A Trusted Information Agent for Security Information and Event Management

Luigi Coppolino
Epsilon S.r.l.,
Naples, Italy
luigi.coppolino@epsilonline.com

Michael Jäger
Technische Hochschule Mittelhessen
Giessen, Germany
michael.jaeger@mni.thm.de

Nicolai Kuntze and Roland Rieke
Fraunhofer Institute for
Secure Information Technology
Darmstadt, Germany
{nicolai.kuntze,roland.rieke}@sit.fraunhofer.de

***Abstract*—This paper addresses security information management in untrusted environments. A security information and event management system collects and examines security related events and provides a unifying view of the monitored system's security status. The sensors, which provide the event data, are typically placed in a non-protected environment at the boarder of the managed system. They are exposed to various kinds of attacks. Compromised sensors may lead to misjudgement on the system's state with possibly serious consequences. The particular security requirements arising from these problems are discussed for large scale critical infrastructures. The main contribution of this paper is a concept that provides trusted event reporting. Critical event sources are holistically protected such that authenticity of the security related events is guaranteed. This enables better assessment of the managed system's reliability and trustworthiness. As a proof of this concept, the paper presents an exemplary realisation of a trustworthy event source.**

***Keywords*-reliability aspects of security information and event management systems; trusted event reporting; trusted android application; critical infrastructure protection.**

I. Introduction

Security information and event management (SIEM) systems provide important security services. They collect and analyse data from different sources, such as sensors, firewalls, routers or servers, and provide decision support based on anticipatory impact analysis. This enables adequate response to attacks as well as impact mitigation by adaptive configuration of countermeasures. The project MASSIF [1], a large-scale integrating project co-funded by the European Commission, addresses these challenges with respect to four industrial domains: (i) the management of the Olympic Games information technology (IT) infrastructure [2]; (ii) a mobile phone based money transfer service, facing high-level threats such as money laundering; (iii) managed IT outsource services for large distributed enterprises; and (iv) an IT system supporting a critical infrastructure (dam) [3].

Common to these use cases is the requirement to prove that a measured value has been acquired at a certain time and within a specified "valid" operation environment. Authenticity of such measures can only be assured together with authentication of the used device itself, it's configuration, and the software running at the time of the event.

In geographically dispersed infrastructures, various equipment, including the critical sources of event data, is often placed in non-protected environments. Therefore, attackers are able to access and manipulate this equipment with relative ease[4].

Proposition 1. *When physical access to the sensoring devices cannot be inhibited, an effective security solution must address detection of manipulations.*

Manipulated equipment can be used to hide critical conditions, generate false alerts, and in general cause misjudgement on system's state. Wrong assumptions about a system's state in turn can lead to false decisions with severe impact on the overall system.

Proposition 2. *Whenever a certain control decision is made, the input information that presumably led to it must be authentic.*

As a consequence, the system has to assure that all safety critical actions using sensor data must only use authentic sensor data. The question, which measurements and system control decisions are critical to the overall system behaviour, cannot be answered independently of the concrete system and application context determined.

Proposition 3. *A risk assessment of the deployed monitoring capabilities is necessary.*

Contribution: By means of a representative example, namely a hydroelectric power plant in a dam, we analyse security threats for critical infrastructures and justify the relevance of the postulated propositions for adequate security requirements. Further, the paper presents both, a concept and a prototypical implementation for trustworthy event reporting. Digital signatures obviously can provide authenticity and integrity of recorded data [5]. However, a signature gives no information on the status of the measurement device at the time of measurement. Our solution, the *trusted information agent* (TIA), is based on trusted computing technology [6] and integrates industry approaches to the attestation of event reporter states. This approach provides a certain degree of trustworthiness and non-repudiation for the collected events, which can be used as a basis for risk assessment according to Proposition 3.

The paper is structured as follows. Section II gives an overview of the related work. In Section III we introduce the exemplary application scenario. We then elicit a number of specific security requirements from the application scenario and justify the propositions for our concept in Section IV. Based on these requirements, we address a solution for our propositions and describe the concept and a prototypical implementation of a trusted information agent in Section V. Finally, the paper ends with conclusions and an outlook in Section VI.

