Secure digital chains of evidence

Nicolai Kuntze, **Carsten Rudolph (corresponding author)**
Fraunhofer Institute for Secure Information Technology (SIT)
Rheinstrasse 75, 64295 Darmstadt, Germany
{nicolai.kuntze|carsten.rudolph}@sit.fraunhofer.de

Abstract

Computers, mobile phones, embedded devices and other components of IT systems can often be easily manipulated. Therefore, in forensic use of digital evidence it is necessary to carefully check that the probative force of the evidence is sufficient. For applications where critical processes can lead to disputes and resolving disputed relies on digital evidence one open question is how to build the system in a way that secure digital evidence is available. This paper introduces the notion of secure digital chains of evidence *and proposes a high-level architecture for systems that can provide such chains of evidence. Finally, possible building blocks are explored for the realisation of a distributed and heterogeneous system with support for secure digital chains of evidence.*

Keywords

Secure digital evidence, event correlation, digital chains of evidence, trusted computing

1. Introduction

For many types of digital data records or logging data for processes it is obvious that they can potentially be relevant as digital evidence in the case of disputes [9]. Computing and storing pictures taken by digital cameras (e.g. for speeding tickets) or critical workflows on enterprise service buses are examples of processes that somehow need to be produced, documented and stored in a secure way in order to enable their use at court [14]. In general, data records or log files are not sufficiently protected to prevent manipulations. Nevertheless, in many cases it would be possible to design systems in a way that these data records can represent valid digital evidence even if strong requirements would be imposed on what can be seen as *valid* evidence. One existing approach uses hardware-based security (i.e. the Trusted Platform Module TPM [10]) to secure digital evidence and to bind evidence records to relevant parameters. Such parameters include the status of the device (e.g. software, hardware, configuration), and also parameters such as the time of producing the evidence record, location of the device, or certificates stating the validity of these parameter values. Nevertheless, such a protection of single evidence records is insufficient if evidence shall prove that a particular process has occurred (e.g. a service-based workflow) or has been followed (e.g. in producing the evidence record). The single evidence record only securely documents one single event, while documenting a process requires looking at various events occurring on different devices or places. Thus, several evidence records are produced and can be relevant. While all of these records can be individually protected by the existing scheme, in addition it is necessary to link these records. This linking needs to show that each evidence record belongs to the right step of a particular instance of the process to be documented. Consequently, rather than presenting only one secure evidence record it becomes necessary to create a secure chain of evidence. This paper presents an approach to create such a secure chain of evidence for the case where the actual events are distributed. First a notion of *secure digital chains of evidence* is introduced and a high-level architecture defines possible components for collecting and storing this type of secure forensic data. Finally, a number of more technical building blocks is identified that can be used to construct a system with the potential to provide information as secure digital chains of evidence.

2. Secure digital chains of evidence

Various digital information is available for use in forensics. Some of this data is especially generated and stored to provide information about what has happened in a particular IT-based system such as an enterprise network, but also in systems with embedded components such as traffic control systems, rail signalling systems, or road toll systems. In all these systems information on many events is already logged for different purposes. All theses trails are in principle available for forensic use and some of this information can also be linked to create chains of evidence in order to show different things. Examples include some individual person (e.g. a user of a system) initiating a particular process, some computations having occurred on a particular device, network messages exchanges, or data records stored. In principle, this information can be used to create digital chains of evidence [6], [4]. However, digital information can be subject to various types of inaccuracies, errors and manipulations. Computers can have intermediate different behaviour by installing and removing software, changing configuration, or by booting in a different state. All these changes can occur without leaving any obvious traces. Digital records can be changed after they have been produced. Even digital signatures only provide parts of the solutions and require proper protection of the private key. Protecting the chain of custody is a challenge for digital data and can be easier if secure digital chains of evidence exist. In addition to the protection of the evidence records themselves, chains of evidence need some way of linking information in the single evidence records. For current available information in IT systems this linking is very difficult and often impossible.

