Securing Tango Control System: A brain storming

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

September 1, 2013
github.Papers: 2013-09-01 (revision 900a569b)

Abstract. [3]
Current use of Tango is mostly in Synchrotron and recently extending it into a neutron source, but industry has expressed a desire to participate in the community. This industry desire has been made with concern on security. Not a concern in IT environmental, that is institution/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 with auditable algorithms and sources.
Keywords: Cryptography[4], Distributed Systems, Secure engineering.

1 Introduction

- Tango is a distributed control system with a middlewares: Corba (*omniORB*) for the synchronous and asynchronous, and ∅MQfor the event based communication.
- It is *free software* developed by a consortium of 9 research institutes. The core is licensed as *LGPLv3* and their developed agents in the distributed system are released with *GPLv3*.
- Tango is basically the *transportation* layer in the OSI schema. There is also Atk, Taurus and Sardanathat adds the skills of the *session* and *presentation* layers. They are keystone in the development of a functional system, but perhaps some aspects of their securing left out the scope of this paper.
- Nowadays tango is has an active support for development in 9 scientific research institutions, as well as other projects uses it. The industry has made a first approach in the 27^{th} tango-meeting held in the Alba Synchrotron. This request has come from Onera (French aerospace lab), but not only aerospace industry can use tango when it reaches the milestone of security. There is a broad spectrum of potential stakeholders with a huge request on security, in terms of cryptography, like energy generation and delivery, home automation, car manufacturers, and many more.

1.1 Defining distributed system security

- Define a distributed system
  - Distributed systems transparencies [1] that Tango complains, and which are nots

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

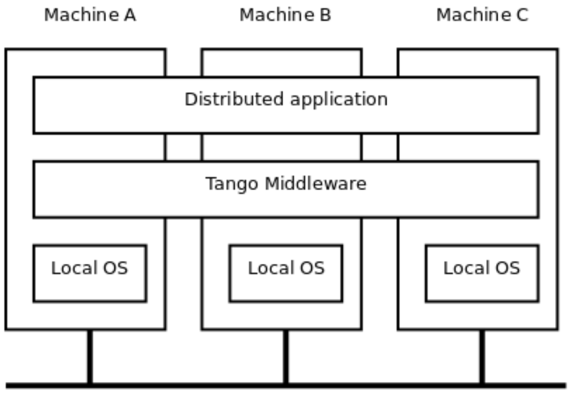

Fig. 1. From [1], A distributed system organized as middleware

- 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.

1.2 Defining security threads

- 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 (section 2).
  - 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.

1.3 Structure of the paper

- 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 4.4, data communication with symmetric encryption in section 4.5 and stream cyphering in section 4.6)
  - Ensure Data layer (section 2.3) with homomorphic encryption (section 4.7) 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)
-
-

---

[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

2 Identifying scenarios

- Securing communications between RFID cards and an authorized reader [2] would be not too different than communication between two agents in a distributed system or between and agent and the element in the presentation layer.
- From the view from [1] over the distributed system transparencies:
  - What is implemented in TANGO and what is not? And why?
  - Is any of the "nots" necessary to ensure a quality service.

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 from [1]

- Basics:
  - Confidentiality (encryption): to ensure privacy, the information must not be accessible to unauthorized parties. Information must be disclosed only *to* the authorized (and only *by* the authorized, but this is in another step),
  - Authenticity: to ensure trustiness, the information origin must be the authorized to emit it
  - 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.
  - Availability: TODO: ""
  - Non-repudiation: When a information in procured, there must be no way to cover saying "it wasn't me".
  - Auditory: trace who access where (extremely useful for a security breach analysis).
- In terms of security threads, what it's call "*thread modeling*" which is more representative from [3] for the current use case? Gather information also from [4]
  - Three may types: *Hospital*, *Bank*, *Military Base* (from where the security threads usually comes from).
  - Practical paranoia [5]:
  - Identify threads
  - Capability attack scenario
- 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.
- Setup & Public-Key distribution protocols [3] sec.3.7.2
- Cryptosystem setup reset.
- 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 4.3.
-
-

2.1 Ensuring presentation layer

- Agent authentication in a distributed system.
  - Not very different than RFID systems, it has similarities and there is a German standard [6] for travel documents (passports).
- Ensuring communication between agents and between those agents with the user interfaces (ATK and TAURUS).
  - *Command*, *Attribute*, *Properties*: Authenticate who can do the *read* and *write* operations. Encrypted logging who did any change, with levels to grant access levelling.
- Deal with multicast can event subscription and emission.
-
-

2.2 Ensuring domain layer

- Trusted Computing and Hardware protections: is an agent allowed to run on this specific machine? (what about transparencies)
- Ensure logging system
-
-

