Chaotic sequences based image encryption

Kirill Trofimov
Institute of Radioelectronics
Riga Technical University
Riga, Latvia
kirils.trofimovs@edu.rtu.lv

Sergej Umnov
Institute of Radioelectronics
Riga Technical University
Riga, Latvia
sergejs.umnovs@edu.rtu.lv

***Abstract*— In this paper we propose an image encryption method based on chaotic sequences produced by a pair of Vilnius oscillators. The solution is provided as a VHDL model and is tested against implementation in Matlab.**

***Index terms*—vilnius oscillator, image encryption, chaos**

I. Introduction

Image encryption is a common process in modern communication and storage systems. It is used to protect confidentiality and integrity of digital images from unauthorized access. Considering the omnipresence of visual data, it's processing takes a considerable amount of compute resources. Chaos-based image encryption supposedly provides a higher data rate and compute efficiency than traditional image encryption methods. In this paper, a simple chaotic system is used[1] as a pseudo-random number generator. A bitstream produced by the generator is then used to diffuse the pixel data of the plain image, resulting in a cipher image. The result of this research is a matlab and VHDL model that implements both the Vilnius oscillator and the image encryption process.

II. Methods

Chaotic sequence that is required for encryption is produced by Vilnius oscillator. When oscillators function crosses $Y = 0$, related X value is compared to an arbitrarily chosen threshold as in formula (1).

$$\begin{cases} 1 \text{ if } x < 60 \land y = 0 \\ 0 \text{ if } x \geq 60 \land y = 0 \end{cases} \tag{1}$$

Bitstream generation method is shown in Figure 1.

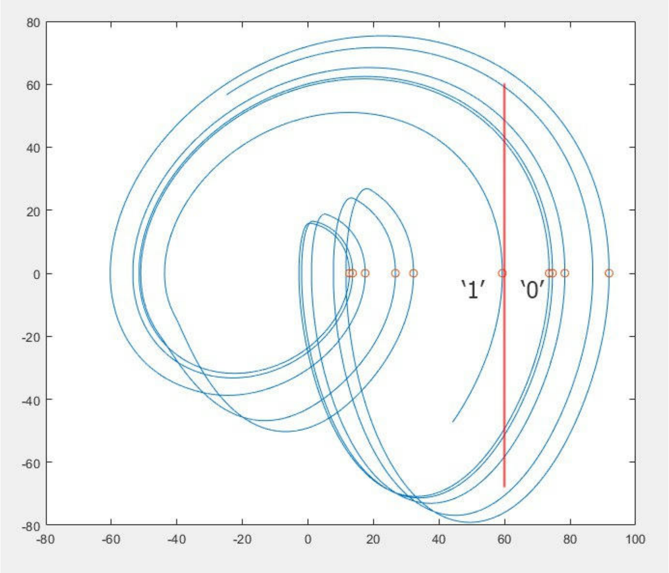

Figure 1: Bitstream generation method

A. Vilnius oscillator

Study depends on the Vilnius oscillator[1] as a chaos oscillator to generate PRNG sequences. Circuit diagram of the Vilnius oscillator is given in Figure 2.

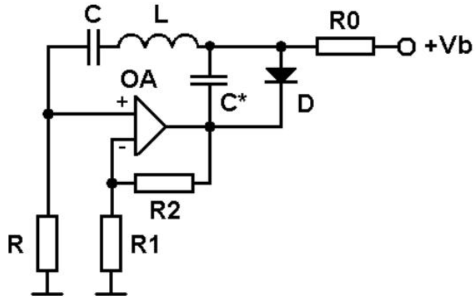

Figure 2: Vilnius oscillator circuit diagram [1]

A system of equations that defines the Vilnius oscillator is shown in (2) [1].

$$\begin{cases} C_1 \frac{\mathrm{d}V_{C_1}}{\mathrm{d}t} = I_L \\ L \frac{\mathrm{d}I_L}{\mathrm{d}t} = (k-1)RI_L - V_{C_1} - V_{C_2} \\ C_2 \frac{\mathrm{d}V_{C_2}}{\mathrm{d}t} = I_0 + I_L - I_D \end{cases} \tag{2}$$

Which is then presented in a more convinient form for simulation in (3) [1].

$$\begin{cases} \dot{x} = y \\ \dot{y} = ay - x - z \\ \varepsilon \dot{z} = b + y - c(\exp z - 1) \end{cases} \tag{3}$$

Constants used for simulation are following:

$$R_1 = 1 \cdot 10^3 \Omega \qquad R_2 = 10 \cdot 10^3 \Omega$$
$$R_3 = 6 \cdot 10^3 \Omega \qquad R_4 = 20 \cdot 10^3 \Omega$$
$$C_1 = 1 \cdot 10^{-9} F \quad C_2 = 150 \cdot 10^{-12} F$$
$$L = 1 \cdot 10^{-3} H$$