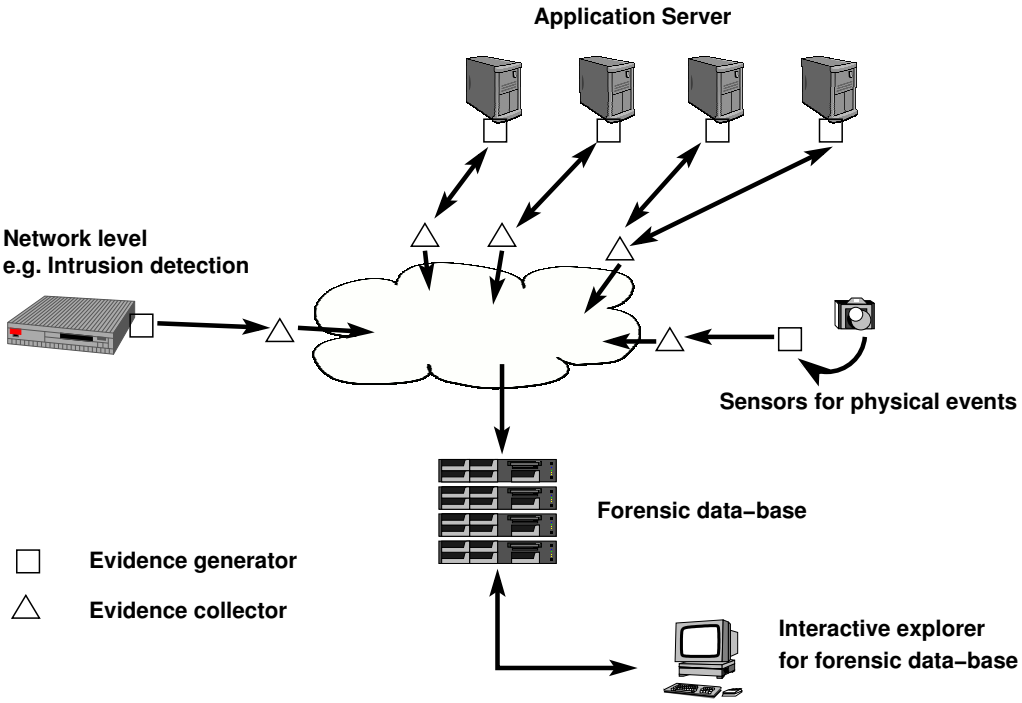

Fig. 1. High-level architecture for collecting secure digital evidence

Improving this situation requires to analyse processes, identify critical parts and related available information and probably store additional linked and protected information.

Clearly, for many existing information and communication technology systems it is not feasible to foresee which processes and data could be subject to future forensic evaluation. Nevertheless, a lot of systems are currently deployed where the risk for disputes in the case of malfunction is very high either because of high financial losses or even due to injuries or deaths of human beings (e.g. in rail signalling systems). In such systems it makes perfect sense to identify in advance all critical processes and to explore which chains of evidence can be used to show that and how a particular process has occurred. Once these processes and events are identified, the system can be build in a way that secure evidence information is collected and also the linking of events to reconstruct critical processes is supported.

A digital chain of evidence consists of digital evidence records. For a secure chain each single evidence record needs to be secure. Therefore, a *secure evidence record* is defined as a set of digital information that is securely bound to all relevant parameters necessary to verify the validity of the information. Obviously, the actual required protection levels and also the relevant parameters totally depend on the scenario and can therefore not be defined in general. Some possible parameters can be the status and configuration of a device, time, network addresses, user information, or the location. Now, a *secure digital chain of evidence* is defined as a set of secure evidence records with clearly defined links between the data records and an overall predication that can be concluded from the chained evidence.

3. A high-level architecture for collecting secure digital evidence

The previous section has introduced a notion of secure digital chains of evidence. Before identifying possible building blocks for actually realising the creation and collection of data for such secure digital chains of evidence, this section introduces a high-level architecture for collection of secure digital evidence.

Obviously, it is infeasible for many real-life systems to explicitly identify, create, collect and store all possible or potentially useful digital chains of evidence. It is more feasible to identify critical events to be documented together with parameters linking events. Thus, the goal of a pro-active collection of digital evidence should be to create and store a graph of linked secure evidence records in a way that a path through the graph can represent a secure chain of evidence. Note that not all paths will represent a useful chain of evidence. Such a systems as depicted in Figure 1 consists of the following elements:

- *Evidence generators* create data records and securely bind them to relevant parameters e.g. by digital signatures using hardware-based security [13].
- *Evidence collectors* can add semantic information to the evidence record and make it available for distribution and storage [12], [11].
- A *Forensic data-base* stores all secure evidence records as a graph structure representing the links between different events.

- Actual creation of chains of evidence is an interactive process using the *Interactive forensic data-base explorer*.

