The CRUTIAL reference critical information infrastructure architecture: a blueprint

Paulo Veríssimo,* Nuno Ferreira Neves and Miguel Correia

University of Lisboa, Faculty of Sciences,
Bloco C6, Campo Grande,
Lisboa 1749-016, Portugal
E-mail: pjv@di.fc.ul.pt
E-mail: nuno@di.fc.ul.pt
E-mail: mpc@di.fc.ul.pt
*Corresponding author

Abstract: Critical infrastructures have evolved over the past decades to become largely computerised and interconnected all over the world. This generated the problem of achieving resilience of Critical Information Infrastructures (CII) against computer-borne attacks and severe faults, similar to those observed in the internet. Governments and industry have been pushing an immense research effort in information and systems security, but we believe the complexity of the problem prevents it from being solved using classical security methods. This paper focuses on the computer systems behind electrical utility infrastructures. It proposes the blueprint of a distributed systems architecture that we believe may come to be useful as a reference for modern CII in general. The architecture is instantiated with a set of classes of techniques and algorithms, based on paradigms providing resilience to faults and attacks in an automatic way.

Keywords: critical information infrastructures; CII; distributed systems; security; fault tolerance; intrusion tolerance; firewalls; access control; middleware.

Reference to this paper should be made as follows: Veríssimo, P., Neves, N.F. and Correia, M. (2008) 'The CRUTIAL reference critical information infrastructure architecture: a blueprint', *Int. J. System of Systems Engineering*, Vol. 1, Nos. 1/2, pp.78–95.

Biographical notes: Paulo Veríssimo is a Professor of the Faculty of Sciences from the University of Lisboa and the Director of LASIGE Research Laboratory. He is Fellow of the IEEE, an Associate Editor of the Elsevier *International Journal on Critical Infrastructure Protection* and former Associate Editor of the *IEEE Transactions on Dependable and Secure Computing*. He belonged to the European Security and Dependability Advisory Board. He is the past Chair of the *IEEE Technical Committee on Fault Tolerant Computing* and of the *Steering Committee of the DSN Conference*. He is author of 130+ refereed publications in international conferences and journals, and coauthor of 5 books.

Nuno Ferreira Neves has been an Assistant Professor of the Department of Informatics, University of Lisboa since 1998. He received a PhD in Computer Science at the University of Illinois at Urbana-Champaign. His research interests

are in parallel and distributed systems, in particular in the areas of security and fault-tolerance. His work has been recognised with a Fulbright Fellowship during the doctoral studies, the William C. Carter Best Student Paper award at the 1998 IEEE International Fault-Tolerant Computing Symposium and the IBM Scientific Award in 2004. More information about him is available at http://www.di.fc.ul.pt/∼nuno.

Miguel Correia is an Assistant Professor of the Department of Informatics, University of Lisboa Faculty of Sciences. He received a PhD in Computer Science at the University of Lisboa in 2003. He is a Member of the LASIGE laboratory and the Navigators Research Group. He has been involved in several research projects related to intrusion tolerance and security, including the MAFTIA and CRUTIAL EC-IST projects, the ReSIST NoE and national projects. More information about him is available at http://www.di.fc.ul.pt/∼mpc.

1 Introduction

The largely computerised nature of critical infrastructures on the one hand, and the pervasive interconnection of systems all over the world, on the other hand, have generated one of the most fascinating current problems of computer science and control engineering: *how to achieve resilience of Critical Information Infrastructures (CII)*.

This problem is concerned with ensuring acceptable levels of service and, in last resort, the integrity of systems themselves, when faced with threats of several kinds. In this paper we are concerned with threats against computers and control computers, not the physical infrastructures themselves. These threats range from accidental events like natural faults or wrong manoeuvres (Madani and Novosel, 2005; Neumann, 1995; Rahman et al., 2006; van Eeten et al., 2006), to attacks by hackers or terrorists (Cieslewicz, 2004; Li et al., 2005; Luiijf and Klaver, 2004; Pollet, 2002; Wilson, 2006). The problem affects systems with great socio-economic value, such as utility systems like electrical, gas or water, or telecommunication systems and computer networks like the internet. In consequence, the high degree of interconnection is causing great concern, given the level of exposure of very high value systems and components to attacks that can be perpetrated in an anonymous and remote way.

Although there is an increase in the concern for using security best practices in these systems (Byres et al., 2005; US Department of Energy, 2002), we believe that the problem is not completely understood and cannot be solved with classical methods. Its complexity is mainly due to *the hybrid composition of those infrastructures*:

- The operational network, called generically Supervisory Control and Data Acquisition (SCADA),[1] composed of the computer systems that yield the operational ability to supervise, acquire data from and control the physical processes. In fact, to the global computer system, SCADA computer systems (e.g. controllers) 'are' the controlled processes (e.g. power generators), since by acting on the former, for example, through a network message, one changes the state of the latter.
- The corporate intranet, where usual departmental services (e.g. web, email, databases) and clients reside, and also the engineering and technical staff, who access the SCADA part through ad hoc interconnections.[2]

- The internet, through which intranet users get to other intranets and/or the outside world, but to which and often unwittingly, the SCADA network is sometimes connected to.

Besides the complexity due to this hybrid composition, this mixture has given an unexpected *interdisciplinary nature* to the problem: SCADA/Process Control Systems (PCS) are real-time systems, with some reliability and fault tolerance concerns, but they were classically not designed to be widely distributed or remotely accessed, let alone open to other more asynchronous and less trusted subsystems. Likewise, they were not designed with security in mind. In consequence, in scientific terms, our problem can be formulated as follows:

- the computer-related operation of a critical utility infrastructure is a distributed systems problem including interconnected SCADA/embedded networks, corporate intranets and internet/Public Switched Telephone Network (PSTN)[3] access subsystems.
- this distributed systems problem is hard, since it simultaneously includes facets of real-time, fault tolerance and security.

In this paper, we focus on the computer systems behind electrical utility infrastructures as an example, and we propose:

1 *the blueprint of a distributed systems architecture* that we believe may come to be useful as a reference for modern CII
2 *a set of classes of techniques and algorithms* based on paradigms providing resilience to faults and attacks in an automatic way.

This work is ongoing and is done in the context of the CRUTIAL European project, CRitical UTility InfrastructurAL resilience (Dondossola et al., 2006), details of which are given in the end.