II. Related Work

The paper addresses the integration of Trusted Computing concepts into SIEM systems for critical infrastructures based on examples from a hydroelectric power plant in a dam.

Security information and event management technology provides log management and compliance reporting as well as real-time monitoring and incident management for security events from networks, systems, and applications. Current SIEM systems' functionalities are discussed in [7]. SIEM systems manage security events but are not concerned with the trustworthiness of the event sources. Security requirements analysis and an authenticity concept for event sources is, however, the main topic of this paper. The specification of the application level security requirements is based on the formal framework developed by Fraunhofer SIT [8]. In this framework, systems are specified in terms of sequences of actions and security properties are constraints on these sequences. Applying the methods of this framework, we derive security requirements for the event sources in the dam scenario.

Dam monitoring applications with *automated data acquisition systems* (ADAS) are discussed in [9], [10]. Usually, an ADAS is organised as a *supervisory control and data acquisition* (SCADA) system with a hierarchical organisation. Details on SCADA systems organisation can be found in [11], [12]. In the majority of cases, SCADA systems have very little protection against the escalating cyber threats.

Compared to traditional IT systems, securing SCADA systems poses unique challenges. In order to understand those challenges and the potential danger, [4] provides a taxonomy of possible cyber attacks including cyber-induced cyber-physical attacks on SCADA systems.

Trusted Computing technology standards provide methods for reliably verifying a system's integrity and identifying anomalous and/or unwanted characteristics [6]. An approach for the generation of secure evidence records was presented in [13]. This approach, which is the basis for our proof-of-concept implementation, makes use of established hardware-based security mechanisms for special data recording devices. Our communication protocols extend the Trusted Network Connect (TNC) [14] protocol suite. We use the open source implementation of IF-MAP presented in [15].

III. Application Scenario

Our analysis of security threats for critical infrastructures is based on examples from a hydroelectric power plant in a dam. The dam scenario is typical for critical infrastructures in many respects. On the one hand, it is a layered system with intra- and cross-layer dependencies, and, on the other hand, there are various other sources of complexity; several distinct functionalities influence controlling and monitoring activities. Moreover, different components, mechanisms, and operative devices are involved, each one with different requirements in terms of produced data and computational loads.

A dam might be devised for a multitude of purposes and its features are strictly related to the aims it is built for, e.g., food water supplying, hydroelectric power generation, irrigation, water sports, wildlife habitat granting, flow diversion, or navigation. Since a dam is a complex infrastructure, a huge number of parameters must be monitored in order to guarantee safety and security. Which parameters are actually monitored, depends on the dam's structure and design (earthfill, embankment or rockfill, gravity, concrete arch, buttress), the purpose (storage, diversion, detention, overflow), and the function (hydroelectric power generation, water supply, irrigation).

Table I
Dam instrumentation sensors

Sensor	Parameter or physical event
Water level sensor (*WLS*)	Current water level (*wl*)
Inclinometer/Tiltmeter (*TM*)	Earth or wall inclination or tilt (*tm*)
Crackmeter (*CM*)	Wall/rock crack enlargement (*cm*)
Jointmeter (*JM*)	Joint shrinkage (*jm*)
Piezometer	Seepage or water pressure
Pressure cell	Concrete or embankment pressure
Turbidimeter	Fluid turbidity
Thermometer	Temperature

Table I lists some of the most commonly employed sensors together with a brief explanation of their usage. The heterogeneity of currently used devices is a relevant challenge in the dam process control: they range from old industrial control systems, designed and deployed over the last 20 years and requiring extensive manual intervention by human operators, to more recently developed systems, conceived for automatic operations (SCADA). Indeed, the trend of development is toward increasingly automated dam control systems. While automation leads to more efficient systems and also prevents operating errors; on the downside, it poses a limit to human control in situations, where an operator would possibly foresee and manually prevent incidents.

Modern automated systems support remote management and also provide for centralised control of multiple infrastructures. As an example, the Terni hydroelectric complex, located about 150 Km in the north of Rome, is composed by 16 hydroelectric power plants, three reservoirs (Salto, Turano and Corbara), and one pumping plant, all of them supervised by a single remote command post located at Villa Valle.

