IJCSS International Journal of Complex Systems in Science

Int. J. Complex Systems in Science
vol. **3(1)** (2013), pp. 45–54

Towards a Complex Networks' based Model for Intentional Technological Risk Analysis

Santiago Moral[1,4,†], Victor Chapela[2], Regino Criado[5,6], Ángel Pérez[3,4] and Miguel Romance[5,6]

[1] *IT Risk, Fraud and Security - Research and Innovation for IT Risk, Fraud and Security, BBVA, Spain*

[2] *Sm4rt, Security Services, Mexico*

[3] *Innovation for Security (I4S), Madrid, Spain*

[4] *Centro de Investigación para la Gestión Tecnológica del Riesgo , Spain*

[5] *Department of Applied Mathematics, Rey Juan Carlos University, Spain*

[6] *Center for Biomedical Technology, Technical University of Madrid, Spain*

Abstract. Some ideas about a model for managing the risk of suffering an intentional attack in digital environments are presented in order to get a new model based on complex networks. This model will allow us to analyze and mitigate the intentional technological risk on digital networks and environments. Within this context, we specifically introduce a complete description of a new algorithm called $max\text{-}path$ devoted to assign a specific value to each component of the network (nodes and edges) by distributing the initial value of information assets which are located in areas (named vaults) throughout the whole network.

Keywords: complex network, intentional complex network, intentional technological risk
MSC 2000: Primary: 05C90, 05C75; Secondary: 68M10, 94C15, 90B18.

† **Corresponding author:** santiago.moral@bbva.com
Received: December 5th, 2013
Published: December 31th, 2013

1. Introduction

A fundamental challenge in network science is to extract a solid foundation and a set of key principles for networked systems from the widely dispersed efforts in analyzing real-world networks and developing mathematical models of networks. Following this guideline, complex networks have been used for modeling different real world systems, such as Internet, neural and social networks, the World Wide Web and many other examples [1, 2, 3, 4, 5, 6].

On the other hand, as technology is developing so rapidly and cyber-attacks are becoming more and more sophisticated, the challenge faced by risk managers is how to manage this increasingly complex digital risk environment having in mind that there is not any standard for measuring intentional technological risk.

Our main proposal is to build a whole mathematical model based on the complex networks' theory and its tools for a better understanding, managing and identification of solutions to prevent and mitigate intentional technological risk of suffering an intentional attack in digital environments.

Up to now, the risk of suffering an intentional attack within digital environments had been mainly studied within the framework of Game Theory. In our model we use an adapted functional complex network that allow us to get a qualitative and quantitative information about the risk of suffering an intentional attack within digital environments.

In our model we assign three attributes to each component (nodes and edges) of a digital network: its value, its anonymity and its accessibility in order to include intentionality within a framework which let us establish a tool for measuring intentional technological risk.

It is remarkable that within our framework we identify two kinds of intentional technological risk: Static Risk (the attacker only use the authorized routes) and Dynamic Risk (the attacker manipulates and modifies the routes and the system to access at his target).

The value of an element or component is, within this context, the value for the defender. It mainly depends on its place within the network, the links to the rest of nodes, the distance from the vaults and the total value placed in the vaults. Anonymity is related to the possible identification of the users who reach that element. The accessibility of an element may be quantified by using a slight variation of the PageRank algorithm which assigns a quantity to every element of the network (its centrality) that allows to sort them in a ranking.

Therefore several algorithms are needed to get an accurate model looking at its applicability in real scenarios, for example, to locate and optimize the places where we must put controls in order to manage intentional technological

risk. One of these algorithms, the algorithm *max-path*, is devoted to assign a specific value to each component(nodes and edges) of the network by spreading and distributing the value of the information laying within the vaults throughout the rest of the edges and nodes of the whole network. It is remarkable that this algorithm may be of interest in other contexts. In any case it is important to highlight that directness must be taken into account in our model since the kind of networks we study present asymmetric interactions. After this introduction, section 2. is devoted to introduce a mathematical framework for risk management, including the notation and some definitions we use in our work. In section 3. we describe in detail the algorithm *max-path* intended to assign a specific value to each component of the network. In section 4. we provide a pseudo-code version of the algorithm, and finally section 5. is devoted to present some conclusions.