4. Building-blocks for secure evidence generation

In [13] an approach for the generation of individual secure evidence records was presented. This approach is based on established hardware-based security mechanisms and is applicable to special devices producing data records with possible forensic use. The presented architecture includes a sketch of the process needed to ensure the security of the evidence record. Figure 2 shows the different steps of this process.

Even in the of individual data records (e.g. images taken by a digital camera) it becomes that the security of the collected evidence records depends on a number of steps in the process that also need to be documented. Thus, even in this relatively simple scenario a number of events can produce additional digital evidence and thus creating a digital chain of evidence consisting of evidential data representing events of very different types. The following paragraph introduces several building blocks that can be used to build a system that supports the construction of secure digital chains of evidence. The roles of the different building blocks correspond to the different components of the architecture for collecting secure digital evidence. Figure 4 shows one possible realisation of an IT infrastructur supporting secure digital chains of evidence.

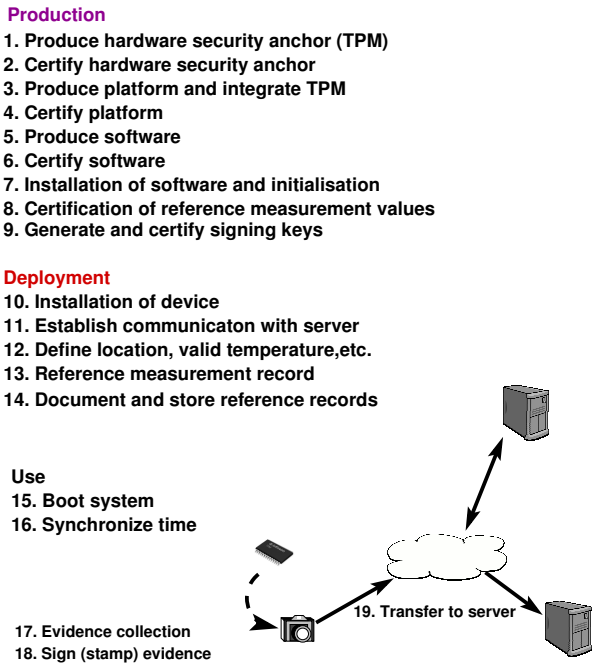

Fig. 2. Process to establish secure evidence records

4.1. Secure evidence generator using Trusted Computing technology

The core part of the architecture is the actual generation of secure digital evidence. One possible approach is the use of hardware-based security mechanisms in particular Trusted Computing and the Trusted Platform Module (TPM) as specified by the Trusted Computing Group (TCG). A TPM provides a variety of security functionality. For the secure evidence generation those parts of the TPM are essential that identify the device, bind data to the identity of the device, and provide authentic reports on the current state of the device. In the context of digital cameras the feasibility of the use of TPMs for the protection of digital images has already been proposed [13] and demonstrated [18]. The following paragraphs revise the most important parameters to be secured.

4.1.1. Proof of software and configuration. One important aspect in the generation of digital evidence is the status of the device used in the process. The used software and configuration to produce evidence needs to be presented and linked to the individual record. One simple scheme hereby is to include software name and version number as a simple string of text in each evidence record. This first (and often used) approach allows for uncertainties with respect to updates and various attacks on the evidence records. Just naming the software is not sufficient if the device can be manipulated. Stronger means of protection are therefore required to reliably document the software and configuration of the particular evidence generator.

To provide proof on the actual state of the evidence generator a trustworthy reporting in the device is required. The Trusted Computing standard introduces a core root of trust for measurement which establishes the foundation to report on the status by creating a chain of trust [2]. This chain of trust can be reported to external entities to allow for a verification of the evidence generator. This verification process is called Remote Attestation.

Application of remote attestation allows for a session based or per record scheme. The session based approach relies on an initial attestation of the system and a session bound to the individual evidence generator and status. Every evidence record is then cryptographically bound to this session and therefore to a particular system state. The second, per record scheme involves an attestation process for each evidence record. As in the basic remote attestation an external random number generator is involved longer delays and higher bandwidth utilisation is to be expected. More advanced schemes as presented in [16] allowing for scalable attestation schemes are to be applied.

One important feature of proposed incorporation of Trusted Computing is the lightweight infrastructure necessary during run-time compared with a traditional Public Key Infrastructure system. Given the assertions of the hardware a single key will not be revealed. Therefore it is not required to maintain certificate revocation lists and to check them before a certificate is accepted. It is also not possible to move a certain identity of an evidence generator (represented by its key) from one device to

the next. These inherent features allow for typical deployment scenarios like embedded, resource constraint environments.