As a severe disadvantage, increased automation and remote

Table II
SECURITY RELATED SCENARIOS AND THE RESPECTIVE MONITORING

Monitored Event	Impact	Detection
Changes in the flow levels of the seepage channels	Seepages always affect dams (whatever their structure and design are). Seepage channels are monitored to evaluate the seepage intensity. A sudden change in flow levels could show that the structure is subject to internal erosion or to piping phenomena. This event can be the cause of dam cracks and failures	By inserting into the channel a weir with a known section the depth of water (monitored by using a water level sensor) behind the weir can be converted to a rate of flow.
Gates opening	Intake gates are opened to release water on a regular basis for water supply, hydroelectricity generation, etc. Moreover spillways gates(aka overflow channels) release water (during flood period) so that the water does not overtop and damage or even destroy the dam. Gates opening must be operated under controlled conditions since it may result in: i) Flooding of the underlying areas; ii) Increased rate of flow in the downstream that can ultimately result in a catastrophic flooding of down-river areas.	A tiltmeter (angle position sensor) can be applied to the gate to measure its position angle.
Changes in the turbine/infrastructure vibration levels	Increased vibrations of the infrastructure or the turbines in a hydro-powerplant can anticipate a failure of the structure. Possible reasons for such event include: i) earthquakes (Fukushima, Japan, a dam failure resulted in a village washed away); ii) unwanted solicitations to the turbines (Sayano-Shushenskaya, Siberia, 75 dead due to a failure of the turbines in a hydro-powerplant).	Vibration sensors can be installed over structures or turbines to measure the stress level they are receiving.
Water levels overtake the alert thresholds	Spillway are used to release water when the reservoir water level reaches alert thresholds. If this does not happen the water overtops the dam resulting in possible damage to the crest of the dam (Taum Sauk hydroelectric power station).	This event can be used to detect unexpected discharges. Water level can also be correlated to other parameters to detect anomalous behaviour (e.g., not revealed gate opening).

control raise a new class of security-induced safety issues, i.e., the possibility that cyber attacks against the IT layer of the dam ultimately result in damage to people and environment.

Dam monitoring aims towards identifying anomalous behaviour related to the infrastructure. Table II summarises a list of possible scenarios illustrating the necessity of monitoring specific parameters.

IV. SECURITY REQUIREMENTS ANALYSIS

We use a model-based approach to systematically identify security requirements for the dam application scenario. Specifically, *authenticity* can be seen as the assurance that a particular action has occurred in the past. For a formal specification of the application-level authenticity requirements, we use Definition 1, which is taken from [8].

Definition 1. $auth(a,b,P)$*: Whenever an action b happens, it must be authentic for an Agent P that in any course of events that seem possible to him, a certain action a has happened.*

In [8] a *security modelling framework* (SeMF) for the formal specification of security properties was presented. Requirements are defined by specific constraints regarding sequences of actions than can or can not occur in a system's behaviour. Actions in SeMF represent an abstract view on actions of the real system, which models the *interdependencies* between actions and ignores their functionality. An action is specified in a parameterised format, consisting of the action's name, the acting agent and a variable set of parameters:

$$actionName(actingAgent, parameter1, parameter2, ...)$$

Table III lists the dam scenario actions used for our security requirements analysis.

Table III
DAM ACTIONS

Action	Description
$sense(WLS,wl)$	Measurement of the water level.
$sense(TM,tm)$	Measurement of the tilt.
$sense(CM,cm)$	Measurement of the crack enlargement.
$sense(JM,jm)$	Measurement of the joint shrinkage.
$sense(PP,power)$	Measurement of voltage and current in the power grid. The power plant PP sends commands ppc to the dam control station depending on these measurements.
$sense(SDC,wdc)$	Measurement of the water discharge on the penstock gates PG.
$sense(PG,open)$	Reporting of the state of the penstock gates.
$display(DCS,X)$	Display X at the dam control station, with $X \in \{wl,tm,cm,jm,ppc,wdc,open\}$.
$activate(Admin,cmd)$	Decision of the administrator, which command shall be triggered.
$exec(PG,cmd)$	Command to be executed by penstock gates.

