Paper Number: IAEA-CN-220-151

EPR Number (for papers with IAEA/SG authors): 697

Paper Title:

The IAEA's Universal Instrument Token

Authors:

Ingo Naumann (1), Keith Morgan (1), Christoph Brunhuber (1), Bernie Wishard (1), Andreas Schwier (2), Frank Thater (2)

Affiliation(s):

(1) IAEA, (2) CardContact, Germany

Address of Main Author's affiliation:

Vienna International Centre
1400 Vienna, Austria

Email Address of Main Author:

i.naumann@iaea.org

Abstract:

The IAEA continuously seeks to improve the harmonization of security approaches across safeguards equipment. The protection of digital safeguards data is based on several principles: a) signing of data in measurement devices using standard public/private-key-based signature generation, b) storage of secret keys on certified, tamper-protected cryptographic devices, and c) use of well-established cryptographic algorithms and protocols based on global standards and internationally recognized cryptographic libraries. This paper discusses a cryptographic token, the Universal Instrument Token (UIT), which constitutes the core element of the overall architecture for digitally signing safeguards data. This architecture supports the above principles and is compliant with the IAEA's information security policies and guidelines. Another important consideration is that the UIT must be deployed across a wide range of operating systems and hardware architectures, necessitating the use of open-source software (OSS) for all software-related parts.

The UIT is permanently connected to the measuring device (usually via the USB port) and requires complex hardware drivers and middleware components. Identifying OSS based, mature and ready-for-use smart card drivers and tools that are compatible with a range of operating systems was a major challenge. Reliable and well-established cryptographic libraries reside at the core of every information-security application. Different types of review software, typically software products used at IAEA Headquarters in Vienna but also occasionally in nuclear facilities, need to contain some specific software modules in order to verify digital signatures attached to data. Finally, enrolment tools are also required, which generate private keys and certify their corresponding public counterparts using the IAEA's internal Certification Authority.

In 2014, the roll-out of the UIT has raised the security of IAEA instrument data authentication to a level currently considered unlikely to be defeated, provided that the correct procedures are observed.

Full paper:

A. Introduction

The IAEA applies various technical measures, referred to as nuclear safeguards, to verify the correctness and completeness of declarations made by States with respect to nuclear material and activities. Safeguards equipment enables IAEA inspectors to collect and verify safeguards data. Safeguards equipment requires inherent security mechanisms in order to protect the integrity and the confidentiality of safeguards data. However, in order to ensure usability and maximize data availability, a certain harmonization throughout a number of disparate safeguards systems is required. This requirement represents a major challenge.

B. Scenario

Data security is an important feature of unattended and remote monitoring safeguards systems. Systems of this type, which are installed for extended periods at facilities and are visited only periodically by inspectors, transmit data over unsecured transmission paths between the components of different systems and between these systems and IAEA Headquarters [1]. The protection of digital safeguards data is based on several principles: a) signing of data in measurement devices using standard public/private-key-based signature generation, b) storage of secret keys on certified, tamper-protected cryptographic devices, and c) use of well-established cryptographic algorithms and protocols based on global standards and internationally recognized cryptographic libraries. An internal security policy mandates the application of these principles [2].