2. A mathematical framework for risk management

From a schematic point of view, a (directed) complex network is a mathematical object $\vec{G} = (V, E)$ composed by a set of nodes or vertices $V = \{v_1 \ldots, v_n\}$ that are pairwise joined by oriented links or edges $E = \{\ell_1, \ldots, \ell_m\}$. The adjacency matrix of a directed network $\vec{G}$, denoted by $A_{\vec{G}}$, is the $n \times n$ dimensional $(0, 1)$-matrix $A_{\vec{G}} = (a_{ij})$ determined by the conditions:

$$a_{ij} = \begin{cases} 1 & \text{if} \quad (v_i, v_j) \in E \\ 0 & \text{if} \quad (v_i, v_j) \notin E. \end{cases}$$

In order to adapt this concept to the environment of digital networks, we will call *intentional network* to a tuple $(\vec{G}, \psi, \alpha, \beta)$, composed by

1. A directed graph $\vec{G} = (V, E)$, where V is the set of *nodes* and $E \subseteq V \times V$ is the set of *edges*.
2. A map $\psi : E \longrightarrow [0, 1] \subseteq \mathbb{R}$, called *value reduction*.
3. A map $\alpha : E \longrightarrow \mathbb{N} \times [0, 1] \times (\mathbb{R}^+ \cup \{0\})$, called *edge attributes*, and defined as $\alpha = (\overline{Anon}, \overline{Acc}, \overline{Val})$, where
   (a) $\overline{Anon} : E \longrightarrow \mathbb{N}$ is the *edge anonymity*.
   (b) $\overline{Acc} : E \longrightarrow [0, 1] \subseteq \mathbb{R}$ is the *edge accesibility*.
   (c) $\overline{Val} : E \longrightarrow \mathbb{R}^+ \cup \{0\}$ is the *edge value*.
4. A map $\beta : V \longrightarrow [0, 1] \times \mathbb{R}$, called *node attributes*, and defined as $\beta = (Acc, Val)$, where

(a) $Acc : V \longrightarrow [0,1] \subseteq \mathbb{R}$ is the *node accesibility.*

(b) $Val : V \longrightarrow \mathbb{R}^+ \cup \{0\}$ is the *node value.*

Now we will describe how the value will be spread throughout the network using a value reduction matrix:

Given a directed graph $\vec{G} = (V, E)$, where $V = \{v_1, \ldots, v_n\}$, and a map $\psi : E \longrightarrow [0,1] \subseteq \mathbb{R}$, we call *value reduction matrix* of $\vec{G}$ to the $n \times n$ matrix B, which has as inputs

$$B(i,j) = \begin{cases} \psi((v_i, v_j)) & \text{if} \quad (v_i, v_j) \in E \\ 0 \quad \text{if} & (v_i, v_j) \notin E. \end{cases}$$

3. The max-path algorithm

In the sequel, given a vector $x \in \mathbb{R}^n$ we denote by $x[j]$ the j-th coordinate of x. The following is a description of the *max-path* algorithm:

Max-path algorithm

Input:

- A directed network $\vec{G} = (V, E)$, where $V = \{v_1, \ldots, v_n\}$.
- A value reduction map $\psi : E \longrightarrow [0,1] \subseteq \mathbb{R}$.
- An initial node *vault* $v_i \in V$.
- An initial value $M \in \mathbb{R}^+$.
- A stop condition and/or a maximum number Max of iterations.

Output:

- A map of the *dispersed value from the node* v_i

$$Val_i : V \longrightarrow \mathbb{R}$$

Step 1 (initial vectors): We consider an initial vector $x_0 \in \mathbb{R}^n$ with M value only in the position that corresponds to the node v_i, that is

$$x_0[j] = \begin{cases} M & \text{if} \quad j = i \\ 0 & \text{if} \quad j \neq i. \end{cases}$$

We define as well $y_0 = x_0$.