As a final note, whilst it is usual to use the designation CII to denote the computer related part of the physical critical infrastructures, we do not make a differentiation of the two in this paper.

This paper is organised as follows. The following section presents the rationale of the architecture proposed. Section 3 presents the architecture. Section 4 presents the CRUTIAL Information Switches (CIS), which are fundamental components of the architecture. Section 5 presents the middleware used by some of the nodes to communicate. Finally, Section 6 concludes this paper.

2 Rationale for the model and architecture

Before presenting the details of the reference architecture, let us bring some further insight on the security problem of critical infrastructures:

- CII feature a lot of legacy subsystems and non-computer-standard components (controllers, sensors, actuators, etc.)
- conventional security and protection techniques, when directly applied to CII controlling devices, sometimes stand in the way of their effective operation.

These two facts will not change, at least for a long time, so they should be considered as additional research challenges. Despite security and dependability concerns with those

individual components being a necessity, we believe that the crucial problem is with the forest, not the trees. That is, the problem of CII insecurity is mostly created by the informatics nature of many current infrastructures, and by the generic and non-structured network interconnection of CIIs, which bring several facets of exposure, from internal unprotected wireline or wireless links, to interconnections of SCADA and corporate intranets to the internet and PSTN. This situation is conspicuous in several of the attacks reported against CIIs. For instance, the January 2003 attack of the Slammer worm against the Davis-Besse nuclear power plant (US) was due both to this combination of a computerised CII with non-structured network interconnections and lack of protection (Geer, 2006). Although the network was protected by a firewall, the worm entered through a contractor's computer connected to the CII using a telephone line.

The problems that may result from this exposure to computer-borne threats range from wrong manoeuvering to malicious actions coming from terminals located outside, somewhere in the internet. The potential targets of these actions are computer control units, embedded components and systems, that is, devices connected to operational hardware (e.g. water pumps and filters, electrical power generators and power protections, dam gates, etc.) or to telecom hardware (core routers, base stations, etc.). The failure perspectives go from unavailability of services supposed to operate 24×7, to physical damage to infrastructures. In the electrical power grid these situations have already been witnessed (Dondossola et al., 2006): among the blackouts that occurred in several countries during the summer of 2003, the analysis report (US-Canada Power System Outage Task Force, 2003) of the North American highlighted the failure of various information systems as having thwarted the utility workers' ability to contain the blackout before it cascaded out of control, leading to an escalating failure.

Whilst it seems non-controversial that such a status quo brings a certain level of threat, we know of no work that has tried to equate the problem by defining a reference model of a *CII distributed systems architecture*, providing the necessary global resilience against abnormal situations.

We believe that evaluation work based on such a model will let us learn about activity patterns of interdependencies, which will reveal the potential for far more damaging fault/failure scenarios than those that have been anticipated up to now. Moreover, such a model will be highly constructive, for it will form a structured framework for

1 conceiving the right balance between prevention and removal of vulnerabilities and attacks

2 tolerance of remaining potential intrusions and designed-in faults.

What can be done at *architectural level* to achieve resilient operation? Note that the crux of the problem lies with the fact that access to operational networks, such as remote SCADA manoeuvering, ended up entangled with access to corporate intranets and to public internet, without there being computational and resilience models that *represent* this situation, unlike what exists in simpler, more homogeneous settings, for example, classical web-based server infrastructures on internet. Our point is that *interference and threats start at the level of the macroscopic information flows between these subsystems*, and can in consequence be stopped there. This should not prevent the study of techniques at the controller level, but in this paper we will not focus on this latter issue.

Now, given the simultaneous need for real-time, security and fault tolerance, this problem is hard vis-a-vis existing paradigms. For example, many classical distributed systems

paradigms handle each of those facets separately, and just solve part of the problem. A unifying approach has gained impressive momentum currently: *intrusion tolerance* (Veríssimo et al., 2003). In short, instead of trying to prevent every single intrusion or fault, they are allowed, but tolerated: systems remain to some extent faulty and/or vulnerable, attacks on components can happen and some will be successful, but the system has the means to trigger automatic mechanisms that prevent faults or intrusions from generating a system failure.

Our approach is thus equated along the following propositions:

Proposition 1: *Classical security and/or safety techniques alone will not solve the problem: they are largely based on prevention, intrusion detection and ad hoc recovery or ultimately disconnection.*

There is a recent and positive trend to make SCADA systems and CIIs at large more secure (Byres et al., 2005; Li et al., 2005; Stamp et al., 2003; Stouffer et al., 2006; US Department of Energy, 2002). However, classic engineering remedies place Real-Time and Embedded (RTE) systems at most at the current level of commercial systems' security and dependability, which is known to be insufficient (Cieslewicz, 2004; Gordon et al., 2006; Turner et al., 2005): systems constantly suffer attacks, intrusions, some of them massive (worms); most defences are dedicated to generic non-targeted attacks; attacks degrade business but only do virtual damage, unlike RTE systems where there is a risk of great social impact and even physical damage. Even *firewalls* in which much hope for securing CIIs is placed (Byres et al., 2005), are constantly plagued by vulnerabilities (Kamara et al., 2003). On the other hand, some current IT security techniques can negatively affect RTE system operation, with respect to availability and timeliness. For example, if security is based on disconnection, significant performance degradation or even defensive restrictions can prevent the actuation or monitoring of the infrastructure.

Proposition 2: *Any solution, to be effective, has to involve automatic control of macroscopic command and information flows, occurring essentially between the physical or virtual Local Area Networks (LANs) composing the CII architecture, with the purpose of securing appropriate system-level properties.*

We believe that a key to the solution lies with controlling the command and information flow at macroscopic level – at organisational level. We are talking about an architectural model, a set of architectural devices and key algorithms, capable of achieving the above-mentioned control of the command and information flow. The devices and algorithms should be capable of securing a set of system-level properties characterising whatever is meant by correct and resilient behaviour.

Proposition 3: *We lack a reference architecture of 'modern critical information infrastructure' considering different interconnection realms and different kinds of risk, throughout the physical and the information subsystems of a CII.*

We must consider the physical or virtual LANs composing the operational SCADA/embedded networks, the corporate intranets and the internet/PSTN access networks, as different first order citizens of the architecture. Likewise, the notion that risk factors may vary and be difficult to perceive accurately, brings the need to reconcile uncertainty with predictability in architecture and algorithmics.