4.1.2. Evidence record order. Time is a very important parameter in the forensic evaluation of evidence records. In most cases it is absolutely necessary to have a more or less precise but reliable information on when a particular event such as the generation of an evidence record) has happened. Therefore, evidence generators need to bind evidence records to timing information. For digital chains of evidence representing a particular process this time information is essential to reconstruct the order of events in the process.

In the case of a single device a monotonic counter (e.g. a clock) can be used to issue for each record a time stamp to ensure the order. To ensure the probative force of the time stamp the time needs a cryptographically strong binding to the evidence record and further it also needs to be bind to time information to be issued by a trustworthy time authority. Especially the latter proves to be a strong requirement and is mostly solved by trusted third parties producing time stamps in specially secured and certified installations. However, direct online time-stamping by a trusted time source is way too inefficient for most reasonable evidence generators. In particular in the case of embedded systems and/or high numbers of records such a remote time stamping would create a bottleneck. Thus, a secure evidence generator should to be able to produce time stamps on its own. Of course, the remote and probably more reliable time-stamping service can be used to synchronise the local time of the evidence generator with an official time source.

A feasible approach is to introduce a certified monotonic and timed tick-counter and a mechanism for digital signatures and secure key storage to provide for time-stamps. The tick-counter as well as the cryptographic functionality should be protected by hardware-means in order to prevent manipulations by malicious software. To achieve a trustworthy local time stamp authority the protected hardware has to provide for a shielded monotonic counter which incremented in a certain interval. To ensure this particular interval a monitoring of the accuracy of the external clock of the hardware incrementing the counter is also required. Such techniques are available in many standard PC architectures equipped with a Trusted Platform Module (TPM) and can also be provided via Smart Cards [15]. The TPM identifies each session starting with the power up of the device with a new random number created within the TPM and is then able to time stamp arbitrary data. These time stamps can then be used to identify the order of the generated evidence records.

Considering distributed evidence generators it is required to establish a link between the individual monotonic counters. Linking two counters results in a measure to translate between the respective local counter value into the other, which can be denoted as $n_1 := n_2 + -\mathit{offset}$. The offset is the expected uncertainty in the association due to delays on the network and computational overhead. Figure 3 depicts a scheme to associate one counter to the other. Herby generator g_1 sends to g_2 a tick stamp on a random value TS. TS is then tick stamped by g_2 and send back to g_1. The returned $stamp_{g_2}(TS)$ is then again stamped by g_1 and the resulting evidence is stored. Due to differences between the initial stamp of g_1 and the latter one the maximum offset can be calculated and an attack on the response time of g_2 can be recorded and documented. To extend this scheme to a mutual link the stamped result of g_2 is to be sent back to g_2. g_2 then can stamps the received message itself to document the delay between g_1 and g_2.

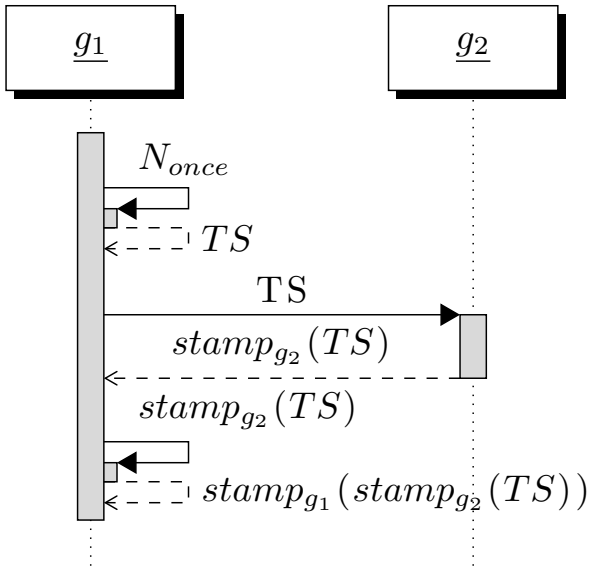

Fig. 3. Trustworthy counter linking

Depending of the particular infrastructure it can be necessary to link the time between several nodes belonging to one process. It can be efficient to use one central node to establish bilateral links between each node and the one central node and by doing this indirectly link all timing ticks of all nodes. Nevertheless, in highly distributed systems it can also be necessary to establish a peer-to-peer structure without any central node. In these cases, more intelligent management procedures need to be set-up in order to ensure that all events in a process can be ordered by their time tick information. Such an approach is particularly important in mobile ad-hoc networks with critical functionality. In such networks the synchronisation of time ticks can be combined with other existing distributed security mechanisms [8].