We now analyse some possible misuse cases, which have been reported in the scenario deliverable [16] of the MASSIF project.

Water level sensor compromise: The attacker takes control of the water level sensors and uses them to send spoofed measurements to the dam control station (*DCS*). This hides the real status of the reservoir to the dam administrator (*Admin*). In this way, the dam can be overflown without alarms being raised by the monitoring system.

From this, we get the requirement that the water level measures have to be authentic for the administrator when they are displayed at the dam control station. More formally, we get the authenticity requirement:

$$auth(sense(WLS, wl), display(DCS, wl), Admin) \quad (1)$$

Tiltmeter compromise: The attacker takes control of the tiltmeter sensors and uses them to send false measurements to the dam control station, thus hiding the real status of the tilt of the dam's walls to the dam administrator. An excessive tilt may lead to the wall's failure. The respective authenticity requirement is:

$$auth(sense(TM, tm), display(DCS, tm), Admin) \quad (2)$$

Crackmeter / jointmeter compromise: The attacker has access to one of the crackmeters or jointmeters deployed across the dam's walls and takes control of it. So the attacker can weaken the joint or increase the size of the crack at the wall's weak point without any alarm being raised at the monitoring station, which leads to the following authenticity requirements:

$$auth(sense(CM, cm), display(DCS, cm), Admin) \quad (3)$$
$$auth(sense(JM, jm), display(DCS, jm), Admin) \quad (4)$$

These examples show that some elementary security requirements can be derived directly from misuse cases. In general, however, information flows between systems and components are highly complex, especially when organisational processes need to be considered. Hence, not all security problems are discoverable easily. In order to achieve the desired security goals, security requirements need to be derived systematically.

An important aspect of a systematic security evaluation is the analysis of potential information flows. A method to elicit authenticity requirements by analysis of functional dependencies is described in [17]. From the use case descriptions, atomic actions are derived and set into relation by defining the functional flow among them. The action-oriented approach considers possible sequences of actions (control flow) and information flow (input/output) between interdependent actions. Actions of interest are specifically the *boundary actions*, which represent the interaction of the system's internals with the outside world. From a functional dependency graph, the boundary actions can be identified. We now give an example of security information flows by a use case of the dam scenario [16].

On demand electric production: The Dam Control Station feeds an hydroelectric turbine, connected to the dam by means of penstocks, for producing electric power on demand. The turbine and hydroelectric power production depends on the water discharge in the penstocks. By analysing the parameters of the command received by the dam control station, we can infer that the safety critical actions are the opening and closing actions of the penstock gates (*PG*).

An identification of functional dependencies reveals that the dam control activity makes use of the (i) current water level, (ii) the state of the gates joined to the hydroelectric power plant, (iii) the gates openness, and, (iv) the discharge through the penstocks. Figure 1 shows the dependency graph of this use case. The decision of the administrator, which command shall be triggered, depends on the displayed measurements. The dashed line indicates that there is no direct functional dependency.

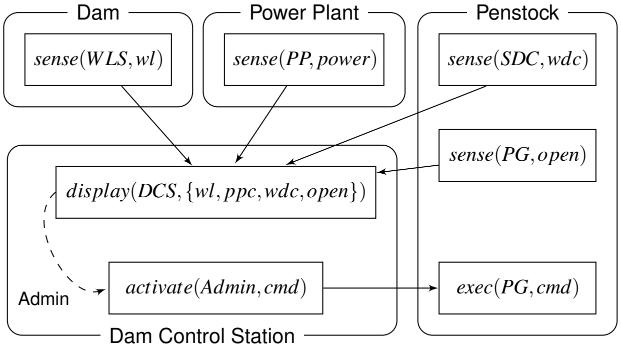

Figure 1. Functional dependencies: *On demand electric production*

An analysis of the dependencies depicted in Figure 1 leads to the following conclusion: The control display values are derived from the measurements of *wl*, *power*, *wdc*, and *open*. From this, we conclude that, in addition to the water level *wl* (1), the measurements of *power*, *wdc*, and *open* have to be authentic. More formally:

$$auth(sense(PP, power), display(DCS, ppc), Admin) \quad (5)$$
$$auth(sense(SDC, wdc), display(DCS, wdc), Admin) \quad (6)$$
$$auth(sense(PG, open), display(DCS, open), Admin) \quad (7)$$