3 CRUTIAL architecture

The CRUTIAL architecture encompasses four aspects:

- architectural configurations featuring trusted components in key places, which a priori induce prevention of some faults and of certain attack and vulnerability combinations
- middleware devices that achieve runtime automatic tolerance of remaining faults and intrusions, supplying trusted services out of non-trustworthy components
- trustworthiness monitoring mechanisms detecting situations not predicted and/or beyond assumptions made, and adaptation mechanisms to survive those situations
- organisation-level security policies and access control models capable of securing information flows with different criticality within/in/out of a CII.

We build on results from the Malicious- and Accidental-Fault Tolerance for Internet Applications (MAFTIA) project[4] in this field (Veríssimo et al., 2006), but extend them significantly to attend the specific challenges of the CII problem, for example, timeliness, global access control and above all non-stop operation and resilience.

Given the severity of threats expected, some key components are built using architectural hybridisation methods in order to achieve *trusted-trustworthy* operation (Veríssimo et al., 2006): an architectural paradigm whereby components prevent the occurrence of some failure modes *by construction*, so that their resistance to faults and hackers can justifiably be trusted. In other words, some special-purpose components are constructed in such a way that we can argue that they are always secure, so that they can provide a small set of services useful to support intrusion tolerance in the rest of the system.

Intrusion tolerance mechanisms are selectively used in the CRUTIAL architecture, to build layers of progressively more trusted components and middleware subsystems, from baseline untrusted components (nodes, networks) (Veríssimo et al., 2006). This leads to an automation of the process of building trust: for example, at lower layers, basic intrusion tolerance mechanisms are used to construct a trustworthy communication subsystem, which can then be trusted by upper layers to securely communicate amongst participants without bothering about network intrusion threats.

One of the innovative aspects of this work, further to intrusion tolerance, is the resilience aspect, approached through two paradigms: *proactive-resilience* to achieve exhaustion-safety (Sousa et al., 2005a), to ensure perpetual, non-stop operation despite the continuous production of faults and intrusions; and *trustworthiness monitoring* to perform surveillance of the coverage stability of the system, that is, of whether it is still performing inside the assumed fault envelope or beyond assumptions made (Bondavalli et al., 2004). In the latter case, dependable adaptation mechanisms are triggered.

Finally, the desired control of the information flows is partly performed through protection mechanisms using an adaptation of the *Organisation-Based Access Control Model (OrBAC)* (El Kalam et al., 2003) for implementing global-level security policies. OrBAC allows the expression of security policy rules as high level abstractions, and the composition of the security policies of the organisations into one global policy.

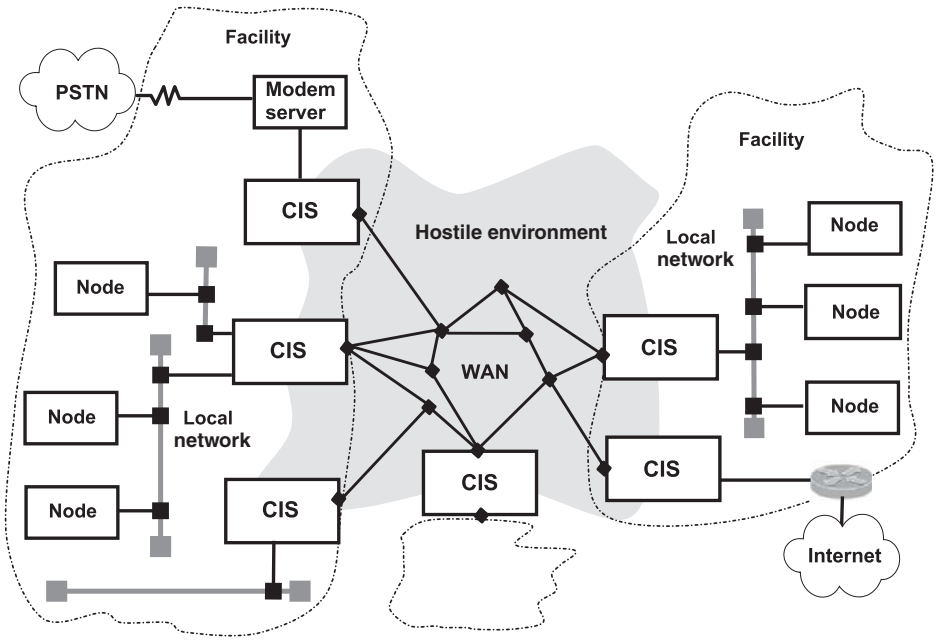
The mechanisms and algorithms in place achieve system-level properties of the following classes: trustworthiness or resistance to faults and intrusions (i.e. security and dependability); timeliness, in the sense of meeting timing constraints raised by real world control and supervision; coverage stability, to ensure that variation or degradation of

assumptions remains within a bounded envelope; dependable adaptability, to achieve predictability in uncertain conditions; resilience, read as correctness and continuity of service even beyond assumptions made.

3.1 Main architectural options

We view the system as a WAN-of-LANs, as introduced in Veríssimo (2002). There is a global interconnection network, the WAN, that switches packets through generic devices that we call *facility gateways*, which are the representative gateways of each LAN (the overall picture is shown in Figure 1). The WAN is a logical entity operated by the CII operator companies, which may or may not use parts of public network as physical support. A LAN is a logical unit that may or may not have physical reality (e.g. LAN segments versus Virtual LANs (VLANs)). More than one LAN can be connected by the same facility gateway. All traffic originates from and goes to a LAN. As example LANs, the reader can envision: the administrative clients and the servers LANs; the operational (SCADA) clients and servers LANs; the engineering clients and servers LANs; the PSTN modem access LANs; the internet and extranet access LANs, etc.

Figure 1 CRUTIAL overall architecture (WAN-of-LANs connected by CIS, *P* processes live in the several nodes)



The facility gateways of a CRUTIAL CII are more than mere TCP/IP routers. Collectively they act as a set of servers providing distributed services relevant to solving our problem: *achieving control of the command and information flow, and securing a set of necessary system-level properties*. CRUTIAL facility gateways are called CRUTIAL Information Switches (CIS), which in a simplistic way could be seen as sophisticated circuit or application level firewalls combined with equally sophisticated intrusion detectors, connected by distributed protocols.

This set of servers must be intrusion-tolerant, prevent resource exhaustion providing perpetual operation (i.e. cannot stop) and be resilient against assumption coverage uncertainty, providing survivability. The services implemented on the servers must also secure the desired properties of flow control, in the presence of malicious traffic and commands and in consequence be themselves intrusion-tolerant.