4.1.3. Real Time Association. The previous paragraph has described that it must be possible to associate the correct order with events represented in a digital chain of evidence. For this requirement it is sufficient to know the time an event happened in relation to other events. Another stronger requirement for valid evidence can be to know the real time an event has happened. In principle, real time information can be established in the same way. However, synchronisation can be quite loose in the case where only the order of events is important. For real time associations there are two major

differences:

- First, the time synchronisation needs to be between the evidence generator and a reliable time source, such as a certified time-stamping service. Indirect synchronisation via other nodes increases the delay and thus the inaccuracy of the time synchronisation.
- Second, the accuracy of the time synchronisation becomes relevant.

Several protocols have been proposed for the use of the TPM tick counter to represent real time information[13], [18] and also the general properties of time synchronisation protocols and algorithms have been analysed [1]. Common to all approaches is that within the digital chain of evidence also information on the time synchronisation has to be recorded. This information contains the original time stamp of the time authority but also information on the accuracy of the synchronisation, the time intervals associated with the tick counter in the evidence generator and also information to keep track of resets of the tick counter. The TPM provides support for all these parameters. Ome example is the tick counter in the TPM that comes with a tick nonce that identifies tick counter sessions. Tick stamps with the same nonce belong to the same session without a reset of the counter. Thus, once the tick nonce has changed, a new synchronisation with the authentic time source is necessary. It should be noted that in contrast to the proposed use of the tick counter in [18] the change of the tick nonce cannot reliable identify a re-boot of the device. As long as the TPM has power, the tick counter will not be explicitly reset during a re-boot of the device.

4.1.4. Other parameters. Various other parameters can become relevant for forensic use of data records. However, not all of them are readily available and can be easily or efficiently be included in the secure digital chain of evidence. As an example we briefly discuss the geographical location of the device at the time of evidence generation. Different techniques exist to determine the location. Depending on the technology used, the accuracy of the location information differs. More and more devices support the Global Position System (GPS). If adequate GPS signals are available, the GPS localisations can be in the range of 5 meters for consumer-grade GPS devices under open skies. The results within buildings under trees or with other obstacles range from 10 meters to no position information at all [17]. Other approaches, such as triangulation in wireless LAN or localisation within the GSM network are usually in general less accurate, although they can be quite accurate in special scenarios such as indoor localisation [5]. Parameters can have very different characteristics and in developing systems to support digital chains of evidence one has to make sure that all relevant parameters are covered and maybe additional sensors are installed to enable the collection of these parameters. Examples can include the temperature of the device (very deep or high temperatures can lead to corrupted evidence data), the orientation of a camera, or the names of users currently active on a multi-user device. Determining relevant parameters and their validity ranges and their meaning for the chain of evidence is a very important step in the engineering of such systems.

4.1.5. Evidence records. In addition to the different parameters related to the evidence records it is also important to generate and secure the evidence records themselves. Obvious security measures such as digitally signing the evidence records and binding these signatures to identity and other parameters described above can be used to guarantee for authenticity and integrity of the data records in a way that these security properties are not violated by distributing evidence records and storing them in the forensic data-base. However, there can also be additional security requirements. One important factor is privacy. Evidence records can potentially contain information in individual persons or other secret information e.g. business related. Therefore, also the confidentiality of evidence records shall not be neglected. Suitable encryption should be used and other best practises for dealing with confidential data shall be applied. Maintaining the confidentiality of digital data is a non-trivial task. Very recent research has shown that for example data stored on solid-state disks and other flash memory and deleted can still be retrieved even after several rounds of re-writing with different data.

4.2. Event Collection and Event Correlation

In the majority of application scenarios a certain decision of the system is not based on a singular event but by the correlation of several factors defining a certain high-level event. A certain high-level event is defined by a set of low-level events and a process for the correlation of the low level events. Low level events are simple occurrences like fire wall incidents, configuration changes but maybe as well as the photographic evidence of the speeding camera. The process defines how these events have to interact based on an operational model that a certain high level event is to be produced.

