eSeal - A System for Enhanced Electronic Assertion of Authenticity and Integrity

Christian Decker[1], Michael Beigl[1], Albert Krohn[1], Philip Robinson[1], Uwe Kubach[2]

[1] Telecooperation Office (TecO), University of Karlsruhe
Vincenz-Priessnitz-Strasse 1, 76131 Karlsruhe, Germany
{cdecker, michael, krohn, philip}@teco.edu
[2] SAP AG, Corporate Research
Vincenz-Priessnitz-Strasse 1, 76131 Karlsruhe, Germany
uwe.kubach@sap.com

Abstract. Ensuring authenticity and integrity are important tasks when dealing with goods. While in the past seal wax was used to ensure the integrity, electronic devices are now able to take over this functionality and provide better, more fine grained, more automated and more secure supervision. This paper presents eSeal, a system with a computational device at its core that can be attached to a good, services in the network and a communication protocol. The system is able to control various kinds of integrity settings and to notify authenticated instances about consequent violations of integrity. The system works without infrastructure so that goods can be supervised that are only accessible in certain locations. The paper motivates the eSeal system and its design decisions, lists several types of integrity scenarios, presents the communication protocol and identifies practical conditions for design and implementation. An implementation in a business relevant scenario is presented as a proof of concept.

1. Introduction

It is an important issue to claim and assert the authenticity and integrity of goods, documents or other valued objects in storage or transit. In these times objects of value like documents, deeds, contracts, goods for trade, and other articles, which we collectively refer to as goods, were stored in a container, which in turn was sealed with wax and the imprint of a seal ring (bearing an insignia) or a plumb. This method ensured two important fundamentals of secure and dependable object handling: Authenticity and integrity. The object's authenticity is detectable through the seal ring imprint on the wax and the integrity can be discerned by inspecting for either of the two physical states of the seal - *valid* or *broken*. Nevertheless, modern technology provides advanced methods for violating both integrity and authenticity but also enables us to better protect objects of value.

This paper introduces an electronic seal concept, the eSeal. Like a wax-seal, an eSeal can be applied to physical goods to electronically claim and assert their authenticity and integrity. The eSeal is intended to claim and assert states but not to

protect the object itself. However, unlike a wax-seal, an eSeal can detect a larger variety of integrity violations – including electronically originated attempts. It can collect context information about this violation including the time and location, can actively monitor and alert and it can perform all these tasks automatically and autonomously. An eSeal can exchange relevant information with other computer systems, maintaining a fine-grained correlation of physical conditions and an interpretation in the information world.

Although the design and concept are generally applicable in many areas, this paper motivates and explains the eSeal concept alongside business applications. In the area of integrity and authenticity supervision, business applications provide an interesting environment with numerous demands for eSeal-related applications. Dwelling in the business application domain also motivates interaction and hence extension of existing information systems through appropriate interfaces to eSeal components.

The paper proceeds to give an analysis about various integrity classes the eSeal can keep track of and explain the eSeal practical considerations, which include particular requirements and constraints of the overall system. This leads into the system design, where the components and their dependencies and tasks are explained and the operational features are outlined. Due to the system constraints and operational features, there were particular security and technical challenges that necessitated further analysis, before practically evaluating the concept through a concrete application. We also discuss related work towards the end of this document.

2. Motivation and Analysis

As a motivating example, see Figure 1, we select a representative logistic scenario to clarify the capability and advantage of an eSeal.

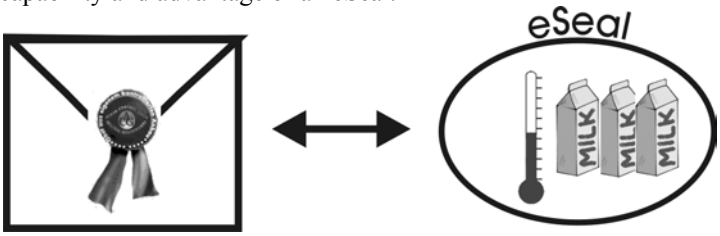

Fig. 1. A traditional seal compared to an eSeal

In this example, a temperature sensitive good like milk is transported and need to be kept in a certain range of temperatures for goods' quality reasons. An eSeal is used to assure the temperature of the goods during transport between two locations. During the transport the eSeal permanently monitors the current temperature of the goods. As long as the temperature is within the acceptable range, the eSeal is considered to be valid, otherwise broken. Once broken it can never be recovered to the valid state like a broken wax seal. When the transported goods arrive at their final destination, the eSeal can report authentically whether the temperature range was held.

Like the traditional wax seal on envelopes the eSeal can protect valuable goods. The simple protection with a wax seal can be matured with the surveillance of

additional conditions. The eSeal can provide protection for goods sensitive to for instance temperature and light changes, vibration and radiation.

