MEIA SYSTEMS: MEMBRANE ENCRYPTED INFORMATION APPLICATIONS SYSTEMS

Nuria Gomez, Alberto Arteta, Luis Fernando Mingo

***Abstract**: Membrane computing is a recent area that belongs to natural computing. This field works on computational models based on nature's behavior to process the information. Recently, numerous models have been developed and implemented with this purpose. P-systems are the structures which have been defined, developed and implemented to simulate the behavior and the evolution of membrane systems which we find in nature. What we show in this paper is a new model that deals with encrypted information which provides security the membrane systems communication. Moreover we find non deterministic and random applications in nature that are suitable to MEIA systems. The inherent parallelism and non determinism make this applications perfect object to implement MEIA systems.*

***Keywords**: P-systems mapping, MEIA systems, membrane systems.*

Introduction

Natural computing is a new field within computer science which develops new computational models. These computational models can be divided into three major areas:.

- Neural networks.
- Genetic Algorithms
- Biomolecular computation.

Membrane computing is included in biomolecular computation. Within the field of membrane computing a new logical computational device appears: The P-system. These P-systems are able to simulate the behavior of the membranes on living cells. This behavior refers to the way membranes process information. (Absorbing nutrients, chemical reactions, dissolving, etc)

In this paper, we design a MEIA system just by explaining the process of encrypting the information that membrane systems process.

In order to do this we will take the following steps:

- Introduction to P-systems theory.
- Introduction to encryption algorithms
- Integration of the encryption with membrane systems
- Description of MEIA
- Applications of MEIA

Introduction to P-systems theory

I. A P-system is a computational model inspired by the way the living cells interact with each other through their membranes. The elements of the membranes are called objects. A region within a membrane can contain objects or other membranes. A p-system has an external membrane (also called skin membrane) and it also contains a hierarchical relation defined by the composition of the membranes. A multiset of objects is defined within a region (enclosed by a membrane). These multisets of objects show the number of objects existing within a region. Any object 'x' will be associated to a multiplicity which tells the number of times that 'x' is repeated in a region.

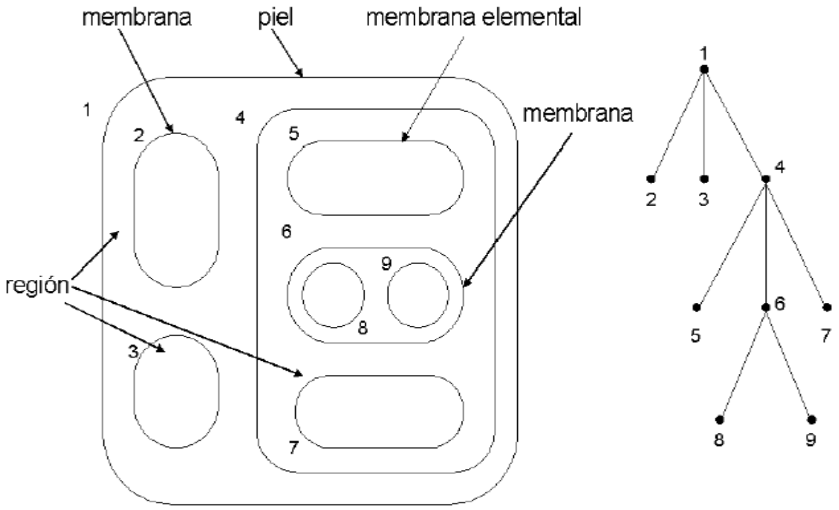

Fig. 1. The membrane's structure (left) represented in tree shape (right)

According to Păun 's definition, a transition P System of degree n, n > 1 is a construct: [Păun, 1998]

$$\Pi = \left(V, \mu, \omega_1, .., \omega_n, (R_{1,}\rho_1), ..(R_{n,}\rho_n), i_0\right)$$

Where:

1. *V* is an alphabet; its elements are called objects;
2. µ is a membrane structure of degree n, with the membranes and the regions labeled in a one-to-one manner with elements in a given set ; in this section we always use the labels 1,2,..,n;
3. $\omega_i \; 1 \le i \le n$, are strings from V^* representing multisets over V associated with the regions 1,2,..,n of µ;
4. $R_i \; 1 \le i \le n$, are finite set of evolution rules over V associated with the regions 1,2,..,n of µ; ρ_i is a partial order over $R_i \; 1 \le i \le n$, specifying a priority relation among rules of R_i . An evolution rule is a pair (u, v) which we will usually write in the form $u \to v$ where u is a string over V and v=v’ or v=v’ δ where v’ is a string over $(V \times \{here, out\}) \cup (V \times \{in_j \; 1 \le j \le n\})$, and δ is a special symbol not in. The length of u is called the radius of the rule $u \to v$
5. i_o is a number between 1 and n which specifies the output membrane of $\prod$

Let *U* be a finite and not an empty set of objects and N the set of natural numbers. A *multiset of objects* is defined as a mapping:

$$M : V \to \mathrm{N}$$
$$a_i \to u_1$$

Where a_i is an object and u_i its multiplicity.

