# Ensuring TANGOControl System

Sergi Blanch-Torné[1], Ramiro Moreno Chiral[2]

[1] Escola Politècnica Superior, Universitat de Lleida. Spain.
`sblanch@alumnes.udl.es`
[2] Departament de Matemàtica. Universitat de Lleida. Spain.
`ramiro@matematica.udl.es`

August 11, 2013
github.Papers: 2013-08-11 (revision 362d574)

**Abstract.** [3]
Current use of TANGO is mostly in Synchrotron and a bit further in neutron a neutron source, but the community like to extend this by an explicit request of the industry. This request has been made with concern on security. Not a concern in IT environmental, that is user choose, it was about the use of cryptology to mathematically protect the system.
The goal of ensure TANGO must produce an outcome as similar as the *TLS* is for the web navigation. Must be possible to co-live with non secured access, but with a tendency to a complete transparent ensuring. Perhaps the migration process would be not as fast as we could want, specially due to the introduction of the certificates infrastructure, but as the TANGO installations are contained in the institutions, and upgrade in this way would be like any other upgrade.
Also as web navigation did, TANGO is used with instances running over different architectures and operating systems, from small embedded devices, up to very big computers. Then the objective in this ensuring process is that must work just as for the larger than for the tiny.
It is very important goal to have the TANGO implementation as Free Software, as this paper cryptography outcomes must be to have public access and auditable algorithms and sources.
**Keywords:** Cryptography[4], Distributed Systems, Secure engineering.

## 1 Introduction

- What is TANGO? It's a CORBA based middleware of distributed system used for scientific installation control.
- Event system ∅MQ(*Zero Message Queue*).
- SARDANA, TAURUS and ATK as *presentation* and *session layers* are in further work. But some aspects are related and are important.
- What is the meaning of a secure system? What is security in a distributed system?
- TANGO as an Industrial Control System (ICS), that with some extra pieces becomes a Supervisory Control And Data Acquisition (SCADA) (like ATK and also what SARDANA and TAURUS does, goes further). As will be explained in 7.3, ensure SARDANA, TAURUS and ATK is out of the scope of this paper, except for those things that defines the interaction with TANGO.
- Distributed systems transparencies [1] that TANGO complains, and which are not

**Definition 1.** *from [1], A distributed system is a collection of independent computers that appears to its users as a single coherent system*

- Security threads, policies and mechanisms. Section 2. Go further that the Locking/Access control
- Why to secure it? Trust in a peripheral firewalls is not enough. Often communications between tango installations (different tango-db) requires firewall rules to allow it, but this doesn't allow to filter by agent or by who is allowed to access the information. In practice, what is filtered is an specific computer traffic, but this breaks many of the distributed system transparencies.

[3] Partially supported by grants MTM2010-21580-C02-01 (Spanish Ministerio de Ciencia e Innovación), 2009SGR-442 (Generalitat de Catalunya).
[4] This big keyword includes proposals over *Public key*, *Elliptic Curves*, *Symmetric algorithms*, *stream cyphers*, *secret sharing* and also *Homomorphic encryption* for databases

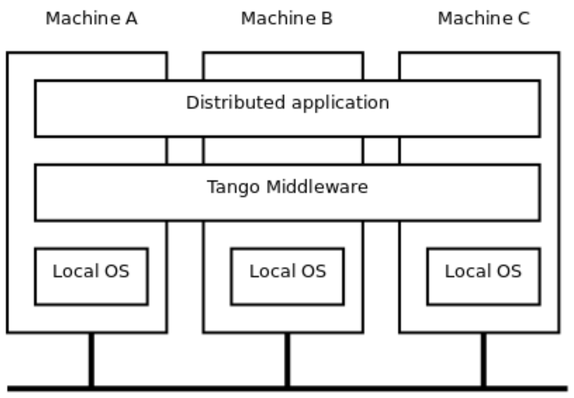

**Fig. 1.** From [1], A distributed system organized as middleware

| | |
|---|---|
| Access | Hide differences in data representation and how a resource is accessed |
| Location | Hide where a resource is located |
| Migration | Hide that a resource may move to another location |
| Relocation | Hide that a resource may be moved to another location while in use |
| Replication | Hide that a resource is replicated |
| Concurrency | Hide that a resource may be shared by several competitive users |
| Failure | Hide a faulure and recovery of a resource |
| Persistence | Hide whether a (software) resource is in memory or on disk |