In non-complex use cases as presented for example in digital cameras [18] the evidence is generated by a single measurement agent and only the evidence records of the particular device are required. To achieve a certain probative force also in complex scenarios it is required also to provide data on other aspects of the IT system not directly related to the evidence in question but documenting the trustworthy state of the infrastructure. For scalability reasons in bandwidth or computational restricted applications it is also required to split one event into a set of corresponding events.

The task of correlating events in order to construct digital chains of evidence is closely related to the task of security information and event management (SIEM) in IT networks. Expensive commercial frameworks can support SIEM processes and also open-source frameworks are available [1]. Nevertheless, in particular correlation of events from different levels and contexts is still very difficult in the SIEM context. In

1. http://www.alienvault.com/community.php?section=Home

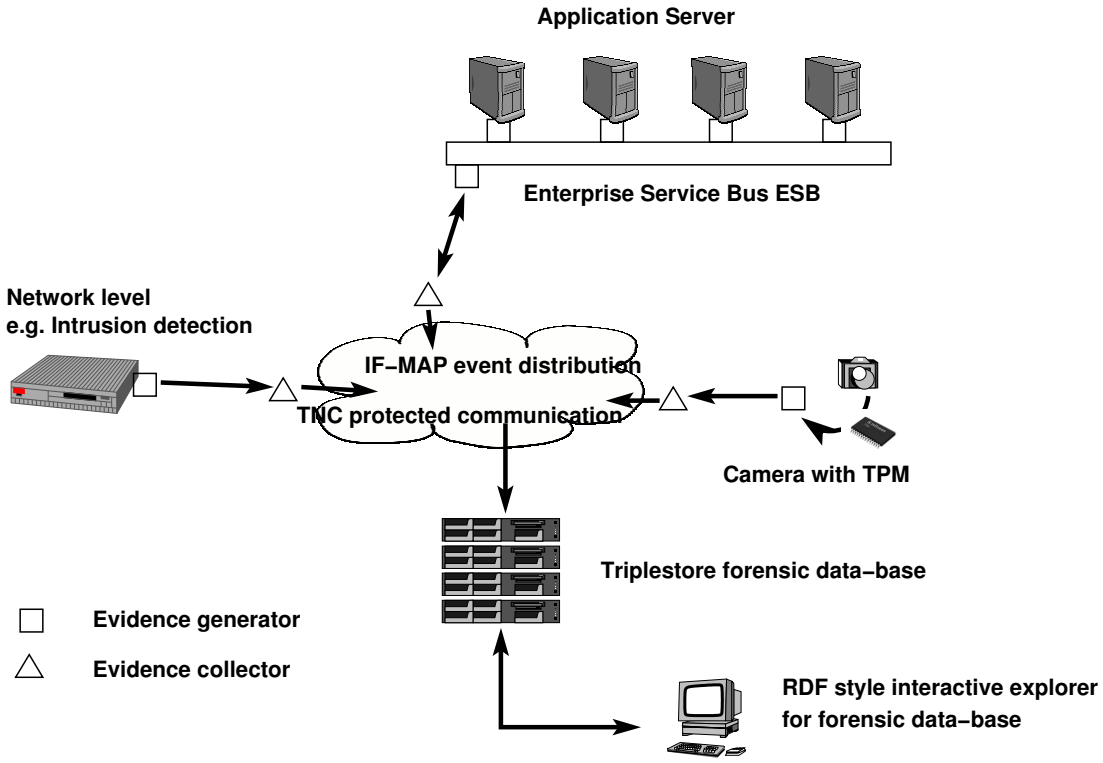

Fig. 4. Example of an infrastructure for collecting secure digital evidence

SIEM systems the correlations need to be explored at run-time to be able to induce appropriate reactions on misbehaviour. The situation is different for forensic use. Digital chains of evidence usually don't have to be created at run-time. In the case of forensic use of event information it is sufficient to collect the event information and maybe add additional semantic information at run-time. The actual evaluation of the correlations between events in order to produce a chain of evidence only occurs in the case of disputes or other forensic evaluations. Thus, for forensic use a bigger effort needs to spend on carefully choosing event information to be stored and to define parameter relating events.

To correlate data it is required to provide for an infrastructure supporting the processing of events from various sources in a unified structure representing the relations between the individual events. The interface to meta-data access points (IF MAP) [3] as an established industry standard and part of the Trusted Network Connect (TNC) protocol stack defines and supports event distribution and correlation in the domain of network access control but can easily be extended to support other types of events and create event graphs representing relations between different types of events. Figure 5 shows how an IF MAP server takes on the central role of distributing meta-level information in an IT infrastructure.