Furthermore, the activation of the penstock command by the administrator has to be authentic for the penstock gate when executing it.

$$auth(activate(Admin, cmd), exec(PG, cmd), PG) \quad (8)$$

So the authenticity requirements for the use case described in Figure 1 are given by: (1) and (5)–(8).

In summary, the analysis of the use case and misuse cases of this critical infrastructure scenario shows that the overall function of the system requires authenticity of measurement values for several sensors, namely (1) – (7). In that sense, the dam scenario is a prime example for the relevance of the requirements postulated in Proposition 1 and 2. It is evident that further types of security requirements are needed in order to cover important liveness properties such as *availability* of necessary information at a certain place and time. In some cases also *confidentiality* of certain information may be required. These requirements are important but not in the scope of the work presented here.

V. Trusted Information Agent

The usefulness of monitoring large systems clearly depends on the observer's level of confidence in the correctness of the available monitoring data. In order to achieve that confidence, network security measures and provisions against technical faults are not enough. As stated above, unrevealed manipulation of monitoring equipment can lead to serious consequences. In order to improve the coverage of this type of requirements in a SIEM framework, we now describe a concept and a prototypical implementation of a trusted information agent (TIA).

A. *Trust Anchor and Architecture*

As shown in Section IV, protection of the identity of the device for measurement collection is necessary. Furthermore, the lack of control on the physical access to the sensor node induces strong requirements on the protection level.

By a suitable combination of hardware- and software-level protection techniques any manipulations of a sensor have to be revealed. In addition to the node-level protection, network security measures are needed in order to achieve specification-conformant behaviour of the sensor network, e.g., secure communication channels that protect data against tampering. This paper is not intended to discuss network security, neither protection of hardware components. We rather concentrate on the important problem of clandestine manipulations of the sensor software.

A commonly used technique to reveal manipulation of a software component is software measurement: Each component is considered as a byte sequence and thus can be measured by computing a hash value, which is subsequently compared to the component's reference value. The component is authentic, if and only if both values are identical. Obviously, such measurements make no sense if the measuring component or the reference values are manipulated themselves. A common solution is to establish a chain of trust: In a layered architecture, each layer is responsible for computing the checksums of the components in the next upper layer. At the very bottom of this chain a dedicated security hardware chip takes the role of the trust anchor or "root of trust".

Trusted Computing [6] offers such a hardware root of trust providing certain security functionalities, which can be used to reveal malicious manipulations of the sensors in the field. Trusted Computing technology standards provide methods for reliably checking a system's integrity and identifying anomalous and/or unwanted characteristics. A trusted system in this sense is build on top of a Trusted Platform Module (TPM) as specified by the Trusted Computing Group (TCG). A TPM is hardened against physical attacks and equipped with several cryptographic capabilities like strong encryption and digital signatures. TPMs have been proven to be much less susceptible to attacks than corresponding software-only solutions.

The key concept of Trusted Computing is the extension of trust from the TPM to further system components. This concept is commonly used to ensure that a system is and remains in a predictable and trustworthy state and thus produces authentic results. As described above, each layer of the chain checks the integrity of the next upper layer's programs, libraries, etc. On a PC, for example, the TPM has to check the BIOS before giving the control of the boot-process to it. The BIOS then has to verify the operating system kernel, which in turn is responsible for the measurement of the next level. Actually, a reliable and practically useful implementation for PCs and systems of similar or higher complexity is not yet feasible. Sensoring and measuring devices, however, typically have a considerably more primitive architecture than PCs and are well-suited for this kind of integrity check concept. Even for modern sensor-equipped smartphones, able to act as event detectors, but having the same magnitude of computing power that PCs had a few years ago, an implementation of the presented concept is possible. A prototypical implementation is presented in more detail now.