To go a bit deeper into the eSeal system, we analyzed two aspects of its general problem domain, presented as research questions. First, what are potential breach-of-integrity/authenticity situations and how are they classified? Second, what are practical constraints for an eSeal system design and implementation?

2.1 Integrity Considerations and Classifications

The concept of integrity we want to target with the eSeal system is more than "inviolability", as guaranteed by wax seals. Depending on the object and context, integrity may still be in place even if the object is touched.
The eSeal domain spans over four different integrity classes, which we derived from an analysis of four scenarios in the business areas: storehouse, supply-chain management, office document management and production. These classes are:

- **Conditional Integrity**. This is upheld when the object's physical properties and object state remain unaltered or undamaged. In this case a full access to a sealed object may be allowed, in that the object may be used, but it is forbidden to change the state of the internal – e.g. information – or external – e.g. physical shape – of the object
- **Relational Integrity**. This is similar to the above, but considers the orientation and relation of constituent objects. Integrity is violated when someone adds or removes something from a sealed object collective. Objects may consist of several constituent objects like a palette of goods consist of several goods.
- **Authorization Integrity**. This is the classical wax seal integrity where no unauthorized party is allowed visual or tangible access to the sealed object. Integrity is broken if someone was able to see a defined state e.g. internal information but also the outline of the object. Beyond the "open the container and look" integrity violation, modern forms of spying include x-ray scans and methods to get access to internal information – stored programs and data – of an object.
- **Environmental Integrity**. In this case the object's integrity is violated if its surrounding conditions or context are unfavorable, e.g. that the object is brought into a place where it should not be.

These classes of integrity concerns must at times be addressed in tandem. For example, a policy could exist that includes access restrictions (Authorization Integrity) and yet that the object's structural properties must not be changed (Intrinsic Conditional Integrity).

Our method of defining these integrity classifications includes only a limited number of scenarios. Based on scenario descriptions we repeated the analysis until we found the same integrity protection situation again. This way we observed important situations that contributed to a design of a first eSeal system, but cannot ensure completeness. Further on, the list is based solely on business scenarios analysis,

whereas other policies may be found when analyzing other areas of life. As potential exploitation scenarios are within the business area, we do not consider this a significant system drawback. Subsequently, the remainder of this section continues within the business area.

2.2 Practical Considerations

The practical considerations and important requirements for the eSeal system design were derived from the inherent goals and properties of the business scenarios that were analyzed for the potential usage of eSeals. They essentially describe and confine the nature of the goods handled, the locations that they are transited between, and the interaction with humans.

- **Mobility.** The eSeal system should not introduce any handling constrains of objects, work without cabling, be small and unobtrusive.
- **Diversity.** Goods have different physical properties like size, shape, weight and experience different environmental conditions. Different values are in the interest to be sealed (e.g. time of transport, temperature). Therefore the eSeal system must provide a flexible platform to realize an electronic seal on a certain good.
- **Incomplete infrastructure coverage.** Physical goods can move through various situations and different locations or environments. Since the support through an electronic infrastructure (e.g. W-LAN, cameras) cannot be guaranteed in all cases, the eSeal system must be able to work offline and autonomous. It must have intensive contact to the object and experience its environment as genuine as possible through monitoring equipment and sensors.

3. eSeal System

Definition: *An eSeal is an electronic seal, which can be applied on physical goods in order to provide the guarantee of important aspects of the protection of those physical goods. The eSeal does not physically protect the sealed goods but can provide propositions and evidence of authenticity and integrity.*

The eSeal system, see Figure 2, consists of three conceptual layers: (1) the *Contractual*, (2) the *Logical*, and (3) the *Technical*.