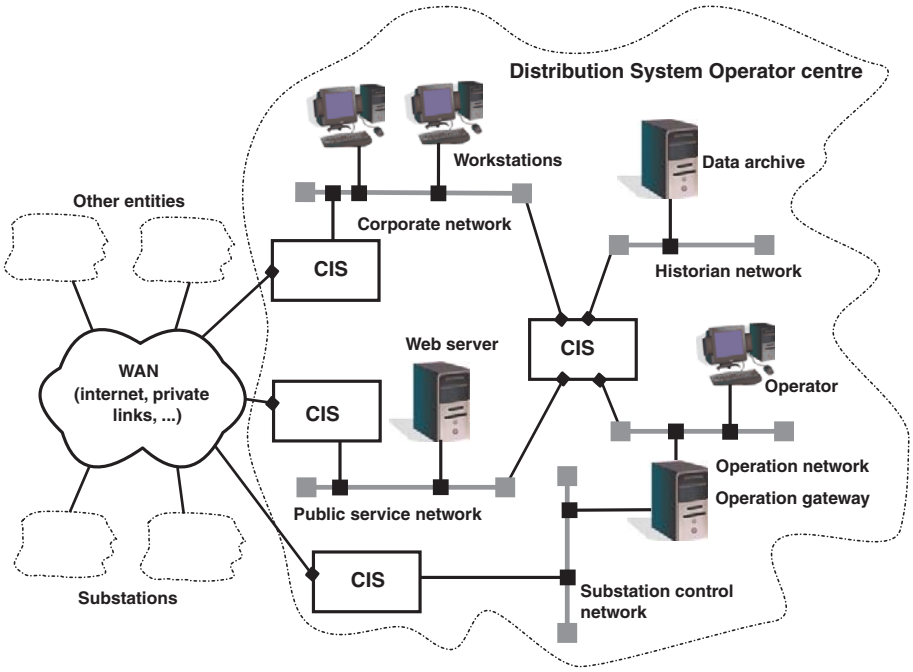
An assumed number of components of a CIS can be corrupted. Therefore, a CIS is a logical entity that has to be implemented as a set of replicated physical units (CIS replicas) according to fault and intrusion tolerance needs. Likewise, CIS are interconnected with intrusion-tolerant protocols, in order to cooperate to implement the desired services. The CIS boxes in the figure represent these intrusion-tolerant, replicated, logical CIS.

3.1.1 An example WAN-of-LANs

The WAN-of-LANs model is very abstract so in this section we use it to represent a small part of a distribution power grid. This example is inspired in a testbed of the CRUTIAL project presented in Deconinck et al. (2007),[5] and the corresponding scenario in Garrone et al. (2007).

Figure 2 presents a Distribution System Operator (DSO) centre. This centre includes several networks and is connected to the substations through the substation control network (bottom). This network is connected to the substations through the (logical) WAN, which can be the internet, a set of private links, VLANs or other type of network. The DSO centre includes the corporate network (top), the public service network were services like web servers are placed (middle), the data historian network were historical information about the infrastructure is stored (top right) and the operation network were operators monitoring and controlling the power generation infrastructure (right). All these networks are modelled as (logical) LANs and are connected by CIS, that protect them from one another and, especially, from the internet/WAN.

Figure 2 Example mapping of part of an infrastructure to the WAN-of-LANs architecture



3.2 CRUTIAL nodes

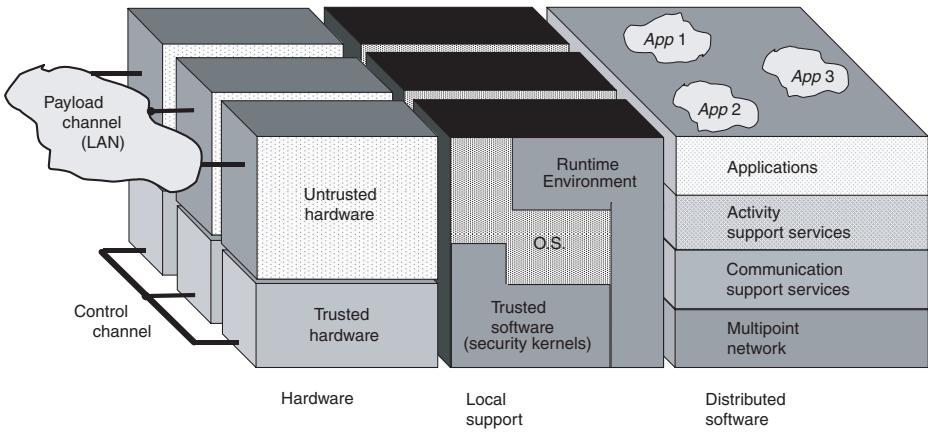
The structure of some of the CII nodes, which we call *CRUTIAL nodes*, can follow the node structuring principles for intrusion-tolerant systems explained in Veríssimo et al. (2006):

- The notion of *trusted* – versus untrusted – *hardware*. For example, most of the hardware of a CIS is considered to be untrusted, with small parts of it being considered trusted-trustworthy.
- The notion of *trusted support software*, trusted to execute a few critical functions correctly, the rest being subjected to malicious faults.
- The notion of *run-time environment*, offering trusted and untrusted software and operating system services in a homogeneous way.
- The notion of *trusted distributed components*, for example software functions implemented by collections of interacting CIS middleware.

In the context of this paper, we consider only one instantiation of CRUTIAL nodes, the CIS nodes. However, other specific nodes, for example, controllers needing to meet high trustworthiness standards, may be, also built to a similar structure.

A snapshot of the CRUTIAL node is depicted in three dimensions in Figure 3, where we can perceive the above-mentioned node structuring principles.

Figure 3 Architecture and interconnection of CRUTIAL nodes (e.g. CIS)



Firstly, there is the *hardware* dimension, which includes the node and networking devices that make up the physical distributed system. We assume that most of a node's operations run on untrusted hardware, for example, the usual machinery of a computer, connected through the normal networking infrastructure, which we call the *payload channel*. However, some nodes – CIS, for example – may have pieces of hardware that are trusted, for example, that by construction intruders do not have direct access to the inside of those components. The type of trusted hardware featured in CIS is an *appliance board with processor*, which may

or not have an *adapter to a control channel* (an alternative trusted network), as depicted in Figure 3. This appliance is plugged to the CIS's main hardware.

