Financial Critical Infrastructure: A MAS Trusted Architecture for Alert Detection and Authenticated Transactions

Guemkam, Guy; Feltus, Christophe; Bonhomme, Cédric; Schmitt, Pierre; Gâteau, Benjamin; Khadraoui, Djamel; Guessoum, Zahia

Published in:
Proceeding of the Sixth IEEE Conference on Network Architecture and Information System Security (SAR/SSI 2011), La Rochelle, France

DOI:
10.1109/SAR-SSI.2011.5931359

Publication date:
2011

Document Version
Early version, also known as pre-print

Financial Critical Infrastructure: A MAS Trusted Architecture for Alert Detection and Authenticated Transactions

Guy Guemkam (Guy.Guemkam@tudor.lu)*
Christophe Feltus (Christophe.Feltus@tudor.lu)*
Cédric Bonhomme (Cedric.Bonhomme@tudor.lu)*
Pierre Schmitt (Pierre.Schmitt@tudor.lu)*
Benjamin Gâteau (Benjamin.Gateau@tudor.lu)*
Djamel Khadraoui (Djamel.Khadraoui@tudor.lu)*
Zahia Guessoum (Zahia.Guessoum@lip6.fr)†

Abstract: Banking and financial system constitute a critical service for our economy. Being able to foreseen real time attacks and failures on financial institutions network is a crucial challenge. In order to face those problems, research tends to elaborate alert mechanisms in order to support risk assessment, analysis and mitigation solution for real time decision-making processes. Our previous works had proposed a multi-agent based architecture to support that alert mechanism. This architecture exploited the Multi-agent system technology and proposes a static assignment of functions to agents. This static assignment was a weakness because isolating an agent or breaking the communication channel between two of them created serious damage on the crisis management. In this paper, we complete our previous works and make mobile the assignment of functions to agents. Our approach exploits the concept of agent responsibility that we assigned dynamically to agent taking into consideration the agent trust. This mobile assignement is illustrated by a case study that aims at adopting agent's secret key according to the sensibility of the financial context.

Keywords: trust model; responsibility; multi-agent system; bank clearing; financial critical infrastructure

1 Introduction

Infrastructures are the basic physical and organizational structures needed for the functioning of a community or society, such as banking and financial systems. These Critical Financial Infrastructures (e.g. payment systems and clearing houses) are characterized by being so vital to a country that the incapacity or destruction of such systems would have a debilitating impact on security, national economic security, national public safety, or a combination of these [Rin04]. In fact, the financial critical infrastructure (FCI) are so important that a significant effort in making sure that they are not damaged or destroyed justifies the uprising research in that field. Additionally, managing the security

* Public Research Centre Henri Tudor, Luxembourg-Kirchberg, Luxembourg
† Laboratoire IP6, Université Pierre et Marie Curie, Paris, France

of FCI systems is a difficult task offering numerous research challenges because they are traditionally characterized by (see Fig. 1):

- deployed on large and complex inter-institutions network, making it difficult to cover all potential vulnerabilities which can be exploited to launch cyber-attacks against them;
- a very strong availability requirement, i.e. FCI systems can rarely be taken down. This can make them hard to maintain, and consequently many critical infrastructure systems run on out-dated technology which may not have been geared to cope with the threat picture of today;
- a major request concerns the confidentiality of the message, i.e. bank secrecy is extremely significant for the private operations, and for the state transactions as well. Continuously monitoring and adapting Public Key Infrastructure (PKI) is mandatory.

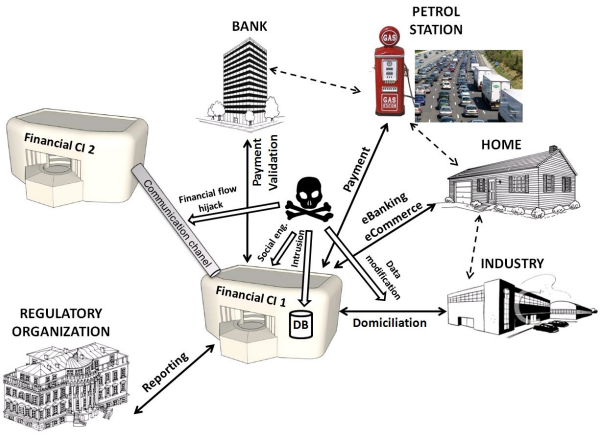

Fig. 1: FCI security challenge

Many critical infrastructure systems including FCI are exposed to the Internet, making cyber attacks a very real problem. According to a recent study on critical infrastructure cyber attacks [BWIM09], "*critical infrastructure owners and operators report that their IT networks are under repeated cyber attack, often by high-level adversaries. The impacts of such attacks are often severe, and their cost is high and borne widely*".