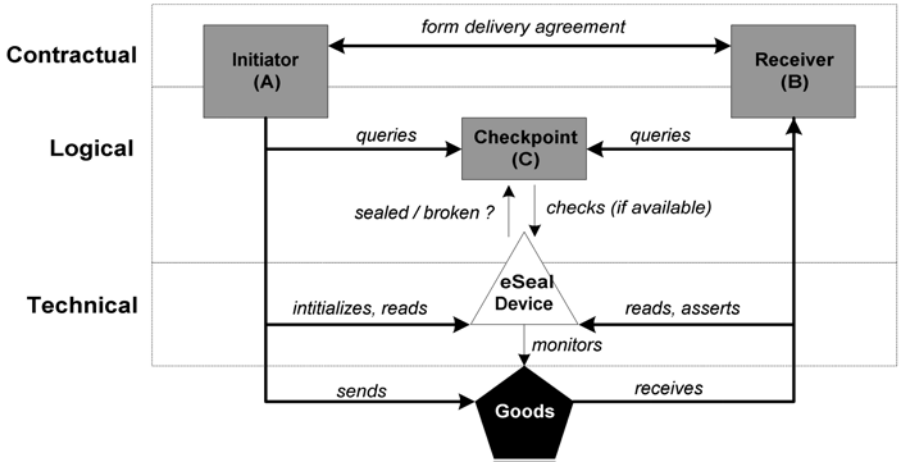

Fig. 2. eSeal System and Key Actors

Firstly, we regard an eSeal as a contract between an *Initiator* (a subject that applies the eSeal to a physical good) and a *Receiver* (a subject that assesses the evidence presented by the eSeal) stating the terms and conditions under which the authenticity and integrity of a physical good can be asserted. Secondly, the system provides logic for determining and presenting the "protection state" of the target goods to which it applies. Thirdly, the system is realized through particular technologies that meet the functional and quality requirements for its operational domain. In addition, we have also considered the actors that drive or benefit from the system's functionality. We have already mentioned the roles of the Initiator and Receiver, who are considered as the end-points of the activity chain and the key actors in the contractual aspects of the system. Supporting the Initiator and Receiver in monitoring the state of the eSeal system, and hence the contract, are *Checkpoints*. A Checkpoint is an intermediary actor that forwards system state to either of the contractual parties upon their query. A Checkpoint is considered the most proximate trusted source to the sealed goods at a particular time. Operational system state of the eSeal is either "valid" or "broken" and this is determined by the processing of delivery conditions, with which the system is initialized, and current conditions. A multi-sensory device, attached to the physical goods, which we simply refer to as the "*eSeal device*", senses these current conditions.

3.1 eSeal Device

The eSeal device is a small embedded computer system directly attached to physical goods. The intention is to have an entity which can provide the Initiator or Receiver with a trustworthy statement about whether the operational system state also referred to as eSeal state is "valid" or "broken". An eSeal device implements three core functionalities: First, computation enables permanent updates of the eSeal's state using an algorithm derived from the contract between Initiator and Receiver. Second, a sensor system as part of the device supplies it with external information serving as

input for the algorithm. Third, the device implements a communication functionality enabling the exchange of the eSeal's state and optionally additionally information with other parties in an authenticated manner. The eSeal device operates independently and without the support of a surrounded electronic infrastructure. Optionally, the eSeal device contains a timer to bind the eSeal's state to a timestamp and some storage capability to consecutively write the eSeal's state history and additional related information.

3.2 eSeal Activity Chain

The flow of the use of an eSeal, presented in Figure 3, involves at least three parts of the eSeal system: The Initiator, the Receiver, the eSeal device and optional Checkpoints.

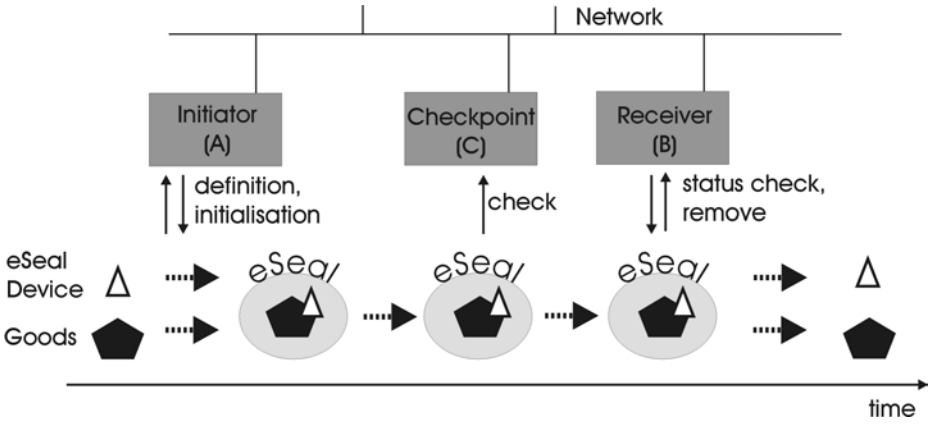

Fig. 3. eSeal Activity Chain

After the successful agreement and conclusion of a contract between Receiver and Initiator, the Receiver queries the Initiator to start the activity chain. The definition of the seal to be applied on the goods is derived from this contract.

The Initiator combines the eSeal device and the goods and initializes the eSeal device order to seal the goods which are subject to the contract and from this moment on, the eSeal device can be identified by the Receiver, Initiator and Checkpoints and carries the start state "valid". The initialization of the eSeal device includes the definition and download of all necessary information and algorithms that define the seal. The eSeal device can then continuously monitor the state of the seal.

If the Receiver or Initiator is interested in the actual state of the sealed goods, i.e. whether the seal is still valid or broken, they can query the state from Checkpoints or directly from the eSeal device using its communication interface. This enables both parties to prove whether there was an integrity breach or not.