B. *Proof of Concept: Base Measure Aquisition*

Figure 2 depicts the architecture of the TIA.

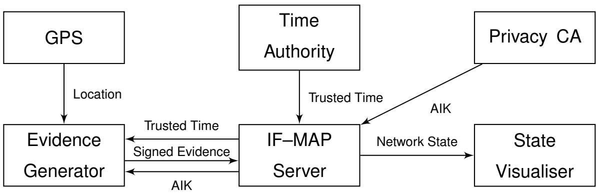

Figure 2. TIA architecture

The main component of the TIA is the *evidence generator* (EG), which collects base measures and provides the measurement functions used to produce derived measures. Furthermore, the EG supports the processing of measures from external sensors, e.g., location data from a GPS module. The EG is expected to operate in unprotected environments with low physical protection and externally accessible interfaces such as wireless networks and USB access for maintenance. A necessary precondition to guarantee authenticity of the measures, is a trustworthy state of the measurement device. To meet this requirement, the EG is equipped with a TPM as trust anchor and implements a chain of trust [18]. As explained above, revelation of software manipulations is based on the comparison between the software checksums and the corresponding reference values. This comparison may be done locally within the node (self-attestation) or by a remote verifier component (remote attestation) [6].

The EG submits the collected measures digitally signed to an IF-MAP [14] server, which acts as an event information broker. During initialisation, the EG obtains two credentials

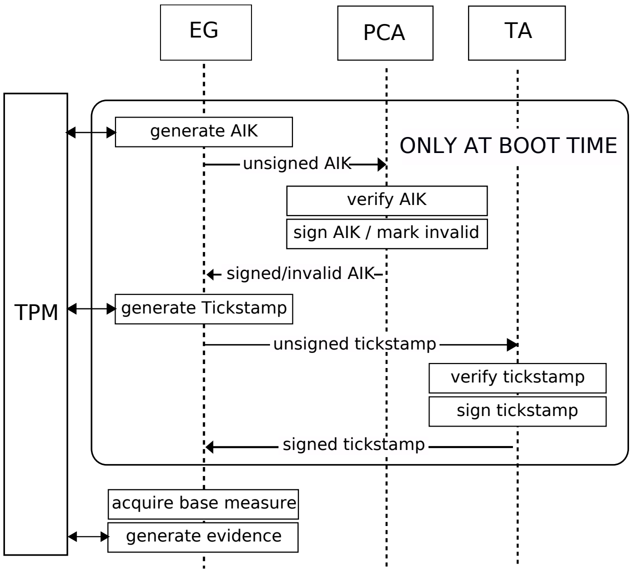

Figure 3. Process model

from trusted third-party services for signature purposes. Figure 3 depicts the boot-time interaction between the EG and those services, and the role of the TPM in this interaction.

An Attestation Identity Key (AIK) is used to sign measurement results in a manner that allows verification by a remote party. The Privacy Certification Authority (PCA) issues a credential for the TPM-generated AIK. The certified AIK is, henceforth, used as an identity for this platform. According to TCG standards, AIKs cannot only be used to attest origin and authenticity of a trust measurement, but also, to authenticate other keys and data generated by the TPM. However, the AIK functionality of a TPM is designed primarily to support remote attestation by signing the checksums of the EG's software components, while signing arbitrary data is, in fact, not directly available as a TPM operation. We have shown elsewhere, how to circumvent this limitation [19]. Hence, we are able to use TPM-signatures for arbitrary data from the EG's sensors.

Any TPM is equipped with an accurate timer. Each event signature includes the current timer value. However, the TPM timer is a relative counter, not associated to an absolute time. A *time authority* (TA) issues a certificate about the correspondence between a TPM timestamp (tickstamp) and the absolute time. The combination of tickstamp and TA-certificate can be used as a trusted timestamp. Alternatively, another trusted time source, such as GPS, could have been used.

Putting it all together, a measurement record includes arbitrary sensor data, a TA-certified time stamp, and a hash value of the EG's software components. The record itself is signed by the TA-certified AIK.