Step 2 (k-iteration): Given the vectors $x_{k-1}, y_{k-1} \in \mathbb{R}^n$ we build the vectors x_k, y_k as follows:

1. Given the vector y_{k-1}, we decompose it as a sum of vectors that have a single coordinate different from zero:

$$y_{k-1} = \sum_{y_{k-1}[j] \neq 0} y_{k-1}[j] \cdot e_j = \sum_{y_{k-1}[j] \neq 0} u_{k-1}^{(j)}$$

   where e_j is the j-th vector of the canonical basis.

2. We multiply each of those vectors by the reduction matrix B:

$$u_k^{(j)} = B \cdot u_{k-1}^{(j)}$$

3. We settle y_k as the maximum, coordinate by coordinate, of the results:

$$y_k[j] = max_l\{u_k^{(l)}[j]\}$$

   for each $j = 1, \ldots, n$.

4. In order to build x_k, we take maximums in every coordinate:

$$x_k[j] = max\{x_{k-1}[j], y_k[j]\}$$

   for each $j = 1, \ldots, n$.

Step 3 (output building): When the maximum number of iterations Max is reached, or the stop condition is fulfilled, no more iterations will be carried out. We denote by x_K the last obtained vector. We set

$$Val_i(v_j) = x_K[j]$$

for every $j = 1, \ldots, n$.

It is remarkable that it has been found that a stop condition that guarantees that the algorithm finishes in a finite number of steps and that the value has been transmitted to all the nodes of the graph is the following:

"when $x_{k-1} = x_k$ we stop the algorithm"

Now, as it is possible that there exists more than one single vault, it is needed to consider the following:

Algorithm of value assignation

Input:

- A directed network $\vec{G} = (V, E)$, where $V = \{v_1, \ldots, v_n\}$.

- A reduction value map $\psi : E \longrightarrow [0,1] \subseteq \mathbb{R}$.
- A subset of nodes with initial value (vaults) $\{v_{i_1}, \ldots, v_{i_l}\}$.
- A set of values associated to the previously mentioned nodes $\{M_{i_1}, \ldots, M_{i_l}\}$
- A stop condition and/or a maximum number Max of iteration for the algorithm *max-path*.

Output:

- A value of edges map $\overline{Val} : E \longrightarrow \mathbb{R}$
- A value of nodes map $Val : V \longrightarrow \mathbb{R}$

Step 1 (dispersion of the value from each vault): For each pair (v_{i_j}, M_{i_j}), we execute the algorithm "max-path" with the corresponding inputs, obtaining $Val_{v_{i_j}}$, the function of dispersed value from the node v_{i_j}.

Step 2 (definition of Val): For every $v \in V$ we define

$$Val(v) = \sum_{j=1}^{l} Val_{v_{i_j}}(v)$$

that is, we sum the dispersed values associated to each vault.

Step 3 (definition of $\overline{Val}$): For every edge $(v, w) \in E$, we define

$$\overline{Val}((v,w)) = Val(w) \cdot \psi((v,w))$$

the value of the destination node multiplied by the reduction of the edge value.

4. Coding the algorithm

In short, a formal and brief description of the algorithm *max-path* to be implemented is the following:

- Assuming that we are distributing a value M, we consider an initial vector v_0 with all of its coordinates are equal to zero except for in the corresponding position to the vault in which the value M appears. We also consider $w_0 := v_0$.
- In general, the components of the vector w_k keep the value that has circulated throughout the network during the iterative step k, while the vector v_k will keep the total value accumulated after step k.

- In the iterative step k, given the vector w_{k-1}, we decompose it as a sum of vectors that will have a single coordinate different from zero:

$$w_{k-1} = \sum_{w_{k-1}[j] \neq 0} u_{k-1}^{(j)}$$

- We multiply each of those vectors by the reduction matrix B:

$$u_k^{(j)} = B \cdot u_{k-1}^{(j)}$$

- Remark for implementation: The calculation of the vector $u_k^{(j)}$ may be done by multiplying the number $w_{k-1}[j]$ by the j-column of the B matrix. This way, there is no need to make the explicit decomposition, it is enough to consider as many vectors u_k as coordinates different from zero that the vector w_{k-1} has.