The activity chain terminates when the sealed goods reach the Receiver. The Receiver and the Initiator query the state of the eSeal device for the states "broken" or "valid" and can decide whether the contract was held. If the eSeal device arrives

"broken" at the Receiver, the contract partners can query the eSeal device for the reason of the breach of the seal. They can distinguish between breaches due to a contract breach or cases of attacks. In cases of severe attacks on the eSeal device which destroyed core functionalities of the eSeal device, this information might be lost. After the queries and contract examination the Receiver removes the eSeal device from the goods and thus deactivates the seal.

4. Operational Analysis and Challenges

When we discussed the eSeal approach in the second section of the paper, we mentioned two particular aspects of the problem domain that influenced the properties of the system architecture. These naturally have a significant bearing on the operational specification of the system, which is likewise separated. We therefore dedicated some resources to analyzing the security requirements and deriving a general functional protocol, and, secondly, analyzing the technical realization of the system based on the practical constraints. This also took into account the security requirements that emerged from the analysis, with respect to storage and processing.

4.1 Security Analysis

The Security Analysis considers the eSeal system actors, the nature of the goods to which it will apply, and the types of transactions and business scenarios that the system will be involved in. The eSeal protection goals of Integrity and Authenticity are once again revisited, but from a more detailed security perspective. There is large commonality with the concerns of authenticity and integrity in cryptographic analysis.

Authenticity: The receiver (B) must assert that a good or item (I) was really sent from a sender (A,) and is hence a genuine article, including that the electronic information also conforms to these properties. Threats include:

- A false initiator sends I by bearing A's identity (source masquerading)
- I or its electronic information is replaced in transit by a falsified item or data (replay attack)
- A false seal sends out item state information to A and B (seal masquerading)

Integrity: both the receiver (B) and initiator (A) must assert that item (I) (as well as its electronic information) is not tampered with while in transit, and that the correct handling policies are upheld. Threats include:

- I is tampered with (seal is broken) while left unattended, or by an authorised third party, therefore degrading quality of the product
- I is subjected to transit conditions that violate its handling policies

Other threats include the inevitable denial-of-service attacks through communications signal interference or continuous, unwarranted depletion of power resources. Additionally, in the case of highly sensitive information on the seal, confidentiality becomes another protection goal of eSeal. The communications

protocol and power management features of the device address the denial-of-service attacks, while confidentiality is captured within the properties of the crypto protocols and physical handling policies enforced. These threats and their countermeasures, especially the asymmetric or public key protocol we use as our foundation, are well known in the field of security engineering [1]. However, it was a good opportunity to explore and assess the applicability of these standards within a domain where the physical and electronic protection goals are so tightly coupled.

4.2 The Detailed eSeal Communication Protocol

The protocol defined is based on the architecture depicted in fig. 1. It was specified in response to the security analysis, and details how the protection goals of the interaction between entity roles are captured. The protocol consists of 7 interaction phases, corresponding to the architecture depicted in fig. 1, but also of a set of security functions and elements defined below.

Security Functions and Elements

K_X:	public key of an entity x
M:	Query and status messages
n, q:	Initial random sequence number, and sequence counter
P:	Handling policy
$D_X\{\}$:	Decryption with private key of entity x
$E_x\{\}$:	Encryption with secret/ private key of an entity x
H{}:	Hash function
$S_X\{\}$:	Signing with private key of an entity x
$V_X\{\}$:	Verification of signature with public key of an entity x

1. **QUERY-ORDER**: receiver (B) sends an order request message (M_n), with which the initiator (A) can initialize an eSeal session. To avoid replay attacks at this stage, a signed hash of M_n, a random number n (used as a sequence number), and ipublic key (K_B) of the receiver (B) are also sent to the initiator (A). These are also encrypted with the public key of the initiator (A) - (i). Upon reception, the initiator (A) decrypts the packet using its private key – (ii), and then verifies the sender of the order, using the public key of B – (iii).

B → A:	E_A {M_n, S_B{H{M_n}}, n, K_B}	(i)
A:	D_A {E_A {M_n, S_B {H{M_n}}, n, K_B}	(ii)
	V_B {S_B{H{M_n}}}	(iii)

2. **INIT-DEFINE**: initiator (A) starts the initialization process by defining a handling policy (P, which is a listing of context parameters), a statement of expected state-on-delivery (M_{n+1}), and by generating a key pair for the seal. The handling policy is encrypted with the private key of the seal to avoid electronic tampering. The seal is then electronically initialized with its private key (in protected memory), the handling policy, the public key of B (for communicating

status updates to B with end-to-end authentication), and the expected state-on-delivery, which is hashed and signed by the private key of A.

$$A \rightarrow Z: \{P, n, S_A\{H\{M_{n+1}\}\}, K_B, M_{n+1}\} \quad \text{(iv)}$$

A then responds to B by sending a STATUS (see protocol operation 7), which includes sending the public key of the seal to B.