B. *FPGA simulation*

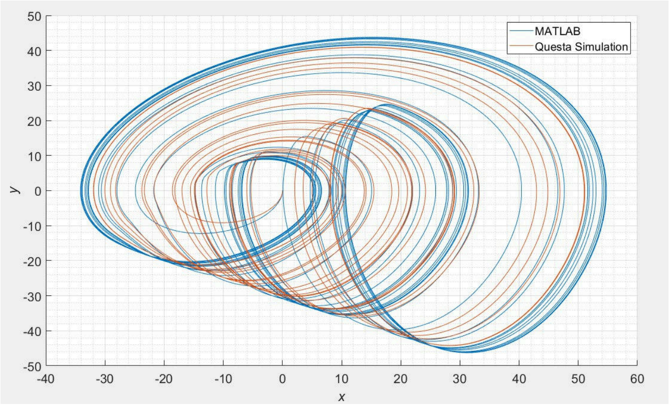

Figure 3: FPGA simulation compared to Matlab simulation

C. *Exponent approximation*

The need to approximate the exponent function on FPGA arises from the fact that the exponential function is computationally expensive and requires a large number of resources to compute accurately. Follwing is the representation of exponent approximation in matlab:

```
function value = exp_approx(x, N, x_min, x_max)
  x_step = (x_max-x_min)/N;
  value = 0;
  for i = 0:N-1
    x1 = x_min + i*x_step;
    x2 = x1 + x_step;
    if x >= x1
      y1 = exp(x1);
      y2 = exp(x2);
      value = (x-x1)/(x2-x1)*(y2-y1) + y1;
    end
  end
end
```

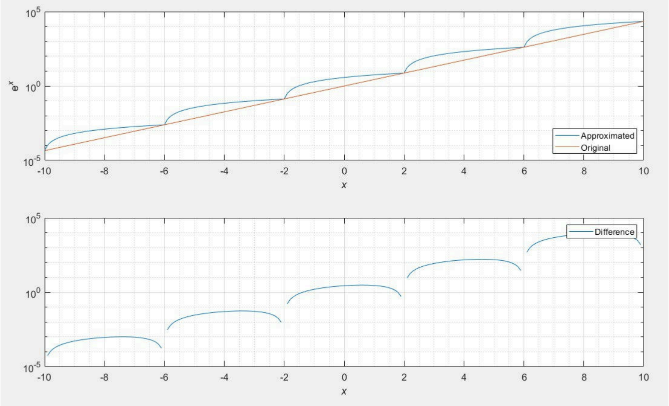

Figure 4: Difference between approximated exponent and e^x

III. Results

While generally the source images are encrypted, nonuniform distribution is clearly visible, especially in Figure 5, where the dark region has impact on the same region in the resulting image.

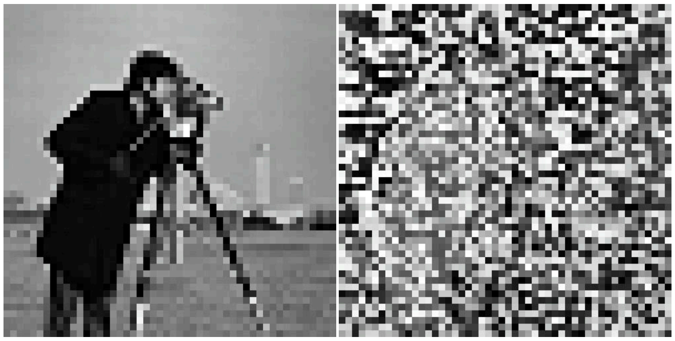

Figure 5: Cameraman: plain image, cipher image

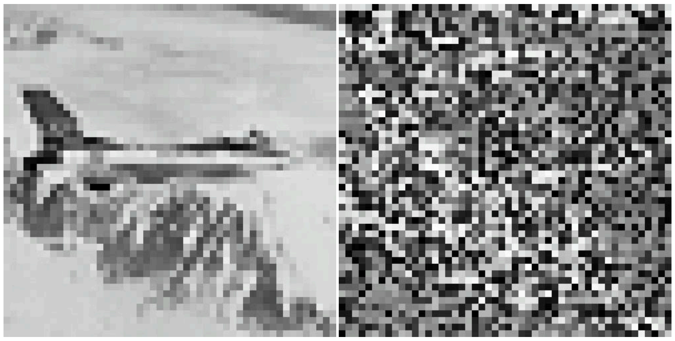

Figure 6: Jetplane: plain image, cipher image

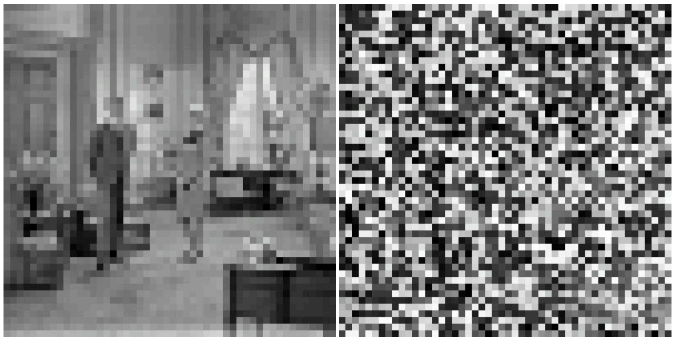

Figure 7: Livingroom: plain image, cipher image

IV. Conclusion

References

[1] A. Tamasevicius, G. Mykolaitis, K. Pyragas, and V. Pyragas, “A simple chaotic oscillator for educational purposes”, *European Journal of Physics*, vol. 26, pp. 61–63, 2004, doi: 10.1088/0143-0807/26/1/007.