4.3. Forensic data-base

For the forensic data-base two main characteristics can be identified:

- The data-base can potentially contain huge numbers of more or less related evidence records representing graph structures where paths through the graphs can be chains of evidence also using semantic information on the events. Thus, the data-base needs to be scalable and it needs to support the exploration of large graphs with semantically enriched information.
- Evidence records need to be securely stored for a potentially long time. Storage of evidence needs to comply to regulations for long-term archiving.

For the first characteristics the so-called triplestore seems to be particularly suitable. Evidence records can consist of relatively short statements about what has happened. Triplestore is a special purpose database type developed for the use in semantic web frameworks. For a system supporting the proactive generation of secure digital chains of evidence, semantics of evidence records in terms of events happened need to be known already at design time. The resulting structure is very similar to what can be expressed as resource triples within the resource Description Framework (RDF) specified by the W3C (http://www.w3.org/RDF/). Some triplestore databases are very powerful with support for billions of triples loaded at a speed of more than 1.000 triples per second. Further, they support a variety of graph representations and rule-based exploration.

Also for the area of secure long-term archiving a variety of solutions exist. According to the national regulations with regard to long-term aspects of the probative force of a certain

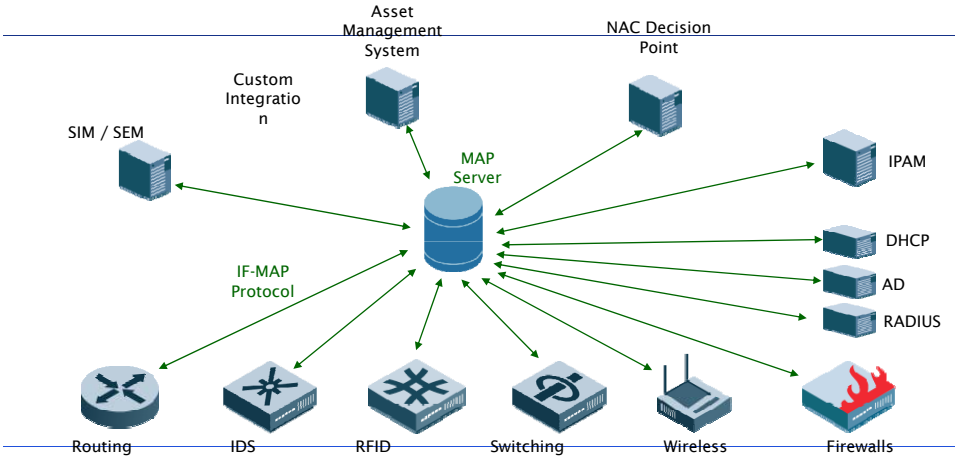

Fig. 5. An IF MAP based IT infrastructur (source: http://www.trustedcomputinggroup.org)

evidence record archival is the last step in the creation process. During the time of a evidence record in the archive the used cryptographic means can wear out resulting in a decreased level of trust in a specific evidence record. Existing work (e.g. [7]) shows approaches to maintain the probative force of digital evidence in long term archives. There are also products in the market supporting long-term archiving and re-signing archived data records.

However, combining long-term archiving with a high-speed triplestore without losing the advantages of the triplestore seems to be very difficult. Therefore, it is probably necessary to follow a dual strategy for the forensic data-base where the long-term archiving and the triplestore are not fully integrated. All evidence records will go into the triple-store, but probably only a small subset really requires secure long-term storage. During the creation of evidence records the record has to be marked in a way that the forensic data-base can decide whether long-term archiving is necessary or not. Then, a digital chain of evidence can be created using the triplestore and after completing this step long-term secured representations of the evidence records are retrieved from the long-term archive in order to produce the complete secure digital chain of evidence.

4.4. Exploring the forensic data-base

This final part of the architecture for secure digital chains of evidence strongly depends on the format and data model of the forensic data-base. If the data-base is implemented as a triplestore with a good meta model for evidence records a variety of tools and languages exist to develop interactive tools to explore the data-base. Relations between evidence records (and thus between events) can be graphically visualised, SPARQL graph queries can be used to find matching evidence records or Prolog can be used to search for evidence records belonging to a particular chain of evidence.