3. **SEAL**: Upon applying the seal to the item, this triggers the sensors to make the first check (see 6) in order to have an initial-sealed-state (M_{n+2}). The physical process of sealing also triggers an electronic process of encryption and signing of the initializing information and initial-sealed state respectively – (v). The seal can only be opened by parties that can respond to a challenge by the eSeal device, such as the initiator (A) and receiver (B), as their public keys are known by the eSeal.

$$Z\ [I]: E_Z\{P, n, S_A\{H\{M_{n+1}\}\}, E_B\{M_{n+1}\}, S_Z\{H\{M_{n+2}\}\}, M_{n+2} \quad \text{(v)}$$

4. **QUERY**: This step in the protocol is equivalent to an ORDER. The only difference is that B may directly contact the seal, having received its public key, or it may need to go via a checkpoint E_A would therefore be replaced with E_Z , an operation on the eSeal itself, in `(i)`, `(ii)` and `(iii)`.

5. **CHECK**: Following an authorized party QUERY or internally scheduled query, the eSeal (Z) does a poll of its sensors and compares with the preferred context parameters specified in the handling policy (P). It then updates the last status (M_n) with current status (M_{n+q}), where q is equal to the sequence number of the query. There are three context states that the seal can be set to, and stated in M_{n+q}:

   - *VALID*: Current context match handling policy - seal remains intact
   - *DEGRADED*: Current context does not fully meet policy, but is within an acceptable bound – seal remains intact but records possible tampering attempt. For example, is currently in the hands of an unauthorized party.
   - *BROKEN*: Current context does not meet handling policy – seal is broken and relevant information is wiped from electronic storage

   The seal can also record the current handling party and labels them as *AUTHORIZED* or *UNAUTHORIZED* (unknown or black-marked). A higher-level notification is given off when the sealed item is being handled by an *UNAUTHORIZED* party.

6. **STATUS-RESPONSE**: There are two types of STATUS operations, which both transmit the CHECK to an authorized party. The first is a response to the authorized parties following a query. It is authenticated with a signature of the seal (S_Z). Additionally, depending on the policy, the status may be encrypted with

the public key of the authorized party before forwarding. This is equivalent to forwarding the result of the crypto procedure in (v), where q = 2.

Z → B:	$E_B\{S_Z\{H\{M_{n+q}\}\}, M_{n+q}\}$	(vi)
B:	$V_C\{E_B\{S_Z\{H\{M_{n+q}\}\}, M_{n+q}\}\}$	(vii)
	$D_B\{S_Z\{H\{M_{n+q}\}\}$	(viii)
	$V_Z\{H\{M_{n+q}\}\}$	(ix)

7. **STATUS-DELIVER**: The second STATUS operation is when the item is physically delivered. The current handling party is set to AUTHORIZED, if B provides its public key K_B, i.e. responds to the eSeal's challenge. Furthermore, without K_B, procedure (viii) is not possible. If (viii) is not possible, then a notification is issued by the eSeal.

Important to note that in implementations where the microprocessor cannot support public key encryption, the eSeal challenge will have to be based on a symmetric approach. This would entail an earlier exchange of the eSeal secret key with the receiver and initiator, over a covert channel.

4.3 Technical Analysis and Realization

Reflecting back at the system architecture (Figure 2) the eSeal device is a central element of the eSeal system since this device is responsible for detecting integrity violation of the sealed goods. This section describes the technical details of the eSeal device and outlines requirements to prevent successful attacks which compromise the device. From section 3.1 the following functionalities are necessary in the eSeal device: computation, communication and sensing. Additionally, the device is supported by a power supply. The functionalities are implemented in different subsystems requiring separate appropriate protection against attacks. The figure below presents an overview about the components of an eSeal device.