Secondly, services based on the trusted hardware are accessed through the *local support* services. The rationale behind our trusted components is the following: whilst we let a local node be compromised, we make sure that the trusted component operation is not undermined (crash failure assumption).

Finally, there is the *distributed software* provided by CRUTIAL: middleware layers on top of which distributed applications run, even in the presence of malicious faults (far right in Figure 3). In the context of this paper, we will discuss the layers of *middleware* running inside a CIS (Section 5).

4 CRUTIAL information switches

The CIS are fundamental components of the CRUTIAL architecture, since they are the components in charge of controlling the information flow in the CII, securing a set of system-level properties (see Section 3.1). In some sense these components can be considered to be sophisticated firewalls since they have the following characteristics:

- *Distributed firewall*: similarly to a distributed firewall (Bellovin, 1999), the CIS can be deployed redundantly, protecting not only the perimeter of a network, but also subnetworks or even individual computers. This protects the subsystems from insider attacks. This idea can be watched in Figure 2.
- *Rich access control model*: the CIS evaluates access control rules that are more complex than normal, since it supports the OrBAC (El Kalam et al., 2003). This is important, for instance, in current electrical power CIIs, with multiple interconnected organisations involved, for example, in generation, transmission and distribution of energy and even regulation agencies, several of which may be allowed to do some operations on the system depending of its state.
- *Application-level firewall*: the CIS filters application-level communication, for example, validating if certain operations on the power system are allowed or not. This is important because some legacy SCADA/PCS systems do not do access control.
- *Intrusion-tolerant*: CIS are intrusion-tolerant, that is, they behave correctly even if some of their components are attacked and corrupted.

Let us present how CIS are made trusted-trustworthy components, that is, how they are ensured to behave according to their specification even if there are intrusions in some of their components.

CIS are built with a combination of untrusted and trusted hardware of varying degrees, depending on the needs and criticality of the traffic and the services they support (recall Figure 3). Consider that a CIS provides a service that can be implemented by a (hard- and software) component C. The CIS is made intrusion-tolerant using two basic techniques:

- *Replication*: a CIS is implemented by a set of n component replicas C_i in such a way that if there are intrusions in at most $f < n$ of those components, a vote of the outputs of all the components allows the CIS to behave according to the specification of the service. Replication is the most commonly proposed technique for intrusion tolerance (see, e.g. Castro and Lisko, 2002; Malkhi and Reiter, 1998; Veríssimo et al., 2006).

- *Proactive recovery*: periodically each component replica C_i is rejuvenated in such a way that if there is an intrusion in C_i, then the intrusion is no longer present after the rejuvenation process (the modifications that the attacker made to the replica state are entirely removed). Recently it has been shown that proactive recovery has to be supported by a construct called Proactive Resilience Wormhole, requiring trusted hardware or trusted software (Sousa et al., 2005b).

A CIS implemented using replication and proactive recovery can aim for perpetual execution, despite continued intrusion and/or failure of an assumed simultaneous maximum number of CIS replicas (f) at an assumed maximum rate. A complete design is presented elsewhere (Bessani et al., 2007).

These notions can be recursively used to construct a logical CIS which is in fact a set of replicated physical CIS units, running some internal intrusion-tolerant protocols so that the whole appears to the protocol users as a single logical entity sinking/sourcing to/from a given LAN, but is in fact resilient to attacks on the CIS themselves. This is a powerful combination since the resilience of protocols running on such intrusion-tolerant CIS components is commensurate to arbitrary-failure counterparts.

CIS are in addition provided with trustworthiness monitoring subsystems, aiming at assessing the trustworthiness of the CIS itself: as a function of the evolution of the coverage of the assumptions underlying the whole fault and intrusion tolerant design. As such, trustworthiness becomes a dynamic property, which provides further resilience to the CIS, through dependable adaptation: automatically reacting to environment uncertainty (changing fault and/or attack levels) and maintaining coverage stability, by changing operation parameters or modes automatically. Finally, for very high levels of resilience, CIS construction and or reconfiguration in the course of proactive recovery may be based on diversity techniques (ex. n-version programming, obfuscation, etc.) (Littlewood and Strigini, 2004; Obelheiro et al., 2006).

The desired properties of the (logical) CIS have to be assured using proper methodologies. At a first stage, we plan to test CIS using attack injection techniques (Neves et al., 2006), in which attacks are generated and performed automatically with the purpose of finding vulnerabilities. However, ultimately CIS will have to pass a certification process, for example, based on the Common Criteria (ISO/IEC Standard 15408, 1999).

5 CRUTIAL middleware

We now observe the part of the system made of the WAN and all the CIS that interconnect all the internal LANs of the critical information infrastructure to the WAN (recall Figure 1).

We model this setting as a distributed system with N nodes (CIS). We use the weakest fault and synchrony models that allow to carry out the application tasks. So, we use the asynchronous/arbitrary model, which does not make any assumptions about either time needed to make operations and faults/intrusions that can occur, as a starting point and strengthen it as needed. For example, by resorting to hybrid models using wormholes (Veríssimo, 2006), and assuming some form of partial synchrony (Dwork et al., 2988).

We assume that the environment formed by the WAN and all the CIS is hostile (not trusted), and can thus be subjected to malicious (or arbitrary or Byzantine)[6] faults. On the other hand, LANs trust the services provided by the CIS, but are not necessarily trusted by the latter. That is, as we will see below, LANs have different degrees of trustworthiness, which

the CIS distributed protocols have to take into account. CIS securely switch information flows as a service to edge LANs as clients.

We assume that faults (accidental, attacks, intrusions) continuously occur during the life-time of the system, and that a maximum number of f malicious (or arbitrary) faults can occur within a given interval. We assume that services running in the nodes (CIS) cooperate through distributed protocols in such an environment. In consequence, these nodes have to be replicated for fault/intrusion tolerance.

Some of the services running in CIS may require some degree of timeliness, given that SCADA implies synchrony and this is a hard problem with malicious faults, so we plan to do research in this issue. We also take into account that these systems should operate non-stop, a hard problem with resource exhaustion (the continued production of faults during the life-time of a perpetual execution system leads to the inevitable exhaustion of the quorum of nodes needed for correct operation (Sousa et al., 2005a)).