The situation of the FCI is that from the IT side, conceptual security is achieved via security technology watch in order to manage new vulnerabilities and avoiding new threats by the introduction of adapted security counter measures into the operational critical elements of the infrastructure. Associated to this, the security controls are also frequently deployed (mainly via audits) on top of the two previous actions, in order to measure the confidence that they have on the security in place. The ultimate goal here is to guarantee that the required controls are properly in place and that the security counter measures and PKI regulation are adapted to the risk situations. Beside this, in certain cases, these

critical infrastructures are interdependent (clearing processing and data exchanges) and therefore additional complex security risks are to be considered. From the business side, the FCI owners are more and more subject to specific regulations requiring them to maintain an assessment and ongoing management of systemic risks by the national financial supervision institutions [Act02, Bas06]. A systemic risk, as defined by the European parliament, is a risk of disruption in the financial system with the potential to have serious negative consequences for the internal market and the real economy. All types of financial intermediaries, markets and infrastructure may be potentially systemically important.

Our state of the art in this field [FKA10] has highlighted that many architectures has already been elaborated to support the reaction after problem detection. Those architectures are mostly elaborated based on a multi-agent system (MAS) approach that offers the possibility to work in a decentralized and heterogeneous environment. However, we have observed that those architectures are based on a static assignment of functions to agents. As a consequence, isolating an agent or breaking the communication channel between two of them could create serious damage on the management of the crisis. In this paper, we propose an innovative approach for making dynamic the assignment of functions to agents. Our approach exploits the concept of agent responsibility that we dynamically assign depending on the crisis situation. Additionally, the crisis context makes vary the agent ability to react (e.g. agent momentary isolated or decreased capability to quickly react). In order to consider that constraints, we introduce the agent trust based on reputation as a parameter that influence the assignment.

Section 2 depicts the reaction mechanism case study. Section 3 presents the responsibilities model and section 4 the responsibility to agent assignment function. Section 5 introduces the trust and reputation calculus method and section 6 the dynamic assignment of responsibilities to agents based on the reputation.

2 Reaction mechanism case study

The agents that compose the architecture are the following:

- The ACE collects, aggregates and analyses network informations coming from probes deployed over the network and customer node. Confirmed alerts are sent to the Policy Instantiation Engine (PIE);

- The PIE receives confirmed alert from the ACE, sets the severity level and the extent of the network response (depending on the alert layer). The PIE instantiates high level alert messages which have to be deployed. Finally the high level alert messages are transferred to the Reaction Deployment Point (RDP);

- The RDP (Fig. 3) is composed of two modules. The Cryptography Analysis (CA) is in charge of analyzing the keys previously instantiated by the PIE. Therefore, the Crypto Status stores required properties for a secure asymetric key so that the CA module is able to check the eligibilty of the newly received key to be deployed. Concretely the CA will check the key strength, if the key has not been used or revoked, test if the cryptographic key is not badly generated (modulus-facotrization, etc.). The second module, the Component Configuration Mapper, selects the appropriate communication channel.

The reaction mechanism (Fig. 2) aims at adapting the ciffering of the data according to the bank interface whenever on alert occurs. This section presents the core components of the reaction mechanism. Agents are disseminated on three layers corresponding to the clearing mechanism[1] (customer/issuer, acquiring/issuing processing) and they retrieve information from probes located at the network layer and representing different values: network traffic, DoS attack (denial-of-service attack, an attempt to make a computer resource unavailable to its intended users), suspicious connections attempts, ...