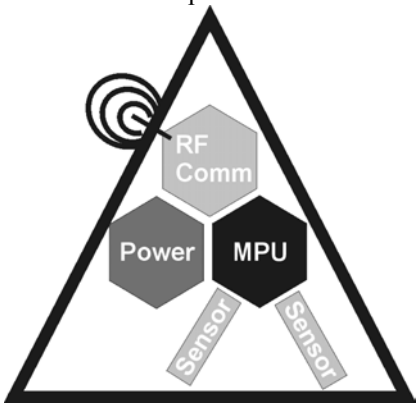

Fig. 4. Schema of an eSeal Device

All necessary computation functionality is implemented in the microprocessor unit (MPU). The MPU covers the tasks of cryptography (including de- and encryption, key management), permanent eSeal state determination and sensor value processing. Therefore the MPU contains the most sensitive data and present the most valuable attacking target. It is required that no invasive or non-invasive method will gain knowledge about the MPU's internal states. This complicated protection task is discussed in [9]. The authors describe there several ways to read out protected data from MPUs on SmartCards, but also effective countermeasures to those attacks Another requirement for the MPU is to hold a state within the MPU, which cannot be reproduced by any method once lost. This internal MPU state includes the eSeal device state and the integrity of the MPU itself. This state is wiped out of the MPU as soon as a seal breach or an attack is detected and will ensure that the seal cannot be reestablished. To our knowledge, there are currently two preferred MPUs on the market which fulfill these requirements. The first one is the DS5002FP [15] series from Dallas Semiconductor and the second one is the IBM 4758 architecture[14]. Both support countermeasures described in [9] to prevent non-invasive attacks. Additionally, both provide a protection of the MPU against invasive attacks using a physical shielding, e.g. a membrane to detect intrusion in order to avoid invasive attacks without notifications. However, publications like [2] and [5] point out that apart from hardware protection, the software layers in such processors need also be considered carefully. Otherwise, protocol attacks can make the hardware protection useless.

The communication hardware itself does not add further security vulnerability. The security tasks are part of the higher level protocols. A destruction of the communication or denial-of-service attack would prevent the receiver from reading the eSeal state. The receiver would consider the eSeal to be absent.

The sensors support the MPU to permanently update the eSeal state. Those sensors reside outside the tamper-proof shield of the MPU. The selection of the appropriate sensors generally depends on the target application. Sensors have to be selected from the requirements of the goods to be sealed and the required integrity situations. Outside the tamper-proof area of the MPU, sensors face attacks including manipulation of sensor values during the transport to the MPU or sensor cheating. In the latter case, the attacker tries to maintain the valid sensor conditions during his attack through creating the right environment in which the sensor is situated in. In order to accomplish attacks to the data transport from the sensor to the MPU, the transport has to be protected by either a physical protection such as shielding of cables and the sensor itself or by the use of crypto protocols for the data transmission. Latter will transform the sensor into another MPU based crypto system. One possibility how this can be realized is described in [11] for secure keyboard input in Next Generation Secure Computing Base (NGSCB) enabled computer. In order to approach the threat of sensor cheating, the MPU can regularly check the sensors' health state. This requires the sensor to record its operation conditions using further internal sensors of itself. This sensor-watches-sensor scenario can be replaced by a seal-watches-seal scenario, where an eSeal device can be supported by neighbored devices in order to verify its own reading. The physical arrangement of the goods to be sealed together with the eSeal device can also mechanically protect the sensors

from an attacker. These considerations have to be made before initializing the eSeal since they depend heavily on the type of goods to seal and the expected attacks.

The eSeal device needs power source supplying all its components. In mobile scenarios, battery supply is appropriate to allow independent operation. Currently, the battery life-time determines the limits of the usage of an eSeal device since a power failure leads to the lost of the state of the MPU and therefore leads to the state "broken" in the eSeal device.

5. Applications and Implementations

We have recently started to implement the eSeal concept in various applications especially with the focus on the scenarios which were the basis of the integrity classification in section 2.1. One application for physical document integrity is considered in more detail.

5.1 General Implementation Details

Implementations of the eSeal devices are based on TecO's Smart-It Particle platform [3] providing the necessary functionality like sensing, computing and wireless communication of an eSeal device. Our eSeal prototype implementation adds more functionality where needed using the Particles hardware and software interfaces. The roles of initiator and receiver were taken over by regular personal computers. Connection between the eSeal devices and the Internet enabled personal computers are carried out via so-called XBridge devices which form a gateway between the wireless eSeal network and the Internet. Such Xbridge devices are installed at the site of initiators and receivers but also in certain checkpoints.

We developed a library and some hardware extensions for the Particle platform, namely to include special sensors needed for the eSeal applications. New software components focused on secure communication using the blowfish algorithm. Although the eSeal system design requires an asymmetric key algorithm, this will be available in the future implementation. The additional hardware we developed are capacitive sensors to supervise the integrity of compounds of goods. Using these kind of sensors we also investigated possibilities to detect invasive physical attacks to the eSeal device. Experiments are hereby still at the very beginning. Furthermore, eSeal devices have access to already built-in Particle functionality like Real-Time-Clock and the Cell-of-Origin location system[4].

5.2 A first eSeal Application for Document Integrity

In office environments documents are usually created in an electronic way. Nevertheless, for convenience or legal reasons they are also printed on paper. The DigiClip [6] is a digitally enhanced paper clip, which aims to bridge the gap between electronically created documents and their physical paper-based representation. It was developed to keep the state of an electronic document and its printed version

consistent. Once clipped on a printed document (Figure 5) it is able to keep track of document locations on a room level granularity and to monitor various environmental and document specific contexts. Currently, it can detect contexts like “document put in a bag”, or “page from/to document removed/inserted”.