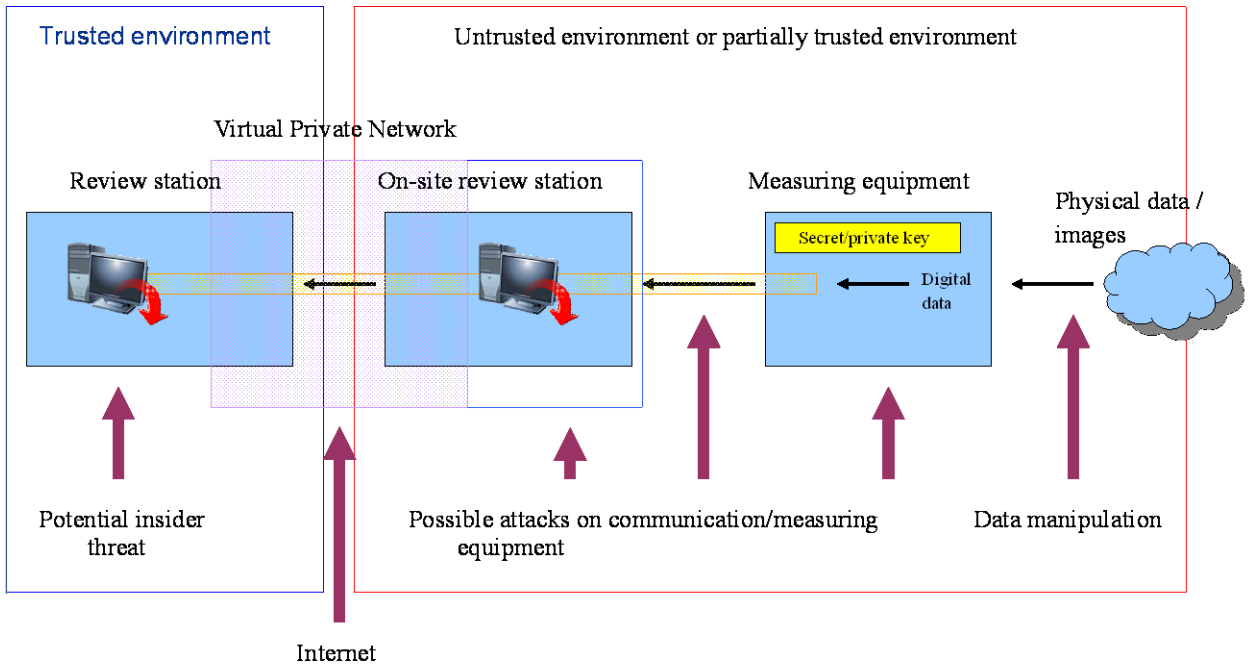

Figure 1: Typical deployment scenario for unattended safeguards instruments [3]

The UIT is a cryptographic token permanently connected to the measuring device, usually via the USB port. It requires a set of complex hardware drivers and middleware components. Different types of review software, typically software products used at IAEA headquarters in Vienna but occasionally also in nuclear facilities, need to contain specific software modules in order to verify digital signatures attached to data.

C. The Smart Card HSM

The SmartCard-HSM [4] is a remotely manageable secure key store for RSA and ECC cryptographic keys and is currently the only UIT in use by the IAEA. Unlike commonly known USB memory sticks that store files, the SmartCard-HSM is a USB device that contains a secure micro-controller with protected memory and a cryptographic engine. Cryptographic keys are generated, stored and used internally, never leaving the secure enclosure of the device. The micro-controller embedded in a SmartCard-HSM is the same tamper-resistant security controller that is used for banking cards, national ID cards or passports. It is specifically designed to protect sensitive information from unauthorized disclosure using direct (e.g. brute-force) or indirect (e.g. side-channel) attacks. The security of the micro-controller, the cryptographic library and the operating system (although notably not the application running on it) is independently evaluated and certified under the Common Criteria (CC) scheme at Evaluation Assurance Level EAL 5 "*Semi-formally Designed and Tested*" (level 5 out of 7). Additional vulnerabilities evaluated at higher levels assume attackers with sophisticated capabilities. This ensures that the private keys have a unique property, namely that they exist only in the physically controllable device. This approach presents a major advantage over the use of software keys, as these can be copied and misused without being noticed. The "Heartbleed" bug [5] recently discovered in OpenSSL [6] demonstrates the risk of using cryptographic keys in software.

As well as cryptographic keys in a SmartCard-HSM being protected from disclosure, their use is also controlled. The SmartCard-HSM requires an authentication code to be presented before a cryptographic operation can be performed. The authentication code must be at least 48 bits long (up to a maximum of 128 bits). Commonly, a 6-digit PIN code is assigned as an authentication code. To prevent a brute force attack on the authentication code through attempting to enter all possible combinations, the SmartCard-HSM includes an error counter that limits the number of wrong entries. If the permitted number of wrong entries is exhausted, then the authentication code is blocked, and keys are temporarily suspended.