As it is well known, there are several representations for multisets of objects.

$$M = \{(a_1, u_1), (a_2, u_2), (a_3, u_3)\ldots\} = a_1^{u_1} \cdot a_2^{u_2} \cdot a_n^{u_n} \ldots\ldots$$

Evolution rule with objects in *U* and targets in *T* is defined by $r = (m, c, \delta)$ where

$$m \in M(V), c \in M(VxT)\, and\ \delta \in \{to\ dissolve, not\ to\ dissolve\}$$

From now on *'c'* will be referred to as the consequent of the evolution rule *'r'*

The *set of evolution rules* with *objects* in *V* and targets in *T* is represented by R *(U, T).*

We represent a rule as:

$$x \to y \quad or \quad x \to y\delta$$

where x is a multiset of objects in M((*V*)xTar) where Tar ={here, in, out} and y is the consequent of the rule. When δ is equal to “dissolve”, then the membrane will be dissolved. This means that objects from a region will be placed within the region which contains the dissolved region. Also, the set of evolution rules included on the dissolved region will disappear.

P-systems evolve, which makes it change upon time; therefore it is a dynamic system. Every time that there is a change on the p-system we will say that the P-system is in a new transition. The step from one transition to another one will be referred to as an evolutionary step, and the set of all evolutionary steps will be named computation. Processes within the p-system will be acting in a massively parallel and non-deterministic manner. (Similar to the way the living cells process and combine information).

We will say that the computation has been successful if:

- The halt status is reached.
- No more evolution rules can be applied.
- Skin membrane still exists after the computation finishes.

Encryption algorithms

Most weak points are reached when there is data exchange in any communication process. The methods to exchange information are targets for attacks. It is normal that hackers trying to break into these methods to get the data. The amount of risk responds to an equation. This equation determines our decisions to take when protecting our system. B=P X L

- B: is the load or expenses that we invest in prevention of an specific loss due to a vulnerability.
- P: is the probability that such vulnerability is affected and that specific loss happens.
- L: is the impact or total cost that means the specific loss due to the vulnerability that has been affected.

If the value B<= P+L we need to set up a security system to prevent attacks otherwise it won't be needed. However this is an equation that not always suits reality. It can be an orientation but in the end common sense will apply. The equation gives us an idea about how much protection we want to set up in our system.

Creating an access control list can be interesting too. That way we can set up rights for users and establish the way the access to the system. As we can see in the slide, we can customize the rights.

Not only intentional threat can put our data in danger. Nature can give us a hard time. Natural disasters are often responsible for data loss. It is important put to allocate sensitive data in a safe place.

Information has been defined in many ways. In order to encrypt information we need to know some concepts of number theory. Congruency is an important operation used in info encryption.

Apart from congruency we are interested in finding numbers with divisibility properties. These are really important in cryptography. In particular, numbers that are only divided by either number 1 or by themselves are widely used (Prime numbers). In cryptography, given 2 numbers we might need to calculate the great common divisor

Inverses in a field (Field is referred as a number). This is a major concept to understand how public/private key systems (asymmetric systems).

Important fact:

- If n the field is prime, then any number has an inverse in the field.
- If the n is not prime, It might happen than a number doesn't have inverse in the field. (That is why in cryptography we will work with (n prime numbers).

The concept RSR is important as this let us have the prime numbers within a field 'n'. Some asymmetric systems such as RSA use it.

This operation has a high computational cost. That is why algorithms of rapid exponentiation have been created. The slides show a trick to reduce the computational operations

For any type of system we can find the most popular algorithms to encrypt information.

(DES, RSA, IDEA, R5......)

Public keys involve the use of one way functions. These functions are easy to calculate. However obtaining the inverse of the function is very difficult.

For instance the function that multiplies 2 big prime numbers is easy to implement.

Example: The product 6399999 and 89558745 is easy to calculate with a computer (573175878441255). However if we have this number, obtaining the 2 numbers that multiplied by each other is equal to it, (573175878441255) involves a high computational cost. This is the concept behind the public key cipher

The most common trapdoor functions are product of two big primes and the problem of the discrete logarithm.

Integration with Transition P-systems

This section explains the encryption protocol used by the membrane systems.

We can encrypt words by selecting its corresponding bit sequence. For example Let's the imagine that we have the membrane 'A' and we want to encrypt the info end to membrane B. Imagine the info is the code 'A.' The ASCII code for A is 65 and representation for 65 is 1000001.

Then we will be encrypting bit by bit.