2.3 Ensuring data layer

- TANGO data base as a centralized "phone guide".
- TANGO database access control
- Ensuring between instrumentation and the agents out of the scope of this paper, often is also out of the device server developer hands. 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 *session* 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*" [7]("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 4.4), generalized Rijndael (section 4.5) and stream cipher (section 4.6).*
- 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

- 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 Secret Splitting

- Multicast and event system. When a event is emitted, many would be subscribed, but encryption must be only made once.
-
-

4.4 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 [8]. Together with the contribution work of [9], [10], [11].
- Capability to reset a curve setup on any of those secrecy levels (section 3)
- Standards about elliptic curves; International [12], [13], [14], [15], USA: [16], [17], German: [18],[19], Russian: [20]
-
-

4.5 Rijndael generalization for symmetric key

- AES contest [**?**] and the book [**?**]
- How to decide the good parameters of Rijndael? (#rounds,#rows,#columns,wordsize of the block and the key) [21]
- Current AES has advantage on 32bit processor implementation, what about 64bits
- AESWrap [22]
- Secrecy levels (section 3)
-

4.6 Key Derivation Functions for stream ciphering

-
-

4.7 Homomorphic Encryption

- Introduce the meaning of the private database query system [23]
-
-

5 Brainstorming attacks

- This has relation with the scenarios identified in section 2. Specially what concerns the security thread types [3].
- How mush work it takes to break the system? What's the value of the protected system?
-
-

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

– All those fields mention on this paper requires a much further detailed paper each.
–
–

7.1 Protocols

– Protocol layers [24]
– 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.
– ISO/IEC 27000-series
–
–

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. S. Martínez, *Protocolos de seguridad para sistemas de indentificación por radiofrecuencia.* PhD thesis, Universitat de Lleida, march 2011. Directed by: Concepció Roig and Magda Valls.
3. R. J. Anderson, *Security engineering - a guide to building dependable distributed systems (2. ed.).* Wiley, 2008.
4. N. Ferguson, B. Schneier, and T. Kohno, *Cryptography Engineering: Design, principles and practical applications.* Wiley, 2010.

5. N. Ferguson and B. Schneier, *Practical Cryptography.* New York, NY, USA: John Wiley & Sons, Inc., 2003.
6. "Bsi tr-03110: Advanced security mechanisms for machine readable travel documents."
7. "Exchange of eu classified information," 2003.
8. S. Blanch-Torné, R. Moreno, and F. Sebé, "Security risk associated with multiple users sharing the same elliptic curve." Draft.
9. J. Valera, *Volcales de l-isogenias de curvas elípticas.* PhD thesis, Sistemas Informáticos. Escola Politècnica Superior. Universidad de Lleida, Sept 2011. Directed by: Josep M. Miret.
10. R. Moreno, *Subgrupos de Sylow de las curva ellúpticas definidas sobre cuerpos finitos.* PhD thesis, Universitat Politècnica de Catalunya, 2005. Directed by: Anna Rio and Josep M. Miret.
11. R. Tomàs, *Volcans d'isogenies de corbes el·lítiques: Aplicacions criptogràfiques en targetes intel·ligents.* PhD thesis, Universitat de Lleida, march 2011. Directed by: Josep M. Miret and Daniel Sadornil.
12. A. Jivsov, "Ecc in openpgp." Internet Draft, OpenPGP working group at ieft.
13. S. Blake-Wilson, N. Bolyard, V. Gupta, C. Hawk, and B. Moeller, "Elliptic curve cryptography (ecc) cipher suites for transport layer security (tls)," May 2006.
14. "Sec 1. standards for efficient cryptography group: Elliptic curve cryptography."
15. "Sec 2. standards for efficient cryptography group: Recommended elliptic curve domain parameters."
16. "Ieee p1363 standard specifications for public key cryptography," January 2000.
17. "Ansi x9.62, public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ecdsa)."
18. "Ecc brainpool standard curves and curve generation," October 2005.
19. "Bsi tr-03111: Elliptic curve cryptography, version 2.0."
20. "Information technology. cryptographic data security. signature and verification processes of digital signature.," 2001. Gosudarstvennyi, Standard of Russian Federation, Government Committee of the Russia for Standards.
21. S. Blanch-Torné, R. Moreno, F. Sebé, and M. Valls, "Generalised rijndael." Draft.
22. J. Schaad and R. Housley, "Advanced Encryption Standard (AES) Key Wrap Algorithm." RFC 3394 (Informational), Sept. 2002.
23. D. B. nad Craig Bentry, S. Halevi, F. Wang, and D. J. Wu, "Private database queries using somewhat homomorphic encryption," *International Association for Cryptologic Research*, June 2013.
24. B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C.* New York, NY, USA: John Wiley & Sons, Inc., 2nd ed., 1995.