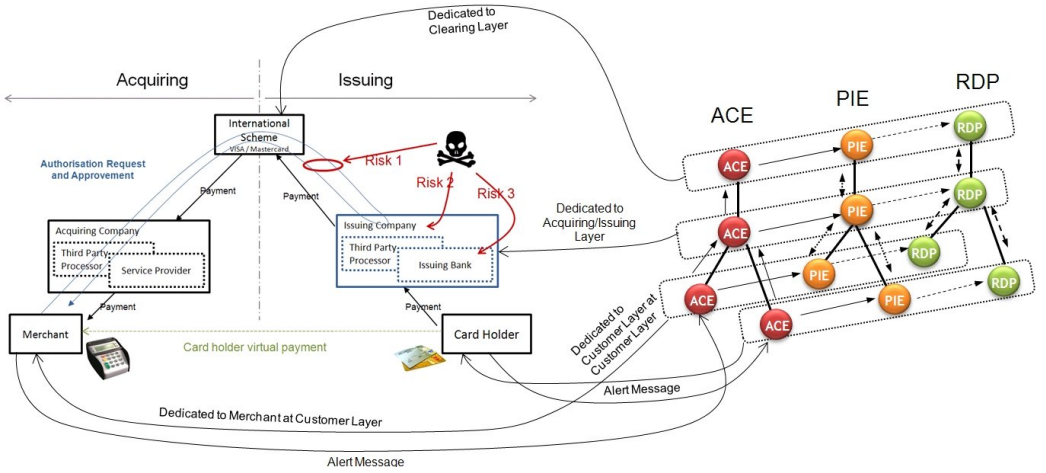

Fig. 2: Reaction mechanism inside

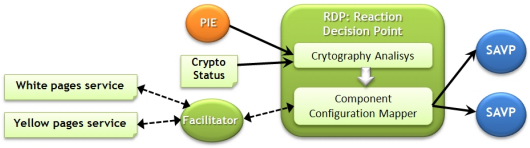

Fig. 3: RDP architecture

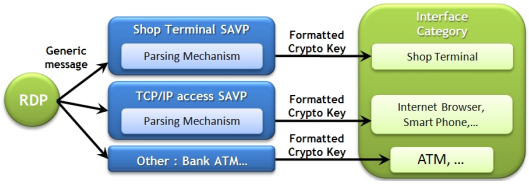

Fig. 4: SAVP architecture

Fig. 4 presents two different kinds of Signature Agent Verification Point (SAVP). Indeed, another advantage of MAS is that it is easy to implement from a model, specific

[1] Clearing denotes all activities from the time a commitment is made for a transaction until is settled.

agents in order to performs specific tasks. Concretely it is adapted to the usage and practical deployment in that it coincides (e.g. shop terminal, smartphone, ATM, etc.) and sends alerts to appropriate level (cutsomer, processing, clearing). Moreover, our attack prevention system is easily extensible for future interface category. SAVPs receive generic alert messages from the RDP. Then a specific parser converts the incoming key type to the appropriate format according to the interface.

In order to consider the mutual trust between agents, each agent is associated to a database of levels of trust towards its pairs. This means e.g. that the SAVP has a dedicated level of trust for the ACE and the RDP. This database can be accessed by $\mathcal{U}$TOPIA [SBAG11], the MAS framework we use to control the assignment of responsibility in an intelligent manner by taking the level of trust between agents into account. The reaction alert architecture presented in this section is based on the ReD (Reaction after Detection) project [FKA10]. The ReD project defines and designs a solution to enhance the detection/reaction process and improves the overall resilience of critical infrastructures. Fig. 5 introduces the developed architecture illustrated with our clearing network alert system. The flow is supposed to begin with an alert detected by a probe (connected to the SAVP at the bottom of the figure).

This alert is sent to the ACE agent (customer layer) that does or does not confirm the alert to the PIE. Afterwards, the PIE decides to adapt key or to forward the alert to an ACE of a higher layer (acquiring/issuring). The PIE agent sends key policy to the RDP agent, which decides which SAVP is able to transform the high level alert message into an understandable format for the selected interface category. In order to manage access rights, we have incorporated to ReD a Context Rights Management module (CRM). Block on the right on Fig. 5. The CRM is in charge of providing access rights to agents (E.g. SAVP to the probes and Logs File database, RDP to the keys Status database). The CRM uses the rights and the crisis context database and is able to detect changes in the situation thanks to $\mathcal{U}$TOPIA functions. The first database includes the access rights corresponding to the responsibilities and the second database includes a set of crisis contexts. Thanks to these databases the CRM agent is able to provide adequate rights at operational layer to an agent depending on the context.

3 Agent Responsibility Model

In a non-crisis context, agents are assigned to responsibilities like RDP, PIE, SAVP, etc. Those assignments of responsibilities to agents require specific access rights. By analyzing for instance the activity of monitoring the bank Internet access (Fig. 5), we can e.g. observe that the SAVP's have the responsibility to collect the number of connection from different IP adresses (to prevent DoS attack) or the location of the connection, to make a basic correlation between the values and the antecedent values records (stocked in the Logs File database) and to report this analysis to the ACE in case of suspected alert. To perform the monitoring activity, the SAVP are assigned to precise obligations and gain corresponding access rights recorded in the access right database Those rights are e.g.: access to the log file or network ATM. When a crisis occurs, for instance if a DoS is detected and isolates some SAVP agents from the rest of the network, the normal monitoring rules and procedures do no longer work as usual and it is requested to change the responsibility of the agents. In the above case, other agents have to assure the responsibilities of the