**Table 1.** Distributed systems transparencies

The example of the Beamlines (read) access to (a few but crucial) accelerator information is a great example of what means a security thread.
The industrial example of "do it fast" or "finish it now" more often than thought hides an insecure system or even worst a "bugged" system.

- Following the 3 layers structure of a distributed system [1] to identify scenarios in section 2 and solutions in sections 4:
  - Agent authentication in the presentation layer (section 2.1). Possible solutions as the zero-knowledge proof (section 4.1) and Secret Sharing (section 4.2).
  - Domain layer communications protection in section 2.2. Trusted computing with elliptic curves 4.3, data communication with symmetric encryption in section 4.4 and stream cyphering in section 4.5)
  - Ensure Data layer (section 2.3) with homomorphic encryption (section 4.6) in the database.
- The price of the information and the balance between the cost to ensure and the value of the ensured goods. Section 3
- Alert on possible attacks to protect against what is already saw as a security thread, section 5, distinguishing between passive (section 5.1), active (section 5.2) and side channels (section 5.3)
- Any already saw thread should have it countermeasure (section 6) with special interest in intrusion detection (section 6.1)
- Final conclusions about ensuring protocols (section 7.1) and IT environmental security (section 7.2)

## 2 Identifying scenarios

- From the view from [1] over the distributed system transparencies, what is implemented in Tango and what is not? Is any of the "nots" necessary to ensure a quality service.
- Confidentiality (encryption and authentication): information must be disclosed only *to* the authorized and only *by* the authorized),
- Integrity (authorization): only authorized can set in the system. That is, only who is authorized can change an attribute, send a command or store a property.
- Auditory: trace who access where (extremely useful for a security breach analysis).

- In terms of security threads, which is more representative from [2] for the current use case? Three may types: *Hospital*, *Bank*, *Military Base* (from where the security threads usually comes from). Practical paranoia [3]
- Cryptosystem configuration, security levels and information classification. Section 3. Can be saw as the nowadays number of rotors from the times of the electro-mechanical machines of last century.
- Cryptosystem setup reset.
- Setup & Public-Key distribution protocols [2] sec.3.7.2
- Secret Shared schemas for (k,n)-to decrypt or (k,n)-signants. Section 4.2.
- multicast and events (∅MQ) can be scenarios of secret splitting. Section 4.2.
-
-

### 2.1 Ensuring presentation layer

- Agent authentication in a distributed system
- Ensuring communication between agents and between those agents with the user interfaces.
  - *Command*, *Attribute*, *Properties*: Authenticate who can do the *read* and *write* operations. Encrypted logging who did any change, with levels to grant access levelling.
  - This can be compared with *RFID* communication between card and readers, but adding communication in between the agents
- Deal with multicast can event subscription and emission.
-
-

### 2.2 Ensuring domain layer

- Trusted Computing and Hardware protections
- Ensure logging system
-

### 2.3 Ensuring data layer

- TANGO database access control
- Ensuring between instrumentation and the agents out of the scope of this paper. This is a very dependant on the instrumentation manufacturers. From the iso layer level view, even if the access to the hardware is not networked, the agent communication to the instrumentation is *data link layer* and this paper is focus in *transport* and above layers.
- Homomorphic Encryption for Database access
-
-

## 3 Security levels

- Security levels: Open or unclassified, confidential, Secret, Top Secret.
- remember the German standard on this levelling and the European commission "*fiche 17*" ("Exchange of EU classified information")

## 4 Communication hybrid schema

- Embedded in instrumentation, limited calculation capacity (it must behave indistinguishable if it's a huge server or an embedded board), limited bandwidth (Don't increase the current needs significantly): *very good candidate for elliptic curves (section 4.3), generalized Rijndael (section 4.4) and stream cipher (section 4.5).*
- Public-key to agreed a season key as the usual hybrid systems. This session keys shall be used for symmetric or stream cyphering.
- Session keys refresh.