5.1 LAN-level services

A LAN is the top-level unit of the granularity of access control, regardless of possible finer controls. It is also and correspondingly, a unit of trust or mistrust thereof. In fact, we are not concerned with what happens inside a LAN, except that we may attribute it a different level of trust. For instance, if the LAN is a SCADA network, the level of trust is high, but if it is the access to the internet then the level of trust is low.

Traffic (packets) originating from a LAN receive a label that reflects this level of trust, and contains access control information, amongst other useful data. The trustworthiness of a label (that is, the degree in which it can or not be tampered with) can vary, depending on the criticality of the service. In the context of this paper, and without loss of generality, we assume it is an authenticated proof of a capacity.

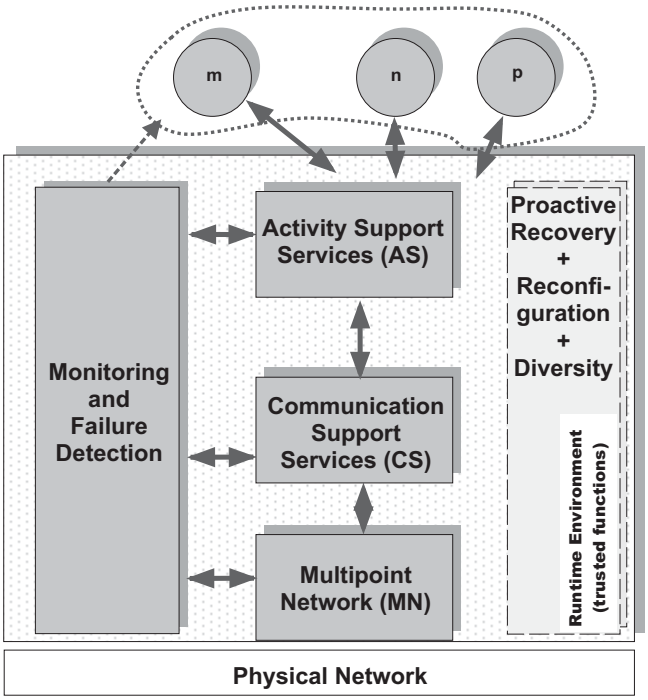
5.2 WAN-level services

The collection of CIS implements a set of core services, aiming at achieving the objectives we placed as desirable for a reference model of *CII distributed systems architecture*:

- intrusion-tolerant information and command dissemination between CIS units, with authentication and cryptographic protection (broadcast, multicast, unicast)
- pattern-sensitive information and command traffic analysis (behaviour and/ or knowledge-based intrusion detection) with intrusion-tolerant synchronisation and coordination between local Intrusion Detection Systems (IDSs)
- CIS egress/ingress access control based on LAN packet labels and/or additional mechanisms, implementing an instance of the global security policy.

The CIS middleware layers implement functionality at different levels of abstraction, as represented in Figure 4. As mentioned earlier, a middleware layer may overcome (through intrusion tolerance) the fault severity of lower layers and provide certain functions in a trustworthy way.

Figure 4 CRUTIAL middleware



5.2.1 Multipoint network

The lowest module in the figure –MN– provides basic communication services. They are 'basic' in the sense that they support upper layer protocols and are provided by standard protocols, like IP, IPsec and TCP. These services stand at the higher layers of the TCP/IP reference model: Network, Transport and Application.

The main service provided by the *Network layer* in the internet is routing data packets, called datagrams, from the source host to the destination host. Hosts are interconnected by nodes called routers that inspect the datagrams to forward – or route – them to the next router or the destination. The main protocol at this level is the Internet Protocol (IP), which has been extended to support group multicast – IP Multicast. IP does not ensure the communication security. This means that messages can be modified and their content read by anyone with access to the network, for example, a hacker controlling a router. To deal with this problem, there is a security extension to IP called IPsec (Kent and Atkinson, 1998). IPSec has an important role in CRUTIAL since it is a basic mechanism to ensure security in the network layer. IPsec is divided into two basic (sub)protocols, which may be applied alone or in combination with each other to provide the desired set of security properties in IP:

- *Authentication Header (AH)* provides connectionless integrity, data origin authentication and an optional anti-replay service.
- *Encapsulation Security Payload (ESP)* provides payload confidentiality (using encryption) and limited traffic flow confidentiality. Optionally, it may also provide connectionless integrity, data origin authentication and an anti-replay service.

IP solves the problem of end-to-end communication between two hosts. However, the problem we really want solved is slightly different: end-to-end communication between applications. This is the problem solved by the *Transport layer*. The standard transport layer internet protocols are the User Datagram Protocol (UDP) and the Transmission Control Protocol (TCP). Both protocols are used to support communication in critical infrastructures, so both are relevant for the CRUTIAL middleware. UDP provides a (unreliable) datagram mode of packet-switched computer communication in an interconnected set of computer networks. TCP, on the other hand, is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols supporting multinetwork applications. TCP over IPsec provides reliable and secure communication channels. Another Transport layer protocol that provides a similar service is the Secure Socket Layer (SSL), later standardised as Transport Layer Security (TLS). This protocol adds security to TCP, instead of relying on IPsec. The security guarantees provided by SSL/TLS are similar to those provided by TCP over IPsec, except for the more powerful authentication scheme and the usual availability of a user-level API, something that is not common with IPsec.

5.2.2 Communication Support services

The CS module provides security and Byzantine fault tolerance primitives, like Byzantine agreement, reliable multicast and view-synchronous atomic multicast, which enable, for instance, the construction of intrusion-tolerant services like a replicated CIS. The CS module depends on the MN module for basic communication. For instance, Byzantine agreement can be implemented over secure channels provided by TCP over IPsec, or by SSL/TLS. All these protocols aim to be used in an environment prone to malicious attacks and intrusions, so they are Byzantine fault-tolerant or intrusion-tolerant. In other words, they behave as expected even if some of the processes that execute them behave maliciously trying to break the protocol properties.

The CRUTIAL middleware, and specifically the CS module, provides primitives for multiparty communication and computation. Therefore, the primitives support group communication. Groups of processes or hosts can be open or closed. An open group model permits arbitrary hosts to send messages to the group, while in a closed model only hosts which are already members of the group may communicate. Groups can also be static or dynamic. In a static group the membership does not change over time, or changes at a very long time scale, such as only upon manual reconfiguration. On the contrary, dynamic groups allow nodes to join a group, leave it or be excluded if they are faulty (e.g. if they crashed or behaved maliciously).