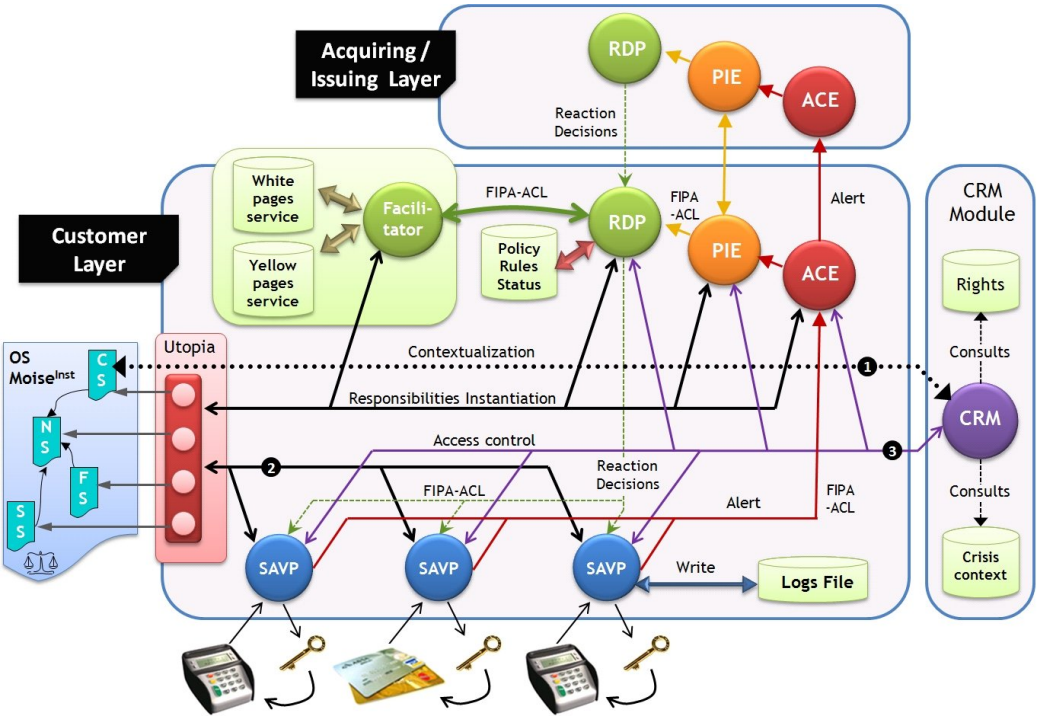

Fig. 5: Detailed reaction architecture for the clearing infrastructure

isolated SAVP. To assure their new responsibilities the agents request additional access rights e.g. monitoring probes related to the isolated ATM.

In practice, the automatic assignment of rights to agents depending on their responsibilities is not immediate and this contradicts the need to react promptly to the attack or to the problem. To tackle the weakness, our approach is based on the refinement of the agent responsibility and on the granting of access rights based on responsibility. The right management module exploits two databases accessed by the CRM agent. The first database includes the rights corresponding to the responsibilities and the second includes a set of crisis contexts. Each crisis context defines, depending on the type of crisis, which agent is assigned to which responsibility.

The review of the literature highlights that the definition of the agent responsibility is mostly incomplete and that most of the architecture only considering the agent against the outcome that he has to produce. Sometimes, advanced solutions integrate the inputs that those agents request to perform the outcome. We define responsibility as a state assigned to an agent to signify him its obligations concerning the task, its accountabilities regarding its obligations, and the rights and capabilities necessary to perform it.

In order to integrate a dynamic re-assignment of the responsibility from one agent working in normal conditions to one agent working in a crisis environment, we consider all the concepts which compose the responsibility. In [FP09] we have proposed a responsibility model that can be used to depict the agent responsibility. This model integrates the main

following concepts:

Obligation is a duty which links a responsibility with a task that must be performed.

Accountability is a duty to justify achievement, maintenance or avoidance of some given state to an authority under the threat of sanction [Sta06]. Accountability contribute to generate trust or to remove trust depending of the accountability outcomes.

Right encompasses facilities required by an agent to fulfill its obligations, e.g. the access right that the agent gets once it is assigned responsible.

Capability describes the requisite qualities, skills or resources necessary to perform a task. Capability may be declined through knowledge or know-how, possessed by the agent such as ability to make a decision, its processing time, its faculty to analyze a problem, and its position on the network.

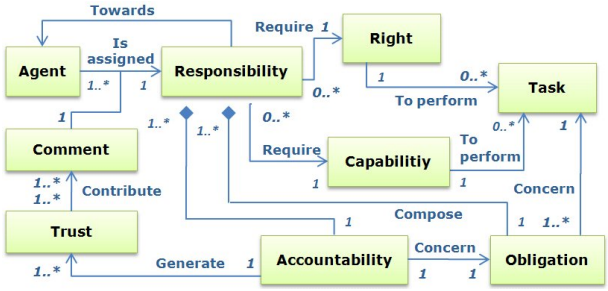

Fig. 6: Agent responsibility model

Commitment pledged by the agent represents its engagement to fulfill a task and the assurance that it does it in respect of good practices. The commitment in MAS has already been subject to many researches as explained in [Sin08]. The semantic analyzes commitment [SD09] and [MDHV02] advocates to consider the mutual **trust** between agents as a pragmatic commitment antecedent [GBKD07]. The trust component signifies the lower trust value that is necessary for an agent to be assigned responsible and is determined by the importance of the obligation that is to be achieved.

Based on the agent responsibility model of Fig. 6, we define the responsibilities for each agent of the architecture. Table 1 summarizes the SAVP and the ACE responsibilities used in practice for the dynamic assignment (section 4). Due to the extent of this paper, only the responsibility for two of them is presented: the SAVP and the ACE, and only the 4 most significant components of the responsible are exploited: Obligations, Trust, Capabilities and Rights.