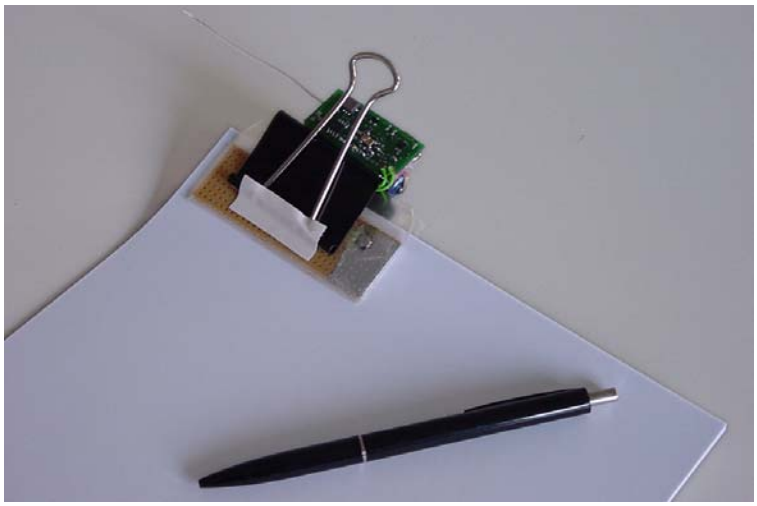

Fig. 5. DigiClip clipped on some Papers

For the eSeal-based application the DigiClip device monitors the conditional integrity and environmental integrity of paper based document. We selected these two integrity situations because they represent two crucial document characteristics: the togetherness of all pages in a document and valid locations of a document. The device’s capacitive sensor is able to detect the number of pages currently clipped and whether the clip is opened or not. Like the electronic file of the document keeps all pages within the document structure it is therewith possible to decide on the physical document whether all pages are still together or a page left the compound. The DigiClip’s cell-of-origin location system enables the definition of areas where the printed document is allowed to stay. Like restrictions applied on the electronic document denying for instance move operations it is possible to apply such restrictions to the physical documents by limiting the handling to certain areas.

Our scenario for using the DigiClip as an eSeal application was as follows: After an electronic document was printed it had to be transported from the initiator to receiver represented by personal computers in two different rooms. In between there were two checkpoints the clip had to pass and one other it was not allowed to pass. The initiator configured the DigiClip device to monitor the opening of the clip, the page count and the DigiClip device’s locations along the path to the receiver. Therewith, the eSeal was established around the physical document. Its state was held in the memory of the Particle’s micro controller. As long as the clip was not opened, the number of pages didn’t change and the DigiClip device was on its way indicated by the checkpoints it has to pass, the structural and environmental integrity of the document was assured, i.e. the eSeal’s state was valid. During the operation the

device constantly monitored these integrity conditions. When the integrity was violated, meaning that the clip was opened or it was seen by the third checkpoint, the eSeal state was set to "broken" and reported back to the personal computers representing initiator and the receiver via Xbridge gateways in the checkpoints. When the clipped document reached its final destination the receiver personal computer queried the DigiClip device and could conclude the eSeal's state "valid" or "broken". All communication was encrypted using a symmetric blowfish algorithm because the micro controller on the Smart-Its particles is not powerful enough to practically implement advanced asymmetric algorithms like RSA. The shared secret therefore had to be exchanged over a covert channel, and out-of-band with respect to the device's communications. Using this first implementation we were able to detect both structural and environmental integrity breaches.

6. Related Work

There is other work which is related to our approach of an eSeal. Siegemund and Flörkemeier describe in [13] a scenario of smart product monitoring. Hereby, products are augmented with sensors to monitor exceptions like dropping of the product. This is then communicated to any mobile phone nearby without explicit pre-configuration. While the eSeal shares the use of sensors for detecting exceptions, it goes beyond this monitoring aspect. The eSeal state "valid" or "broken" is determined from conditions during initialization and current conditions. Sensor measurements are used to derive these current conditions. Further, the eSeal system design guarantees that only authenticated parties are able to query the eSeal's state and further that manipulations on the eSeal device are recognized.

The proliferation of electronic business processes has fostered the need to integrate physical goods into the electronic world. Especially in applications like supply chain management where goods are distributed among many different players, which might be spread around the world, it has become very important to electronically track such goods and to electronically assure their integrity and authenticity. As a consequence first electronic solutions like MacSema's ButtonMemory[10], Elogicity's eSeal [7], Hi-G-Tek's Active Hi-G-Seal[8] or Savi Technology's SmartSeal [12], which claim to seal physical goods, are available on the market. These solutions are based on various technologies like electrical contact in case of the ButtonMemory, RFID in case of Elogicity's eSeal, and GPS support in case of the Active Hi-G-Seal and the SmartSeal. They mainly provide some tracking feature that makes it possible to monitor if your goods arrive at pre-defined checkpoints. Additionally, one can conclude whether someone access the device or the goods sealed by these solutions. Other integrity surveillance based on environmental conditions for instance is not achieved. Furthermore, except for the Active Hi-G-Seal, which uses 3DES, no other seal offers a secure communication. Our eSeal approach covers a wider scope towards other integrity conditions as well as a secure and authenticated communication to whom it is allowed to query the eSeal's state.