- Use the Symmetric key to seed a shared PseudoRandomGenerator as a key for a stream cipher of transmitted data and listened data between talkers
- *PseudoRandomGenerator* (PRG), can be use the KeyDerivationFunction (KDF) of the Rijndael or better other possible alternatives
-
-

### 4.1 Zero-knowledge proof for authentication

- The agents in the distributed system must be authenticated to be sure that they hasn't been supplanted
-
-

### 4.2 Secret Sharing and secret splitting

- Multicast and event system. When a event is emitted, many would be subscribed, but encryption must be only made once.
- To allow some one access to some specific data, perhaps it can require the "grant" from more than one agent of the distributed system. That is, to give it the key may (k,n) must act to.
- Authorization units may be bigger than one agent. A (k,n)-signature to have only one to verify for all.

### 4.3 Elliptic curves for public key

- Set institution set of curves with different sizes for different level of secrecy (or even different curves for a separable sets in the same secrecy level). Isogeny volcanoes [4].
- Capability to reset a curve setup on any of those secrecy levels (section 3)
-
-

### 4.4 Rijndael generalization for symmetric key

- How to decide the good parameters of Rijndael? (#rounds,#rows,#columns,wordsize of the block and the key) [5]
- Current AES has advantage on 32bit processor implementation, what about 64bits
- AESWrap [6]
- Secrecy levels (section 3)
-

### 4.5 Key Derivation Functions for stream ciphering

-
-

### 4.6 Homomorphic Encryption

-
-

<

## 5 Brainstorming attacks

-
-

### 5.1 Passive attacks

- Eavesdropping
- Noise to block an alarm transmission
-
-

### 5.2 Active attacks

- Men-in-the-middle (active attacks) between agents
- Interruption: Break the public face, web site or gui. Kill a vital agent.
- Modification/Fabrication: Supplant agents.
-
-

### 5.3 Side channel attacks

-
-

## 6 Attacks countermeasures

-
-

### 6.1 Intrusion Detection

- Detection and recovery
-
-

## 7 Conclusions

- Al those fields mention on this paper requires a much further detailed paper each.
-

### 7.1 Protocols

- Protocol layers [7]
- Security architecture patterns
- Trust ring vs. trust tree (institution CA until the leaves)
- Streaming protection systems (specially for DevEncoded transmission of big images when fast acquisitions)
-
-

### 7.2 Environmental IT Security

- The weakest brick: secure the transmission but store in a plain file system
- Human behaviour and psychology.
-
-

### 7.3 Further work

- ATK/ TAURUS user authentication using PAM system (or equivalent in non unix-like systems). Any other user interface that can access tango.
- in all the algorithms on this paper this must be taken into account to minimize redesigns.

## References

1. A. S. Tanenbaum and M. van Steen, *Distributed systems, Principles and Paradigms.* Prentice Hall, 2002. International Edition.
2. R. J. Anderson, *Security engineering - a guide to building dependable distributed systems (2. ed.).* Wiley, 2008.
3. N. Ferguson and B. Schneier, *Practical Cryptography.* New York, NY, USA: John Wiley & Sons, Inc., 2003.
4. S. Blanch-Torné and R. Moreno, "Security risk associated with multiple users sharing the same elliptic curve."
5. S. Blanch-Torné and R. Moreno, "Generalised rijndael."
6. J. Schaad and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm." RFC 3394 (Informational), Sept. 2002.
7. B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* New York, NY, USA: John Wiley & Sons, Inc., 2nd ed., 1995.