For the SAVP, we observe that responsibility includes obligations such as the obligation to retrieve logs from the component it monitors (ATM, IP network, shop VISA interface, ...), to provide an immediate reaction if necessary, to report the incident to the RDP, etc. To perform this obligation, it needs the capabilities to be on the same network as the component it controls and to communicate with the ACE and the facilitator agent. It also must have the right to read log on the ATM or other interface category parameters on the concerned network component and to write the values down in a central logs database.

	Obligations	Trust	Capabilities	Rights
SAVP	• Retrieve the logs from the component it monitors; • Provide an immediate reaction if necessary; • Communicate with the facilitator to get the address of the other components (ACE, CRM); • Report the incident to the ACE.	0.8	• SAVP must be on the same network as the parameter (browser, ATM, shop, etc.) it monitors; • SAVP must be able to communicate with the ACE and the facilitator agent.	Read parameter value on the concerned network component Write parameter value in the central logs database
ACE	• Decide which keys are appropriated to be deployed by the SAVP; • Confirm the alert to the PIE; • Communicate with the facilitator to get the address of the other components (SAVP, CRM).	0.9	• Fast bandwidth; • High CPU resources; • Central position on the network.	• No specific right.

Tab. 1: Agents Responsibilities Specifications

4 Responsibility to Agent Assignement

To assign responsibilities to agents, we have associated our architecture with $\mathcal{U}$TOPIA an insTitution Oriented ProgrammIng frAmework which aims at simplify agent based institutions programming. This architecture provides organizational specification for agents institutions using $\mathcal{M}$OISE^{Inst}.

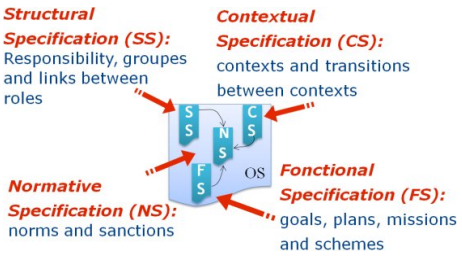

Fig. 7: $\mathcal{M}$OISE^{Inst} organization

$\mathcal{M}$OISE^{Inst} (Fig. 7.) is an Organization Specification (OS) system composed of four dimensions:

- Structural Specification (SS);
- Functional Specification (FS);
- Contextual Specification (CS);
- Normative Specification (NS).

These specifications are described in a XML file and loaded by $\mathcal{U}$TOPIA. From this defined organization, $\mathcal{U}$TOPIA instantiates a framework allowing to specify responsibilities and trust, and as consequence, to establish the agent responsible in terms of tasks, obligations, rights and trust. In the following, we explain the Structural and the Functional Specifications which are related to the trust parameter. The paper [GBKD07] addresses these specifications in details.

The Structural Specification (Fig. 8) defines (i) the responsibilities which agents are assigned to in the Organization (ii) the relations between the responsibilities in terms of data exchange during the execution of the tasks. E.g. <Link source="SAVP" destination="ACE"/> specifies that the SAVP is allowed to send information to the ACE (iii) the Groups of agents assigned to the same responsibility. E.g. <Responsibility id="SAVP" min="3" max="3"/> creates a SAVP group with a cardinality sets to 3. And (iv) the initial level of trust of each agent. E.g. According to Fig. 8 the level of trust for the ACE is set to 0.5 at boot. This value will change over the time based on acknowledgment and task performance.

```
<StructuralSpecification>
    <Role id="RDP" trust="0.5" min="1" max="1"/>
    <Role id="PIE" trust="0.5" min="1" max="1"/>
    <Role id="ACE" trust="0.5" min="1" max="1"/>
    <Role id="SAVP" trust="0.5" min="3" max="3"/>
    <Role id="SAVP" trust="0.5" min="1" max="1"/>
    <Role id="SAVP" trust="0.5" min="1" max="1"/>
    <Role id="SAVP" trust="0.5" min="1" max="1"/>

    <Link source="SAVP" destination="ACE" type="communication"
        scope="intra-group" extendsToSubGroups="true" symmetric="false"/>
    <Link source="ACE" destination="PIE" type="communication"
        scope="intra-group" extendsToSubGroups="true" symmetric="false"/>
    <Link source="PIE" destination="RDP" type="communication"
        scope="intra-group" extendsToSubGroups="true" symmetric="false"/>
    <Link source="RDP" destination="SAVP" type="communication"
        scope="intra-group" extendsToSubGroups="true" symmetric="false"/>
</StructuralSpecification>
```

Fig. 8: Structural Specification schema

In case of a crisis, if a SAVP from a SAVP group is corrupted, the responsibility of that failing agent is transferred to new agents from the same group.