When using the SmartCard-HSM in an instrument, secure storage of the authentication code presents a new problem requiring resolution; the authentication code must be protected against unauthorized disclosure, as an adversary might otherwise steal the SmartCard-HSM and make use of its keys. One possible method of application is to have an operator enter the authentication code whenever the instrument is put into operation. The code need only be entered once, after which cryptographic operations can be performed until power is disconnected or the device is reset (by logging out). A second option is to change the authentication code to a pairing code during installation of the SmartCard-HSM. In this case, the instrument will generate a secure pairing code known only to the instrument, effectively binding the SmartCard-HSM to the instrument. The pairing code must then also be held securely within the instrument. The pairing code could be generated as random value, or derived from non-public information available in the instrument.
A third option is to require contact with a central authentication server whenever the authentication code needs to be presented. A PKI built into the SmartCard-HSM will allow the server to identify the device and to establish an authenticated and confidential communication channel with the device. Using this channel, the authentication code is presented by the server and the device unlocked, enabling local cryptographic operations.

D. Real-time and Integrated STream-Oriented Remote Monitoring (RAINSTORM) and UIT

D.1. The sc-hsm-embedded Library

The generally recognized cross-platform application programming interface (API) for accessing smart card readers is *Personal Computer/Smart Card* (PC/SC). The OSS PC/SC library OpenSC/pcsc-lite includes support for the SmartCard-HSM token. While this library supports myriad features and is suitable for many systems, the IAEA often fields resource-constrained embedded systems, where it may not be possible to use OpenSC. For these systems, the Card Terminal API (CT-API), a lightweight API, is more suitable. As part of the UIT effort, the IAEA commissioned CardContact to develop the sc-hsm-embedded library, which includes a CT-API for the SmartCard-HSM token. The relevant source code is published on GitHub [7].

D.2. The sc-hsm-ultralite library

In many scenarios, the IAEA requires the UIT to perform only a private-key operation to digitally sign data. In these scenarios, using cryptography libraries such as OpenSSL, cryptlib [8] or Crypto++ [9] might represent overkill, as their function for creating a digital signature is simply hashing data and generating an ASN.1 enclosure. For this reason, the IAEA remote monitoring (RM) programming team developed the sc-hsm-ultralite library, a companion to the sc-hsm-embedded library. The sc-hsm-ultralite library uses a clever trick to patch template signature files simply, thereby eliminating the need for a cryptography library such as OpenSSL. More details can be found in [10].

E. Open Source

There are strong arguments in favour of using open source software in security products. Obvious bugs can more easily be spotted if the source code is reviewed by various testers with different skill sets. Many security product developers have made their source code publicly available [11]. Reliable and well-established cryptographic libraries (which are often distributed under an open source license) reside at the core of every information security application.

The wide range of applications envisaged for the UIT requires open-source based, mature and ready for use smart card drivers and tools compatible with a range of operating systems. Any UIT implementation must be compliant with the generic open-source CCID driver [12] as well as with the OpenSC [13] libraries which implement the PKCS#11 API [14]. This is how the UIT is different from those cryptographic tokens used previously for safeguards instruments. In terms of software (and hardware) sustainability, open-source implies that software modules can be replaced, software may be re-used for other platforms, and that tokens may be replaced by another product if no longer supported (second-source products have already been selected and initially tested).

F. Applications

F.1. LMCV

The laser mapping for containment verification (LMCV) system uses a laser triangulation technique to perform an accurate mapping of closure welds on containers. Once a container has been loaded with nuclear

material and its lid closed and welded, the closure weld is scanned using LMCV. The weld scan is used to derive a unique signature for a specific container, which remains valid provided the weld is not modified.
An embedded system, running Windows 7 Embedded on a single-board PC, serves as the core of the LMCV system. In this scenario, the UIT uses the cryptlib and OpenSC [13] OSS libraries. The LMCV also includes an embedded HTTP RAINSTORM server which serves as the interface with the IAEA's remote monitoring data retrieval system.

F.2. OLEM

The On-Line Enrichment Monitor (OLEM) is a system designed to provide continuous enrichment measurements at gaseous centrifuge enrichment plants. The OLEM design incorporates the latest IAEA methods and approaches for physical and data security including the use of the UIT. Recently, a field trial of OLEM prototype systems has been conducted — the OLEM design and measurement approach, along with feedback from the field trial, will be presented in a separate paper at the IAEA Safeguards Symposium 2014 [15]. The OLEM software runs on a Technologic Systems TS-7800 embedded ARM single board computer (SBC) running Debian Sarge Linux. The UIT implementation uses the sc-hsm-ultralite library. An Apache HTTP server plus RAINSTORM PHP serves as the interface with the IAEA's remote monitoring data retrieval system. Although the TS-7800 has sufficient computing power to use cryptlib plus OpenSC, one function of the OLEM was to explicitly prove new IAEA technologies such as sc-hsm-ultralite and RAINSTORM [10][15].