Figure 4 shows a prototype EG, which has been implemented based on the Android smartphone platform. This platform has been selected for various reasons. Modern smartphones are equipped with a variety of sensors such as GPS, gyro sensor, electronic compass, proximity sensor, accelerometer, barometer, and ambient light sensor. Furthermore, photos, video and sound can be regarded and processed as event data. Moreover, Android is well-suited as a software platform for future embedded devices.

The TPM-anchored chain of trust is extended to the linux system and linux application layers by using the Integrity Measurement Architecture (IMA), which is integrated into any stock linux kernel as a kernel module. The Android application layer is based on libraries and the Dalvik Virtual Machine (VM). While the linux kernel layer can check the Android system libraries and the VM, Android applications run on top of the VM and are invisible to the kernel. Thus, we built a modified VM, which extends the chain of trust to the Android application level by computing the applications' checksums. A timestamp-based variant of remote attestation provided by the TPM is used for the verification of the node authenticity. All communication is based on the Trusted Network Connect (TNC) [14] protocol suite, which offers advanced security features, such as dedicated access control mechanisms for TPM-equipped nodes.

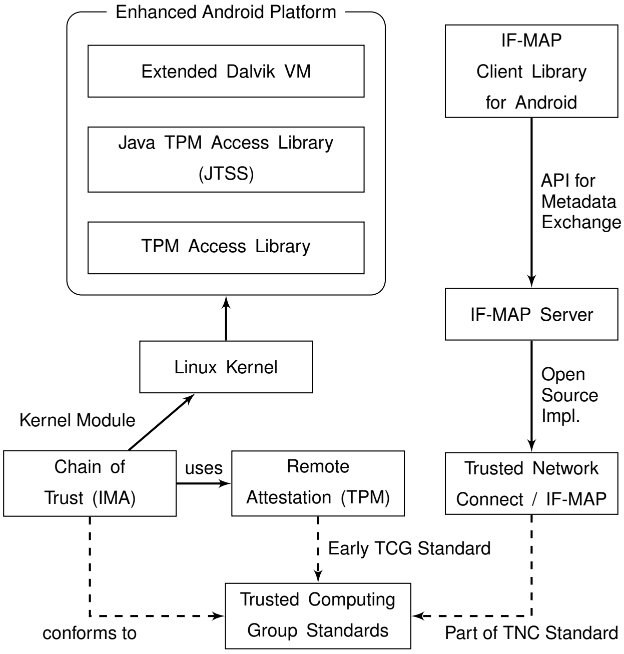

Figure 4. Technical building blocks

VI. Conclusion and Future Work

In geographically dispersed infrastructures the critical sources of event data are often placed in non-protected environments. Attackers can thus easily manipulate these sensors and thereby hide critical conditions, generate false alerts, and in general cause misjudgement on system's state. By exemplary analysis of a typical application scenario we have demonstrated that this can lead to false decisions with severe impact on the overall system. In order to prevent such threats, we presented a concept for holistically protected critical event sources by assuring a trustworthy state of the measurement devices. This enables better assessment of the managed system's reliability and trustworthiness.

As a proof of this concept, the paper presented an exemplary realisation of a trusted information agent based on trusted computing technology. Planned next steps include a detailed analysis on the impact on scalability and bandwidth of different schemes to generate evidence using this architecture. Especially, the correlation of independent events may allow for improvements but also requires trustworthy schemes to cryptographically link various events to one evidence record. Also, the hardware-based security functionalities can be improved with respect to scalability and performance. Further, suggestions to improve standards for future hardware security modules, are planned.

Acknowledgement

Luigi Coppolino and Roland Rieke developed the work presented here in the context of the project MASSIF (ID 257475) being co-funded by the European Commission within FP7. Nicolai Kuntze developed the work presented here in the context of the project ESUKOM (ID 01BY1052) which is funded by the German Federal Ministry of Education and Research.

References

[1] "Project MASSIF website," 2012. [Online]. Available: http://www.massif-project.eu/

[2] E. Prieto, R. Diaz, L. Romano, R. Rieke, and M. Achemlal, "MASSIF: A promising solution to enhance olympic games IT security," in *International Conference on Global Security, Safety and Sustainability (ICGS3 2011)*, 2011.