The Functional Specification (FS) on Fig. 9 defines global business processes that can be executed by the different agents participating to the Organization according to their responsibilities and Groups.

```
<FunctionalSpecification>
    <Mission id="mTransmission" headGoal="gRoot">
        <GoalId>gRoot</GoalId>
        <Plan headGoal="gRoot" operator="parallel">
            <GoalId>gSend</GoalId>
            <GoalId>gListen</GoalId>
        </Plan>
    </Mission>
    <GoalId>gListen</GoalId>
    <GoalId>gSend</GoalId>
</FunctionalSpecification>
```

Fig. 9: Functional Specification schema

5 Trust and reputation management

The trust and reputation model (TRM) evaluates the responsability fulfillment of each component involved in the architecture. A large review of computational trust models can be found in [SS05] and more details for cognitive aspects of trust is presented in [CF98].

The proposed reputation-based trust management scheme is used to predict the future behavior of a component in order to establish trust among agents and hence to improve security in the system. The way trust information is quantified depends on the application field. Some authors used probabilistic approaches [VMC10] [PTJL05] [MMA+01] to quantify trust. This approach is used in several fields such as Mobile Ad-hoc Networks, Social Networks or e-commerce. Other approaches such as Fuzzy approaches or Bayesian networks [BB04][BB02][TPJL06] are also exploited to quantify trust with some differences, if a ageing factor is used or not, the way reputation parameters are aggregated. The information needed to evaluate the trustworthiness of a component is provided by the SAVP's probe.

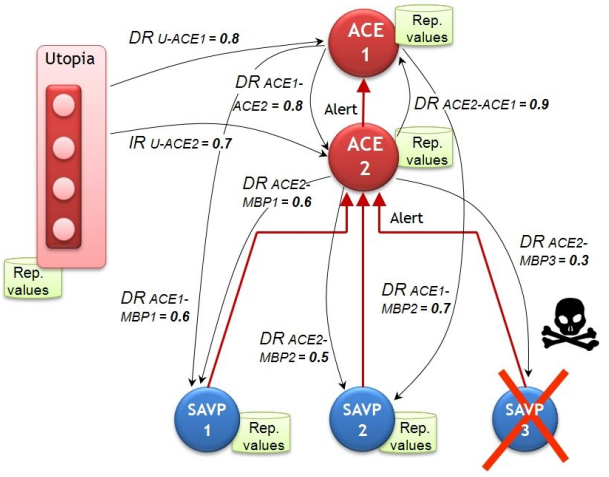

Fig. 10: Reputation Repartion Schema

As mentioned in the agent responsibility specifications, if one of the SAVP's agents is not trustworthy because for instance it has been corrupted, it has to be removed and

replaced. The assignment of a new responsibility to another SAVP (SAVP1 or SAVP2 in Fig. 10) is done based on the trust value of that agent.

5.1 Computing trust and reputation

The trust (T) and reputation (R) of each component is provided by trust assessment regarding an SAVP agent based on direct information collected by the SAVP agents himself and by second hand-information coming from other ACE to which it belongs. This trust based reputation is calculated at time t as follows:

$$T_{ACE1-SAVP2} = R_{ACE1-SAVP2} = DR_{ACE1-SAVP2} \qquad (1)$$

Where $DR_{ACE1-SAVP2}$ is the reputation derived from the direct reputation of SAVP2 in the view of ACE1. The scenario depicted in Fig. 10 shows that ACE1 has no direct connectivity with SAVP2 and has to rely on information coming from ACE2. In this case (1) leads to (2):

$$T_{ACE1-SAVP2} = R_{ACE1-SAVP2} = DR_{ACE1-SAVP2} + \lambda * IR_{ACE1-SAVP2} \qquad (2)$$

Where λ is the weighted factor associated to the reputation value and $IR_{ACE1-SAVP2}$ is the indirect reputation value provided by other agents about agent SAVP2. In the followings we presented how the direct reputation (DR) and the indirect reputation (IR) are calculated. The direct reputation is defined as the probability evaluated by ACE1 that ACE2 will achieve his goal at the next solicitation. Several approaches can be used to deal with reputation information such as Beta, Gaussian, Poisson, binomial. [BB02] prooves that the Beta distribution system is the most promising due to its flexibility and simplicity as well as its strong foundation on the theory of statistics.

Our model used the Beta probability density function $Beta(\alpha, \beta)$ as shown in equation (3) to represent direct reputation.

$$p(\theta) \ = Beta(\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} \theta^{\beta-1} \qquad (3)$$

Where θ represents the probability that the next event will occur. This Beta function is indexed with two parameters α and β, which represent the sum of positive and negative assessment, respectively that ACE1 has for ACE2. We consider that ACE1 rates ACE2 during an assessment in a binary scale (positive (1), negative (0)). The direct reputation value is the mean of the Beta distribution between ACE1 and ACE2. Reputation value for agent ACE2 about agent ACE1 is calculated as the expectation value of $Beta(\alpha, \beta)$ with E equal the standard deviation :

$$R_{ACE1-ACE2} = E(Beta(\alpha, \beta)) \qquad (4)$$

An agent ACE1 is trusted by agent ACE2 if and only if $R_{ACE1-ACE2} \leq \tau$ where τ is a threshold value from which the agent is considered trustworthy. This threshold is defined by the responsibilities specifications. For instance, the trust threshold for SAVP agents is set to 0.8. Initially, $\alpha = \beta = 0$, which corresponds to uniform distribution and indicates absence of knowledge.