F.3. NGAM

The Next Generation ADAM Module (NGAM) is a data acquisition module which, similarly to the OLEM, is permanently installed in nuclear facilities. It contains two ARM microprocessors running the SMX real-time operating system. An ARM9 processor handles external communication, transferring data, state of health, and hosting the HTTP interface. In this case, the UIT implementation uses a custom port of the sc-hsm-ultralite library to the SMX operating system. An SMX embedded HTTP server serves as the interface with the IAEA's remote monitoring data retrieval system.

G. Enrolment

Finally, enrolment tools are also required to generate private keys and certify their corresponding public counterparts using the IAEA's internal Certification Authority (CA). This process is largely transparent to the authorized technician who prepares UITs for use in the field. A script-based tool initializes the token automatically, generates an individual PIN code, triggers key generation and, upon authentication by the technician, requests certification of the token's public key by the dedicated subordinate Certificate Authority (SubCA). The IAEA tracks the ownership and location of UITs using the safeguards equipment inventory management system (EQUIS).

Conclusions:

This paper discusses a cryptographic token, the UIT, which constitutes the core element of the overall architecture for authenticating safeguards data. This architecture supports a number of key overarching principles and is compliant with the IAEA's information security policies and guidelines. An important consideration is that the UIT must be implemented across a wide range of operating systems and hardware architectures, which necessitates the use of OSS for all software-related parts. In 2014, the roll-out of the UIT has raised the security of IAEA instrument data authentication to a level commensurate with all current security policies and procedures.

References:

[1] IAEA, Safeguards Techniques and Equipment: 2011 Edition, ISBN 978-92-0-118910-3, http://www-pub.iaea.org/books/iaeabooks/8695/Safeguards-Techniques-and-Equipment-2011-Edition

[2] IAEA SGTS, Information Security Requirements for the Development of IAEA Safeguards Equipment, Policy and Guideline, Document No. SG-PL-11859, Version 1, 2013-01-16

[3] Naumann, Ingo; Wishard Bernard, Liguori Cesare: Data Security Risk Management for Nuclear Safeguards – Getting Your Threat Model Right, International Nuclear Materials Management Conference 2011

[4] CardContact, The SmartCard-HSM, product website, http://www.smartcard-hsm.com/

[5] Heartbleed, Wikipedia Page, http://en.wikipedia.org/wiki/Heartbleed

[6] The OpenSSL Project, Cryptography and SSL/TLS Toolkit, https://www.openssl.org/

[7] Light-Weight CT-API and PKCS\11 Library for Using the SmartCard-HSM in Embedded Systems, https://github.com/sc-hsm-clone/sc-hsm-embedded

[8] Gutmann, Peter: cryptlib, Cryptographic Library, http://www.cs.auckland.ac.nz/~pgut001/cryptlib/

[9] Wei Dai: Crypto++ Library, http://www.cryptopp.com/

[10] Brunhuber, Christoph; Morgan, Keith: The Evolution of RAINSTORM, Symposium on International Nuclear Safeguards: Linking Strategy, Implementation and People, Vienna, Austria, 20 - 24 October 2014

[11] Anderson, Ross: Security Engineering, Second Edition, 2008, Wiley Publishing Inc., ISBN 978-0-470-06852-6

[12] CCID Free Software Driver, https://pcsclite.alioth.debian.org/ccid.html

[13] OpenSC Tools and Libraries for Smart Cards, https://github.com/OpenSC/OpenSC/wiki

[14] PKCS11 #11: Cryptographic Token Interface Standard, http://www.emc.com/emc-plus/rsa-labs/standards-initiatives/pkcs-11-cryptographic-token-interface-standard.htm

[15] Ely, James; Lebrun, Alain; Pochet, Thierry; Younkin, J.; Garner, James, March-Leuba, J.; Smith, Eric: On-Line Enrichment Monitor (OLEM) - Supporting Safeguards at Enrichment Facilities, Symposium on International Nuclear Safeguards: Linking Strategy, Implementation and People, Vienna, Austria, 20 -24 October 2014