Kingdom, 10-12 Dec. 2012, pp. 67-72. Invited paper.
9. C.Y. Song, Y.L. Qiao, and X.Z. Zhang, "An image encryption scheme based on new spatiotemporal chaos," Optik, October 2012
10. R. L. Tataru, D. Battikh, S. El Assad, H. Noura, O. Deforges, " Enhanced Adaptive Data Hiding in Spatial LSB Domain by using Chaotic Sequences" . IIH-MSP 2012: 85-88
11. C.-K. Chan and L.M Cheng, "Hiding data in images by simple LSB substitution ", The Journal of the Pattern Recognition society, pp. 469-474, 2004.
12. Mielikainen, J. (2006). LSB matching revisited. IEEE Signal Process. Lett., vol. 13, no. 5, pp. 285–287

Thanks for your Attention

Questions ?

References

[Burda K, 2006], “Error Propagation in Various Cipher Block Modes”. International Journal of Computer Science and Network Security, vol. 6, no. 11, 2006

[Caragata et al.,], “On the security of a new image encryption scheme based on a chaotic function”. Signal, Image and Video Processing, vol. 8, Issue 4, pp. 641-646.

[Chen et al., 2004], “A symmetric image encryption schemes based on 3D chaotic cat maps”. Chaos Solitons and Fractals vol. 21, 2004, pp. 749-761.

[Dworkin, 2001], “Recommendation for Block Cipher Modes of Operation; Methods and Techniques”. NIST Special Publication 800-38A, 2001 Edition.

[El Assad et al., 2008], “Design and analyses of efficient chaotic generators for cryptosystems" . WCECS, pp. 3-12, 2008, Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science, 2008.

[El Assad et Noura, 2011], “Generator of chaotic Sequences and corresponding generating system” WO Patent WO/2011/121,218,2011.
PCT Extension: Europe : EP-2553567 A1, February 2013.
China : CN-103124955 A, May 2013. United States: US-20130170641, July 2013.

References

[El Assad et al., 2014],“Chaos-based Block Ciphers: An Overview”, IEEE, 10th International Conference on Communications, COMM-2014, Bucharest, Romania, May 2014, pp. 23-26. Invited talk

[El Assad, Farajallah, 2016], “A new Chaos-Based Image Encryption System”. Signal Processing: Image Communication 41, (2016) 144-157.

[Farajallah et al., 2013], ”Dynamic adjustment of the chaos-based security in real-time energy harvesting sensors”. IEEE, International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China, August 2013, pp. 282-289.

[Farajallah et al., 2016], “Fast and secure chaos-based cryptosystem for images”., International Journal of Bifurcation and Chaos, IJBC, Vol. 26, No. 2 (2016) 1650021 (21 pages).

[Fridrich , 1998], “Symmetric ciphers based on two-dimensional chaotic maps”. Int. J. Bifurcat Chaos, vol. 8, no. 6, 1998, pp. 1259-1284.

[Lian et al., 2005a], “A bloc cipher based on a suitable use of the chaotic standard map“. Chaos Solitons and Fractals, vol. 26, 2005, pp. 117-129.

References

[Lian et al., 2005b], “Security analysis of a chaos-based image encryption algorithm”. Physica A vol. 351, 2005, pp. 645-661.

[Lozi, 2012],”Emergence of randomness from chaos”, International Journal of Bifurcation and Chaos, IJBC, Vol. 22, No. 2 (2012) 1250021 (15 pages).

[Masuda et al., 2006], “Chaotic block ciphers: from theory to practical algorithms”. IEEE Trans on Circuits and Systems-I, vol. 53, no. 6, 2006, pp. 1341–1352.

[Abu Taha et al., 2017], “Design and Efficient Implementation of a Chaos-based Stream Cipher”, IJITST, International Journal of Internet Technology and Secured Transactions, to be published, 2017, 15 pages

[Schneier, 1996], “Applied Cryptography — Protocols, Algorithms, and Source Code”, C. John Wiley & Sons, Inc., New York 2nd edition.

[Zhang et al., 2013], “An image encryption scheme using reverse 2-dimentional chaotic map and dependent diffusion”. Commun Nonlinear Simulat, vo. 18, 2013, pp. 2066-2080.

[Wang et al., 2009], “A chaos-based image encryption algorithm with variable control parameters”. Chaos Solitons and Fractals vol. 41, 2009, pp. 1773-1783.

References

[Wang et al., 2011], “A new chaos-based fast image encryption algorithm”. Applied Soft Computing, vol. 11, 2011, pp. 514-522.

[Wong et al., 2008], “A fast image encryption scheme based on chaotic standard map“. Physics Letters A, vol. 372, no. 15, 2008, pp. 2645-2652.

[Wong et al., 2009], “An efficient diffusion approach for chaos-based image encryption”. Chaos Solitons and Fractals vol. 41, 2009, pp. 2652-2663.