- We build w_k as the maximum, coordinate by coordinate, of the results:

$$w_k = max\_co_{w_{k-1}[j] \neq 0}(u_k^{(j)})$$

- We build v_k as the maximum, taken coordinate by coordinate, of v_{k-1} and w_k, i.e.

$$v_k := max\_co(w_k, v_{k-1})$$

- Stop condition: In the moment in which $v_{k-1} = v_k$ the algorithm stops. The coordinates of the vector v_k provide the distributed value from the vaults to each of the nodes.

Next we provide pseudo-code description of the algorithm.

Input:

- `matrix B /* The value reduction matrix */`
- `real M /* The value to transmit */`
- `integer j0 /* The position of the vault, the node with value, in the node list */`
- `integer size /* The size of the list of nodes (and the matrix B) */`

Variables:

- `vector v /* To store the accumulated value until the last step */`
- `vector v_prev /* To store the vector v from the previous step */`
- `vector w /* To store the value transmitted in the last step */`
- `vector w_prev /* To store the vector w form the previous step */`
- `vector u /* auxiliar vector */`

Functions:

- `vector col(matrix B, integer j) /* Outputs the vector which is the j-th column of B */`
- `vector max_co(vector u, vector v) /* Outputs the vector which results from taking the maximum of u and v in each coordinate */`

Pseudo code:

```
/* Initialize the vectors v and v_prev.  The vector v_prev is initialized
to 0 to make it different from v and let the condition for the external
loop true for the first iteration.  Whenever a vector is initialized
to 0, this means that every coordinate is set to 0 */

v_prev := 0;
v := 0;
v[j0] := M; /* This stores the initial value M in the appropriate
coordiante */

/* Initialize the vectors w and w_prev to 0 */

w_prev := 0;
w := 0;

/* The external loop will execute until two consecutives iterations
give the same accumulated value vector.  This should always end, because
it should be executed at most so many times as the longest path in
the graph (without duplicated edges) */
```

```
while (v<>v_prev) do

/* Store the vectors from previous iteration */
v_prev := v;
w_prev := w;

/* Set the vector w to 0 */
w := 0;

/* Checks the coordinates of w_prev for values different to 0.  If
true, computes the transmitted value from the j-th node, by multiplying
the j-th column of B by the j-th coordinate of w_prev.  Then this
is stored in w by using the funtion max_co */

for j:= 1 to size do
     if (w_prev[j]>0) then
        u := w_prev[j]*col(B,j); /* to multiply a vector by a
        number, multiply each component by the number */
        w := max_co(u,w);
     end if;
end for;

/* Actualize vector v */
v := max_co(w,v);

end while;
```

5. Conclusions

Intentional technological risk can be described and analyzed by using the complex networks' theory and its tools since the structural properties of certain functional network provide valuable information to prevent and mitigate the intentional technological risk on digital networks. A useful algorithm related to the vaults values distribution throughout a digital network has been introduced in this context. This algorithm may be of interest in other contexts.

Acknowledgements

The authors thank BBVA IT Risk, Fraud and Security - Research and Innovation for IT Risk, Fraud and Security, Sm4rt Security Services, Innovation for Security(I4S), Centro de Investigación para la Gestión Tecnológica del Riesgo

(CIGTR) and University Rey Juan Carlos for their support and facilities. We also want to thank Francisco García, Rosa María Quintanar, Elena Alonso and Luis Solá for their useful comments and suggestions. This paper was partially supported by FEDER Funds MTM2009-13848.

References

[1] Y. Bar-Yam, *Dynamics of Complex Systems*, Addison-Wesley, 1997.

[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.-U. Hwang,*Physics Reports* **424**, 175–308 (2006).

[3] R. Criado, J. Flores, A. García, J. Gómez and M. Romance, *IJCM* **89,3**, 291–312 (2012).

[4] R. Albert and A. L. Barabási, *Rev. Mod. Phys.* **74**, 47–97 (2002).

[5] M. E. J. Newman, *SIAM Review* **45**, 167–256 (2003).

[6] S. H. Strogatz, *Nature* **410**, 268–276 (2001).