5. Conclusions

The architecture presented in this paper is a first approach towards building IT systems with the inherent ability to provide secure digital chains of evidence. The main goal of this work is to identify the requirements for such a system, develop a possible high-level architecture and then explore existing technology with regards to the possible use within such a framework for secure evidence.

The collection of different applicable technology shows that most parts of the architecture can be realised using existing building blocks. Nevertheless, implementing the high-level architecture for complex systems is a challenging task. First, the identification and description of events and the implementation of adequate evidence generation is not always as straightforward as in the case of digital cameras. Second, relating events e.g. by adding semantic information is a task that is not easy to introduce in all types of infrastructures.

Service-oriented infrastructures are probably more suitable than embedded systems with real-time requirements. In a service-oriented system there can be a so-called enterprise service bus ESB as the central communication and control element. Further, there are clear interfaces between applications (services) and the rest of the system. The enterprise service bus and the interfaces are the right places for event collection and also for adding semantic information e.g. from a workflow engine.

Obviously, not all events occurring in a complete IT infrastructure can be collected. Events exist on different levels from network to application or one could even consider physical events via sensors. Relevant events need to be carefully chosen with respect to their importance in processes in the system. Even in large systems with many users there might be a small subset of critical processes where the additional overhead of provident collection of evidence records can be considered

useful.

On the long run, IT experts and law experts should co-operate in setting the standards of what should be accepted as valid digital evidence. For some processes future regulations might include the collection of secure evidence records or of secure digital chains of evidence.

References

[1] P. Gladyshev and A. Patel. Formalising event time bounding in digital investigations. *International Journal of Digital Evidence*, 2005.

[2] D. Grawrock. *Dynamics of a Trusted Platform: A Building Block Approach*. Intel Press, 1st edition, 2009.

[3] T. C. Group. "tcg trusted network connect – tnc if-map binding for soap version 2.0". www.trustedcomputing.org, 2010.

[4] C. Hosmer. Digital evidence bag. *Communications of the ACM*, 49(2):69–70, 2006.

[5] F. Izquierdo, M. Ciurana, F. Barcelo, J. Paradells, and E. Zola. Performance evaluation of a toa-based trilateration method to locate terminals in wlan. In *1st International Symposium on Wireless Pervasive Computing*, 2006.

[6] O. Kerr. Digital Evidence and the New Criminal Procedure. *Columbia Law Review*, 105(1):279–318, 2005.

[7] T. Kunz, S. Okunick, and U. Pordesch. Data Structure for Security Suitabilities of Cryptographic Algorithms (DSSC)-Long-term Archive And Notary Services (LTANS). Technical report, IETF Internet-Draft, 2008.

[8] J. Liu, F. Yu, L. C.-H., and H. Tang. Optimal combined intrusion detection and biometric-based continuous authentication in high security mobile ad hoc networks. *IEEE Transactions on Wireless Communications*, 8(2), 2009.

[9] U. Maurer. New approaches to digital evidence. *Proceedings of the IEEE*, 92(6):933–947, 2004.

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

[11] M. Pollitt. Report on digital evidence. In *13th INTERPOL Forensic Science Symposium*. Citeseer, 2001.

[12] M. Reith, C. Carr, and G. Gunsch. An examination of digital forensic models. *International Journal of Digital Evidence*, 1(3):1–12, 2002.

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

[14] C. Rudolph, Z. Velikova, and N. Kuntze. Secure web service workflow execution. *Electronic Notes in Theoretical Computer Science*, 236:33–46, 2009.

[15] G. Starnberger, L. Froihofer, and K. Goeschka. Using smart cards for tamper-proof timestamps on untrusted clients. In *International Conference on Availability, Reliability and Security, ARES*, pages 96–103. IEEE Computer Society, 2010.

[16] F. Stumpf, A. Fuchs, S. Katzenbeisser, and C. Eckert. Improving the scalability of platform attestation. In *Proceedings of the Third ACM Workshop on Scalable Trusted Computing (ACM STC'8)*, pages 1–10, Fairfax, USA, October 31 2008. ACM Press.

[17] M. Wing, A. Eklund, and L. Kellogg. "consumer-grade global positioning system (gps) accuracy and reliability". *Journal of Forestry*, 103, 2005.

[18] T. Winkler and B. Rinner. TrustCAM: Security and Privacy-Protection for an Embedded Smart Camera based on Trusted Computing. In *Proceedings of the Conference on Advanced Video and Signal-Based Surveillance*, 2010.