The main types of primitives provided by the CS module are:

- *Byzantine consensus (or Byzantine agreement)*: reaches agreement on one of the values proposed by each of a set of nodes. This is a classical distributed systems problem, with a great practical interest, since several other distributed systems problems are reducible or equivalent to it (Correia et al., 2006).
- *Reliable multicast*: a multicast primitive defined in terms of two properties:
  - all correct nodes deliver the same messages
  - if the sender is correct, then the message is delivered.

- *Atomic multicast*: similar to reliable multicast but the messages are delivered in all nodes in the same order. This primitive, however, is more expensive than reliable multicast, since it involves solving consensus about the order in which messages have to be delivered.

5.2.3 Activity Support services

The AS module implements building blocks that assist participant activity, such as replication management (e.g. state machine replication, voting), IDS and firewall support functions, access control. It depends on the services provided by the CS module. The main service currently envisaged at this level is access control based on the Poly-OrBAC model, an extension of the OrBAC model, which allows to define and verify policies for the collaboration among CII organisations.

5.2.4 Other modules

The block on the left of the figure generically implements *Monitoring and Failure Detection*. Failure detection assesses the connectivity and correctness of remote nodes, and the liveness of local processes. Trustworthiness monitoring and dependable adaptation mechanisms also reside in this module, and have interactions with all the modules on the right. Both the AS and CS modules depend on this information, for example, to maintain updated information about group membership.

The block to the right represents the support services. These include the usual operating system's services, but also the trusted services supplied in support to the algorithms in the various modules: proactive recovery, reconfiguration and diversity management.

6 Conclusion

This paper presents a blueprint of a distributed systems architecture for resilient CII, with respect to both accidental faults and malicious attacks and intrusions. The rationale for this work was based on three fundamental propositions: classical security and/or safety techniques alone will not be enough to solve the problem; any effective solution has to involve automatic control of macroscopic command and information flows between the LANs composing the CII; and, the unifying paradigm should be a reference architecture of 'resilient CII' performing the integration of the different realms of a CII system.

The proposed solution encompasses a range of mechanisms of incremental effectiveness, to address from the lowest to the highest criticality operations in a CII. Architectural configurations with trusted components in key places induce prevention of some attacks. Middleware software attains automatic tolerance of the remaining faults and intrusions. Trustworthiness enforcing and monitoring mechanisms allow unforeseen adaptation to extremely critical, not predicted situations, beyond the initial assumptions made.

Functionally, the information flow is controlled by basic mechanisms of the firewall and intrusion detection type, complemented and parameterised by organisation-level security policies and access control models, capable of securing information flows with different criticality within a CII and in/out of it.

Acknowledgements

CRUTIAL is a project of the IST programme of the European Commission. Several institutions participating in the project: CESI RICERCA (Italy), FCUL (Portugal), CNR-ISTI (Italy), LAAS-CNRS (France), K.U.Leuven-ELECTA (Belgium), CNIT (Italy). Details about the project can be found at the CRUTIAL portal: `http://crutial.cesiricerca.it/`. We warmly thank our partners in the project for many discussions on the topics of this paper. We also thank our colleagues and students at the Navigators group for their collaboration and feedback on this work.

This work was mainly supported by the EC, through project IST-FP6-STREP 027513 (CRUTIAL) and NoE IST-4-026764-NOE (ReSIST), and by the FCT through the Multiannual and the CMU-Portugal Programmes.

References

Bellovin, S.M. (1999) 'Distributed firewalls', *Login*, November.

Bessani, A.N., Sousa, P., Correia, M., Neves, N.F. and Veríssimo, P. (2007) 'Intrusion-tolerant protection for critical infrastructures', DI/FCUL TR 07–8, Department of Informatics, University of Lisbon, April.

Bondavalli, A., Chiaradonna, S., Cotroneo, D. and Romano, L. (2004) 'Effective fault treatment for improving the dependability of COTS and legacy-based applications', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 1, No. 4, pp.223–237.

Byres, E., Karsch, J. and Carter, J. (2005) 'NISCC good practice guide on firewall deployment for SCADA and process control networks', Technical Report, NISCC, February 2005, Revision 1.4.

Castro, M. and Liskov, B. (2002) 'Practical Byzantine fault tolerance and proactive recovery', *ACM Transaction on Computer Systems*, Vol. 20, No. 4, November 2002, pp.398–461.

Cieslewicz, J. (2004) 'Attacks and accidents: policy to protect the power grid's critical computing and communication needs', Senior interdisciplinary honors thesis in international security studies, Stanford University, May.

Correia, M., Neves, N.F. and Veríssimo, P. (2006) 'From consensus to atomic broadcast: time-free Byzantine-resistant protocols without signatures', *Computer Journal*, Vol. 41, No. 1, pp.82–96, January.

Deconinck, G., Dondossola, G., Garrone, F. and Rigole, T. (2007) 'Testbeds deployment of representative control algorithms interim report', Project CRUTIAL EC IST-FP6-STREP 027513 Deliverable D24, January.

Dondossola, G., Deconinck, G., Di Giandomenico, F., Donatelli, S., Kaaniche, M. and Veríssimo, P. (2006) 'Critical utility infrastructural resilience', *International Workshop on Complex Network and Infrastructure Protection*, March.

Dwork, C., Lynch, N. and Stockmeyer, L. (1988) 'Consensus in the presence of partial synchrony', *Journal of the ACM*, Vol. 35, No. 2, pp.288–323, April.

El Kalam, A.A., Elbaida, R., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., Miège, A., Saurel, C. and Trouessin, G. (2003) 'Organization-based access control', *IEEE 4th International Workshop on Policies for Distributed Systems and Networks*, pp.277–288, June.

Garrone, F., Brasca, C., Cerotti, D., Raiteri, D.C., Daidone, A., Deconinck, G., Donatelli, S., Dondossola, G., Grandoni, F., Kaaniche, M. and Rigole, T. (2007) 'Analysis of new control applications', Project CRUTIAL EC IST-FP6-STREP 027513 Deliverable D2, January.

Geer, D. (2006) 'Security of critical control systems sparks concern', *IEEE Computer*, pp.20–23, January.

Gordon, L.A., Loeb, M.P., Lucyshyn, W. and Richardson, R. (2006) 'CSI/FBI computer crime and security survey', *Computer Security Institute*.

ISO/IEC Standard 15408 (1999) Evaluation Criteria for IT Security, parts 1 to 3.