IBM's secure coprocessor, the IBM 4758, is guaranteed to work in a secure manner despite physical attacks [6]. In contrast to standard cryptographic accelerator

chips this coprocessor puts cryptographic secrets and a tamper detecting and responding circuitry in a secure box. Any detected tamper event immediately results in loss of the cryptographic secrets. Hence this coprocessor unit can be considered as a sealed object, for which the integrity of condition is guaranteed. The scope of the seal is limited to the detection of intrusions into the secure box surrounding the coprocessor unit. Nevertheless the IBM 4758 can well serve as a hardware platform to built upon, for some forms of specialized eSeals as we introduced them in this paper.

7. Conclusion and Future Work

The background, approach, design, operational analysis and an applied example of the eSeal system have been presented in this paper. It has been shown that electronic counterparts may uphold the function of inert seals, in everyday applications. Furthermore, this primal functionality is extended by incorporating sensors, communications and micro processing, with the added capability of interaction with other information systems.

We foresee both economic and social impact if such an architecture were to be taken up by industry, and we are actively investigating such "take-up", by forming research and development projects and coalitions with industrial partners. Nevertheless, there is further work to do, as the extremities of reference implementations of this architecture have not been explored. There may be other application areas besides business and commerce. Sensors, microchips and communications capabilities will continue to evolve. Continuing experience commensurate with these developments will be disseminated throughout the research community.

References

1. Anderson, A. Security Engineering: A Guide to Building Dependable Distributed Systems. Published by John Wiley & Sons, 2001, ISBN 0-471-38922-6
2. Anderson, R., Kuhn, M.: Tamper Resistance - a Cautionary Note. The Second USENIX Workshop on Electronic Commerce Proceedings, November 18-21, 1996. Oakland, California. pp 1-11, ISBN 1-880446-83-9
3. Beigl, M., Zimmer, T., Krohn, A., Decker, C., Robinson, P.: Smart-Its - Communication and Sensing Technology for UbiComp Environments. Technical Report ISSN 1432-7864 2003/2
4. Beigl, M., Zimmer, T., Decker, C.: A Location Model for Communicating and Processing of Context. Personal and Ubiquitous Computing Vol. 6 Issue 5-6, pp. 341-357, ISSN 1617-4909, 2002
5. Bond, M.: Attacks on Cryptoprocessor Transaction Sets, Workshop on Cryptographic Hardware and Embedded Systems (CHES2001), 31st January 2001, Paris.
6. Decker, C., Beigl, M., Eames, A., Kubach, U. DigiClip: Applying electronic properties to physical documents. To appear in the Proceedings of the IWSAWC 2004, March 23rd 2004, Tokyo.
7. elogicity.com global track and trace solutions to all parties within the supply chain management process Available Online: http://www.elogicity.com/solutions.htm [Accessed: 07/11/2003]

8. Hi-G-Tek: Secured Cargo. Available Online: http://www.higtek.com/cargo2.htm [Accessed: 08/02/2004]
9. Koemmerling, O., Kuhn, M.: Design Principles for Tamper-Resistant Smartcard Processors. Proceedings of the USENIX Workshop on SmartCard Technology, 10-11 May 1999, Chicago, USA.
10. MacSema Inc.: MemoryButton Technology. Available Online: http://www.macsema.com/solutions.htm [Accessed: 08/02/2004]
11. Microsoft: Hardware Platform for the Next-Generation Secure Computing Base. Available Online: http://www.microsoft.com/resources/ngscb/documents/NGSCBhardware. doc [Accessed: 08/02/2004]
12. Savi Technology: Securing the Smart Supply Chain. Available Online: http://www.savi.com [Accessed: 07/11/2003]
13. Siegemund, F., Flörkemeier, C.: Interaction in Pervasive Computing Settings using Bluetooth-enabled Active Tags and Passive RFID Technology together with Mobile Phones. In Proceedings of IEEE PerCom 2003 (IEEE International Conference on Pervasive Computing and Communications), March 2003, Fort Worth, USA.
14. Smith, S.W., Weingart, S.H.: Building a High-Performance, Programmable Secure Coprocessor. In Computer Networks, Special Issue on Computer Network Security, Vol. 31, pp. 831-860. April 1999.
15. Dallas Semiconductor: Datasheet to Secure Microprocessor DS5002FP. Available Online: http://pdfserv.maxim-ic.com/en/ds/DS5002FP.pdf [Accessed: 08/02/2004]