$$R_{ACE1-ACE2} = E(Beta(\alpha, \beta)) = \frac{\alpha + 1}{\alpha + \beta + 2} \qquad (5)$$

The update of this reputation value is done with respect to the Beta function. This trust represents the trustworthiness of one agent to another. In order to highlight the impact of recent behavior, the ageing factor is introduced. We assume that this ageing factor g depends on the context. This value is update in (6) as follows:

$$\alpha^{new}_{ACE1-ACE2} = g * \alpha_{ACE1-ACE2} \ and \ \beta^{new}_{ACE1-ACE2} = g * \beta_{ACE1-ACE2} \qquad (6)$$

A small value of g gives more weight to recent trust values. However choosing the right value of g is not straightforward. For this purpose, we assume that the value of g is chosen by a Human at the initialization of the system. Unlike the components weights which represents the personal view that an agent has to another.

Our solution for setting components weights is based on the similarity beliefs between two agents. This similarity is based on the assumption that if two agents rate a third agent in the same way, this means that they have similar beliefs and views of this agent. Given two agents ACE1 and ACE2, let the set $(ACE1, ACE2)$ be the set of all mutual agents, that both ACE1 and ACE2 have interacted with and hence have a direct reputation with those agents in (7) as follows:

$$Set(ACE1, ACE2) = \{c, d, e, ..., n\} \qquad (7)$$

The similarity belief (Λ) between those two agents is given in (8).

$$\Lambda = Sim(ACE1, ACE2) = 1 - Dis(ACE1, ACE2) \qquad (8)$$

Where the distance function between 2 agents ACE1, ACE2 is calculated in (9) based on the Euclidian method.

$$Dis(ACE1, ACE2) = \sqrt{\sum_{i \in Set(ACE1, ACE2)} (R_{ACE1-ACEi} - R_{ACE2-ACEi})^2} \qquad (9)$$

By doing that, an agent can have no common direct history with another and in the same time have a trustworthy (or not) view of that agent.

Agents	ACE1	ACE2	λ CRM	CRM
SAVP1	*unknown*	0.6	*unknown*	0.8
SAVP2	*unknown*	0.8	*unknown*	0.9
SAVP3	*unknown*	0.1	*unknown*	0.16
ACE1		0.8	0.6	0.8
ACE2	0.9		0.7	0.7

Tab. 2: Example of Reputation Calculation

Table 2 represents the reputation value that the agents have for each other based on their common past interaction history. The grey boxes represent the reputation that agent has about itself. This is not possible since we assume that reputation is not reflexive. The unknown value means that $\mathcal{U}$TOPIA has no common history with that agent.

5.2 Example of responsibility assignments based on reputation.

In a normal situation one agent is assigned to a SAVP responsibility, each agent is associated to one or more probes (ATM, IP network,...) and reports to the ACE. Each time a SAVP agent wants to access a component of the network, it previously needs to contact the CRM that consults the access rights database in order to retrieve the rights associated to the responsibility assigned to the agent.

Based on the values of the Table 2, CRM allows to transfer the responsibilities of the corrupted agent.

During an abnormal situation, the context evolves and an appropriate crisis context should be selected. Each context is adapted to a specific case. In this use case, the crisis concerns the corruption of the SAVP3 agent and calls for a transfer of that agent responsibility from the corrupted agent to another agent from the same group. That transfer of responsibility implies, in the mean time, a transfer of access rights.

The diagram of activities, as highlighted by white numbers in black circles on Fig. 5, is the following:

1. Under a threshold of 0.5, CRM considers that an agent is no longer trustworthy. According to Table 2, the trust level of the SAVP3 is set to 0.16. Consequently the CRM adapts and sends the new crisis context to $\mathcal{U}$TOPIA. Based on that context, the CRM is able to update the access rights data base;

2. Afterwards, the SAVP3 loses its access rights and is isolated, and the responsibilities of SAVP3 are transferred to the SAVP2 which has a reputation of 0.9;

3. The agent SAVP2 that receive the additional SAVP3 responsibilities requests the access rights corresponding to that responsibility to the CRM.

6 Conclusions

Critical infrastructures are more and more present and needs to be seriously managed and monitored regarding the increasing amount of threats. This paper presents a solution for bank clearing security using a MAS based architecture. The system, initially developed based on static assignments of responsibilities to agents, has needs for more dynamism in order to keep aligned to the new arising risks. This paper firstly addresses the new challenge by providing a framework for dynamically assigning responsibilities to agents depending on the crisis context. That contextualized responsibility assignment permits, *in fine*, to manage the agent access rights to critical information. Secondly, the paper enhances that dynamic assignment taking into account the trust and reputation between the agents during the assignment mechanism.