Kamara, S., Fahmy, S., Schultz, E., Kerschbaum, F. and Frantzen, M. (2003) 'Analysis of vulnerabilities in internet firewalls', *Computers and Security*, Vol. 22, No. 3, pp.214–232, April.

Kent, S. and Atkinson, R. (1998) 'Security architecture for the internet protocol', *IETF Request for Comments: RFC 2093*, November.

Lamport, L., Shostak, R. and Pease, M. (1982) 'The Byzantine generals problem', *ACM Transactions on Programming Languages and Systems*, Vol. 4, No. 3, pp.382–401, July.

Li, H., Rosenwald, G.W., Jung, J. and Liu, C. (2005) 'Strategic power infrastructure defense', *Proceedings of the IEEE*, Vol. 93, No. 5, pp.918–933, May.

Littlewood, B. and Strigini, L. (2004) 'Redundancy and diversity in security', in P. Samarati, P. Rian, D. Gollmann and R. Molva (Eds). *Computer Security – ESORICS 2004, 9th European Symposium on Research Computer Security*, LNCS 3193, Springer, pp.423–438.

Luiijf, H. and Klaver, M. (2004) 'The current state of threats', *e-Security in Europe: Todays Status and The Next Step*, October.

Madani, V. and Novosel, D. (2005) 'Getting a grip on the grid', *IEEE Spectrum*, Vol. 42, No. 12, pp.42–47, December.

Malkhi, D. and Reiter, M. (1998) 'Byzantine quorum systems', Vol. 11, pp.203–213.

Neumann, P.G. (1995) *Computer Related Risks*, Addison Wesley.

Neves, N.F., Antunes, J., Correia, M., Veríssimo, P. and Neves, R. (2006) 'Using attack injection to discover new vulnerabilities', *Proceedings of the IEEE International Conference on Dependable Systems and Networks*, June.

Obelheiro, R.R., Bessani, A.N., Lung, L.C. and Correia, M. (2006) 'How practical are intrusion-tolerant distributed systems?' DI-FCUL TR 06–15, Department of Informatics, University of Lisbon, September.

Pollet, J. (2002) 'Developing a solid SCADA security strategy', *Proceedings of the ISA/IEEE Sensors for Industry Conference*, pp.148–156, November.

President's Critical Infrastructure Protection Board and Office of Energy Assurance US Department of Energy, (2002) *21 Steps to Improve Cyber Security of SCADA Networks*, US Department of Energy.

Rahman, H.A., Beznosov, K. and Marti, J.R. (2006) 'Identification of sources of failures and their propagation in critical infrastructures from 12 years of public failure reports', *Proceedings of the 3rd International Conference on Critical Infrastructures*.

Sousa, P., Neves, N.F. and Veríssimo, P. (2005a) 'How resilient are distributed f fault/intrusion-tolerant systems?' *Proceedings of the IEEE International Conference on Dependable Systems and Networks*, June.

Sousa, P., Neves, N.F. and Veríssimo, P. (2005b) 'Resilient state machine replication', *Proceedings of the 11th Pacific Rim International Symposium on Dependable Computing*, pp.305–309, December.

Stamp, J., Dillinger, J., Young, W. and DePoy, J. (2003) 'Common vulnerabilities in critical infrastructure control systems', *Technical report, Sandia National Laboratories*, May.

Stouffer, K., Falco, J. and Kent, K. (2006) 'Guide to supervisory control and data acquisition (SCADA) and industrial control systems security', *Recommendations of the National Institute of Standards and Technology*, Special Publications 800-82, NIST, September.

Turner, D., Entwisle, S., Friedrichs, O., Ahmad, D., Blackbird, J., Fossi, M., Hanson, D., Gordon, S., Cole, D., Cowlings, D., Morss, D., Bradley, B., Szor, P., Chien, E., Ward, J., Gough, J. Talbot, J. (2005) 'Symantec internet security threat report, Trends for January 05–June 05, Vol. VIII, September.

US-Canada Power System Outage Task Force (2003) *Interim Report: Causes of the August 14th Blackout in the United States and Canada*, November.

van Eeten, M., Roe, E., Schulman, P. and de Bruijne, M. (2006) 'The enemy within: system complexity and organizational surprises', in M. Dunn and V. Mauer (Eds). *International CIIP Handbook 2006*, Center for Security Studies, ETH Zurich, Vol. II, pp.89–110.

Veríssimo, P. (2002) 'Lessons learned with NavTech: a framework for reliable large-scale applications', DI/FCUL TR 02–17, Department of Informatics, University of Lisbon, December.

Veríssimo, P. (2006) 'Travelling through wormholes: a new look at distributed systems models', *SIGACTN: SIGACT News (ACM Special Interest Group on Automata and Computability Theory)*, Vol. 37, No. 1, pp.66–81.

Veríssimo, P., Neves, N.F., Cachin, C., Poritz, J., Powell, D., Deswarte, Y., Stroud, R. and Welch, I. (2006) 'Intrusion-tolerant middleware: the road to automatic security', *IEEE Security and Privacy*, Vol. 4, No. 4, pp.54–62, July/August.

Veríssimo, P., Neves, N.F. and Correia, M. (2003) 'Intrusion-tolerant architectures: concepts and design', in R. Lemos, C. Gacek and A. Romanovsky (Eds). *Architecting Dependable Systems*, Vol. 2677, pp.3–36.

Wilson, C. (2006) 'Terrorist capabilities for cyber-attack', in M. Dunn and V. Mauer (Eds). *International CIIP Handbook 2006*, Center for Security Studies, ETH Zurich, Vol. II, pp.69–88.

Notes

[1]Or Process Control System (PCS).

[2]In some companies there is a healthy reluctance against interconnecting SCADA networks and the corporate network or the internet. However, in practice this interconnection is a reality in many companies all over the world. We believe this is indeed the situation in most companies and this is the case we are interested in this paper.

[3]Public Switched Telephone Network.

[4]The web site of the project is at www.maftia.org.

[5]See Section 3.1.1 of that document.

[6]Arbitrary faults, which include attacks and intrusions, are usually called 'Byzantine faults' after the seminal paper that explained the problem in terms of 'Byzantine generals' (Lamport et al., 1982). Byzantine fault tolerance and intrusion tolerance often mean the same in recent literature for example, (Castro and Liskov, 2002; Malkhi and Reiter, 1998).