In order to encrypt bit a bit we have to:

1) Convert the text we want to encrypt into binary code (0's and 1's). (in the end the info are sequences of 0 and 1);
2) Selecting a random number in binary code: Example 0010001;
3) Apply the XOR function to 1) and 2) i.e.
   0 XOR 0=0
   0 XOR 1=1
   1 XOR 0=1
   1 XOR 1=0

If we want to encrypt the text: "xxaaci" which is the standard information shared in every membrane.

We could divide the text into data blocks and then encrypt those.

Considering that a character has 1 byte size, we would need 2 data blocks to encrypt the message.

Then the algorithm encrypts every block.

We might use a secret key for encrypting every block. In the next slides we show advantages and disadvantages of secret key management blocks

In order to prevent the problems of using private keys, we can use a combination of the two (public/private) keys.

This example shows an algorithm that encrypts with only private keys.

MEMBRANE A only knows his private key. So Only MEMBRANE A can encrypt /decrypt a message.

MEMBRANE B only knows its private key. Only MEMBRANE B can encrypt /decrypt with that key.

The problem of the integrity arises. Anyone can encrypt with a different private key. MEMBRANE A can never be sure that B is really MEMBRANE B or someone else. Anyone can intercept the message in the system and then encrypt it with their own key. That is why the use of public keys is needed.

Here it explains how someone can steal the identity in the membrane communication phase.

In this case if B encrypts a message with a private key and provides the public key to A, if A receives the message she knows for sure that B is the real B and not someone else. However anyone can obtain the public key and decrypt the message.

Now we use a combination of public and private keys.

A has a private key (PRa) and gives B their public one (PUa) (Anyone can have the public one).

B has a private key (PRb) and gives A their public one (PUb) (Anyone can have the public one)

1st) B performs a first encrypting operation to the info processed with the public key of A (PUa) so only A can decrypt it because it is required the private key of A(PRa) because only A knows it.

Let's say that the obtained cryptogram is X.

2st) B performs another encryption (now over X) and he does it with his private key (PRb) and send the cryptogram through a channel (this is used to start the session of communication. In this case anyone who has (PUb) can intercept it and decrypt it but the maximum they could do is getting X (which is a cryptogram itself encrypted with PUa) which means that it is necessary to have also (PRa) In this case A is the only MEMBRANE who can decrypt it because they have PuB and PuA.

In this case A is sure that the message can from B because B use PRb and only B knows it.

Also we make sure that no one can decrypt the message but A because PRa is required

3st) By doing this we make sure no one can steal A and B identity (integrity) and no one can intercept the message (confidentiality).

The 2 main characteristics in information security are preserved.

Randomly dimensioned applications

The concept of MEIA arises when applying the encrypted membrane systems to nature random applications. Thus it makes sense to apply these encrypted systems in security areas that must have non known size and can be often redimensioned.

Several applications as E-grid, diseases patterns, raid clustering, etc can use a MEIA system to protect data and speed up problems solving.

Conclusion

In this paper, we have studied some topics of membrane computing. As a part of this study, we have explained some concepts of the p-systems. Concepts such as:

- Components;
- Interactions between the components;
- The evolution of a p-system;
- Encriyption information;
- Applications of encrypted membrane systems (MEIA systems).

Nowadays we work with the p-system as an entire compacted block of components that are going through an evolutionary process. The p-system functioning is treated under a global perspective. The intregration qith application of nature can empower much more the use of p-systems in the future.

Besides, security is a must when using technology.

Bibliography

[Păun, 1998] "Computing with Membranes", Journal of Computer and System Sciences, 61(2000), and Turku Center of Computer Science-TUCS Report nº 208, 1998.

[A. Arteta, 2008] "Algorithm for Application of Evolution Rules based on linear diofantic equations" Synasc 2008, Timisoara Romania September 2008[1] A. Syropoulos, E.G. Mamatas, P.C. Allilones, K.T. Sotiriades "A

[Arroyo, 2001] "Structures and Bio-language to Simulate Transition P Systems on Digital Computers," Multiset Processing ([Arroyo, 2003] "A Software Simulation of Transition P Systems in Haskell, Membrane Computing,"

Authors' Information

***Alberto Arteta Albert** – Associate professor U.P.M Crtra Valencia km 7, Madrid-28031, Spain; e-mail: aarteta@eui.upm.es*

Research: Membrane computing, Education on Applied Mathematics and Informatics

***Nuria Gomez**– Associate professor U.P.M, Crtra Valencia km 7, Madrid-28031, Spain; e-mail: ngomez@eui.upm.es*

Research: Membrane computing, Education on Informatics

***Luis Fernando Mingo** –Associate Professor, Crtra Valencia km 7, Madrid-28031, Spain e-mail: lfmingo@eui.upm.es*

PHD on Artificial Intelligence, Education on Mathematics and Informatics