[3] L. Coppolino, S. D'Antonio, V. Formicola, and L. Romano, "Integration of a System for Critical Infrastructure Protection with the OSSIM SIEM Platform: A dam case study," in *SAFECOMP*, ser. Lecture Notes in Computer Science, F. Flammini, S. Bologna, and V. Vittorini, Eds., vol. 6894. Springer, 2011, pp. 199–212.

[4] B. Zhu, A. Joseph, and S. Sastry, "Taxonomy of Cyber Attacks on SCADA Systems," in *Proceedings of CPSCom 2011: The 4th IEEE International Conference on Cyber, Physical and Social Computing, Dalian, China*, 2011.

[5] J. Choi, I. Shin, J. Seo, and C. Lee, "An efficient message authentication for non-repudiation of the smart metering service," *Computers, Networks, Systems and Industrial Engineering, ACIS/JNU International Conference on*, vol. 0, pp. 331–333, 2011.

[6] C. Mitchell, *Trusted Computing*. Iet, 2005.

[7] M. Nicolett and K. M. Kavanagh, "Magic Quadrant for Security Information and Event Management," Gartner Reasearch, May 2010.

[8] S. Gürgens, P. Ochsenschläger, and C. Rudolph, "Authenticity and provability - a formal framework," in *Infrastructure Security Conference InfraSec 2002*, ser. LNCS, vol. 2437. Springer, 2002, pp. 227–245.

[9] M. Parekh, K. Stone, and J. Delborne, "Coordinating intelligent and continuous performance monitoring with dam and levee safety management policy," in *Association of State Dam Safety Officials,Proceedings of Dam Safety Conference 2010*, 2010.

[10] B. K. Myers, G. C. Dutson, and T. Sherman, "Utilizing Automated Monitoring for the Franzen Reservoir Dam Safety Program," in *25th USSD Annual Meeting and Conference Proceedings (2005).*

[11] L. Coppolino, S. D'Antonio, and L. Romano, "Dependability and resilience of computer networks (scada cybersecurity)," in *CRITICAL INFRASTRUCTURE SECURITY: Assessment, Prevention, Detection, Response*. WIT press, in press.

[12] L. Coppolino, S. D'Antonio, L. Romano, and G. Spagnuolo, "An intrusion detection system for critical information infrastructures using wireless sensor network technologies," in *Critical Infrastructure (CRIS), 2010 5th International Conference on*, sept. 2010, pp. 1 –8.

[13] J. Richter, N. Kuntze, and C. Rudolph, "Security Digital Evidence," in *2010 Fifth International Workshop on Systematic Approaches to Digital Forensic Engineering*. IEEE, 2010, pp. 119–130.

[14] T. C. Group, "TCG Trusted Network Connect – TNC IF-MAP Binding for SOAP Version 2.0," www.trustedcomputing.org, 2010.

[15] J. v. H. I. Bente, J. Vieweg, "Towards Trustworthy Networks with Open Source Software," in *Horizons in Computer Science Volume 3*. Nova Science Publishers Inc., T. S. Clary (Eds.), 2011.

[16] M. Llanes, E. Prieto, R. Diaz, , L. Coppolino, A. Sergio, R. Cristaldi, M. Achemlal, S. Gharout, C. Gaber, A. Hutchison, and K. Dennie, "Scenario requirements (public version)," MASSIF Project, Tech. Rep. Deliverable D2.1.1, 2011.

[17] A. Fuchs and R. Rieke, "Identification of Security Requirements in Systems of Systems by Functional Security Analysis," in *Architecting Dependable Systems VII*, ser. LNCS. Springer, 2010, vol. 6420, pp. 74–96.

[18] N. Kuntze and C. Rudolph, "Secure digital chains of evidence," in *Sixth International Workshop on Systematic Approaches to Digital Forensic Engeneering*, 2011.

[19] N. Kuntze, D. Mähler, and A. U. Schmidt, "Employing trusted computing for the forward pricing of pseudonyms in reputation systems," in *Axmedis 2006, Proceedings of the 2nd International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution, Volume for Workshops, Industrial, and Application Sessions*, 2006.