The architecture we exploit to demonstrate the enhanced reaction mechanism, relies on the two main pillars ReD and $\mathcal{U}$TOPIA, which are being tested and currently produced in our labs. Practically ReD defines the structural bases for the clearing mechanism and

$\mathcal{U}$TOPIA permits to deploy the responsibilities to agents. The paper is illustrated based on a use case that shows how the transfer of rights for a corrupted SAVP is performed during an evolution from normal to crisis situation.

Acknowledgment

This research was funded by the National Research Fund of Luxembourg in the context of TITAN (Trust-Assurance for Critical Infrastructures in Multi-Agents Environments, FNR CO/08/IS/21) project.

References

[Act02] S.O. Act. Sarbanes-Oxley Act. In *United States of America 107th congress, US Congress (2002) Available from:(http://www. sec. gov/about/laws/soa2002. pdf)*, 2002.

[Bas06] II Basel. International convergence of capital measurement and capital standards. *A revised framework*, 2006.

[BB02] S. Buchegger and J.-Y. Le Boudec. The Beta Reputation System. *15th Bled Conference on Electronic Commerce, Bled, Slovenia, 17-19 June 2002*, 2002.

[BB04] S. Buchegger and J.-Y. Le Boudec. A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks. *P2PEcon 2004, Cambridge, MA, 4-5 June*, 2004.

[BWIM09] S. Baker, S. Waterman, G. Ivanov, and Inc McAfee. *In the Crossfire: Critical Infrastructure in the Age of Cyber War*. McAfee, Inc., 2009.

[CF98] C. Castelfranchi and R. Falcone. Principles of trust for mas: Cognitive anatomy, social importance, and quantification. In Yves Demazeau, editor, *ICMAS*, pages 72–79. IEEE Computer Society, 1998.

[FKA10] C. Feltus, D. Khadraoui, and J. Aubert. A Security Decision-Reaction Architecture for Heterogeneous Distributed Network. In *Availability, Reliability, and Security, 2010. ARES'10 International Conference on*, pages 1–8. IEEE, 2010.

[FP09] C. Feltus and M. Petit. Building a Responsibility Model Including Accountability, Capability and Commitment. In *Availability, Reliability and Security, 2009. ARES'09. International Conference on*, pages 412–419. IEEE, 2009.

[GBKD07] B. Gâteau, O. Boissier, D. Khadraoui, and E. Dubois. Controlling an interactive game with a multi-agent based normative organizational model. In J. Vázquez-Salceda, G. Boella, O. Boissier, and E. Matson, editors, *Post-Proceedings of the Workshop COIN at the 17th ECAI conference*, LNCS 4386, pages 86–100. Springer-Verlag, 2007. ISBN: 978-3-540-74457-3.

[MDHV02] J.B. Mehdi, M. Dastani, Z. Huang, and L. Van Der Torre. Trust and Commitment in Dynamic Logic. In *In Procs. of EurAsia ICT 2002, LNCS 2510*. Citeseer, 2002.

[MMA+01] Lik Mui, Mojdeh Mohtashemi, Cheewee Ang, Peter Szolovits, and Ari Halberstadt. Ratings in Distributed Systems: A Bayesian Approach, 2001.

[PTJL05] J. Patel, W. T. Luke Teacy, N. R. Jennings, and M. Luck. A probabilistic trust model for handling inaccurate reputation sources. In Peter Herrmann, Valérie Issarny, and Simon Shiu, editors, *iTrust*, volume 3477 of *Lecture Notes in Computer Science*, pages 193–209. Springer, 2005.

[Rin04] S.M. Rinaldi. Modeling and simulating critical infrastructures and their interdependencies. 2004.

[SBAG11] P. Schmitt, C. Bonhomme, J. Aubert, and B. Gâteau. Programming Electronic Institutions with Utopia. In W. Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, C. Szyperski, P. Soffer, and E. Proper, editors, *Information Systems Evolution*, volume 72 of *Lecture Notes in Business Information Processing*, pages 122–135. Springer Berlin Heidelberg, 2011.

[SD09] M.J. Smith and M. Desjardins. Learning to trust in the competence and commitment of agents. *Autonomous Agents and Multi-Agent Systems*, 18(1):36–82, 2009.

[Sin08] M.P. Singh. Semantical considerations on dialectical and practical commitments. In *Proceedings of the 23rd Conference on Artificial Intelligence (AAAI)*, pages 176–181, 2008.

[SS05] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1):33–60, September 2005.

[Sta06] B. Stahl. Accountability and Reflective Responsibility in Information Systems. *The Information Society: Emerging Landscapes*, pages 51–68, 2006.

[TPJL06] W. T. Luke Teacy, J. Patel, N. R. Jennings, and M. Luck. Travos: Trust and reputation in the context of inaccurate information sources. *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006.

[VMC10] G. Vogiatzis, I. MacGillivray, and M. Chli. A probabilistic model for trust and reputation. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 225–232. International Foundation for Autonomous Agents and Multiagent Systems, 2010.