Security Extensions for Improving Data Security of Event Repositories in EPCglobal Networks

Matthieu-P. Schapranow, Alexander Zeier, Hasso Plattner
Hasso Plattner Institute, Enterprise Platform and Integration Concepts
August–Bebel–Str. 88, 14482 Potsdam, Germany
{matthieu.schapranow|alexander.zeier|hasso.plattner}@hpi.uni-potsdam.de

Abstract—Location-based event data is captured in RFID-aided supply chains for tracking individual goods. They are stored in distributed event repositories by involved supply chain parties. Performing anti-counterfeiting checks involves exchange of event data without exposure of sensitive business secrets. Current EPCglobal standards leave the definition of security strategies open for concrete implementation.

We consider data security as the major aspect that needs to be clarified before wide adaption of EPCglobal standards will be considered by industries. The given work contributes by defining security extensions for EPCglobal networks and sharing implementation details of our research prototype. We show that incorporating in-memory technology enables history-based access control while keeping response times fast.

I. INTRODUCTION

Radio Frequency Identification (RFID) technology is named as a possible basis for future anti-counterfeiting by providing enhancements of existing business processes [1]. Hereby, the use of unique Electronic Product Codes (EPCs) [2] for identification improves processing times during goods receipt and enables automated product tracking and tracing. The EPC is used to refer to a concrete item instance in a software system. For example, it identifies a concrete bottle of analgesic that was manufactured on Jun. 01, 2011 at 08:15 a.m. In contrast, currently used barcodes identify a class of pharmaceuticals, e.g. all analgesics of a certain manufacturer. RFID technology shows prevailing advantages in contrast to barcodes, RFID tags can be read without establishing a direct line of sight, multiple tags can be read simultaneously, and they can cope with dirty environments [3], [4].

In the following, we refer to an RFID-aided supply chain when dealing with a supply chain solution that builds on goods' tracking and tracing functionality by integrating RFID technology [5]. In context of the pharmaceutical supply chain, the integration of tracking functionality is widely considered, e.g. two-dimensional data matrix or RFID technology, since this specific industry is confronted with increasing counterfeit rates [6]. However, advantages of using RFID technology only apply when all supply chain participants integrate tracking solutions.

The transformation towards an RFID-aided supply chain in huge companies involves the introduction of various IT systems that are not required today. For example, EPC Information Service (EPCIS) repositories [7] are required for storing event data captured at various locations within the company and for exchanging event data in certain granularity with other supply chain participants, e.g. for electronic advice letters, verification of products, or anti-counterfeiting. We agree, that fast adaption of EPCglobal networks supports various business processes. However, considerations about data security of sensitive business data prevent full advantages of automatic data exchange in today’s EPCglobal networks. Fig. 1 depicts the components and the message flow of our research prototype enhancing security of EPCIS repositories.

The rest of the paper is structured as follows: Sect II outlines challenges of the pharmaceutical supply chain, which draw the motivation of the given paper. Our work is set in context of related work in Sect III. Details about security components of our research prototype and their design are presented in Sect. IV while the extended communication protocol is introduced in Sect. V. The definition and enforcement of access rights by the proposed security extensions are discussed in Sect. VI and we apply them in Sect. VII to an existing EPCIS repository. In Sect. VIII a security evaluation for specific threats in context of the given pharmaceutical scenario is provided. The work concludes with an outlook in Sect. IX.
II. PRODUCT COUNTERFEITS IN THE PHARMACEUTICAL SUPPLY CHAIN

The European pharmaceutical industry hits newspaper headlines with operation MEDI-FAKE announcing 34 million detected fake drugs in a period of two months [8]. The European Commission reports an increase of 118% of pharmaceutical counterfeits detected at borders in 2008 compared to the previous year. The pharmaceutical product category is the third largest product category in terms of quantities of intercepted articles besides the categories CDs/DVDs and cigarettes [9].

Counterfeits are a risk for customers and for suppliers, because their effects are neither tested nor validated. A high level of supply chain integrity is the basis for reliable product tracking to support anti-counterfeiting.

The European pharmaceutical supply chain consists of approx. 2,000 manufacturers, 50,000 wholesalers, and 140,000 retailers [10]. In an RFID-aided supply chain, each participant stores events $e^*$ with the Electronic Product Code (EPC) [2] of a certain item in an EPC Information Service (EPCIS) [7] repository for handled products. A model giving supply chain roles and involved data flows within the RFID-aided pharmaceutical supply chain is depicted in Fig. 2 using the Fundamental Modeling Concepts (FMC) [11]. It shows the interaction between supply chain roles $A$-$F$ and a dedicated service provider $D$ performing anti-counterfeiting checks. We prefer FMC as we understand its notation to be intuitively focusing on actors and their communications. We introduce the role service provider for counterfeit checks, which includes the following tasks. It accesses distributed EPCIS repositories of supply chain participants involved in handling a certain item to create the virtual product history [12]. We agree that reliable tracking and tracing across the entire supply chain can be implemented using RFID [13], but this technology is not designed to be immunized against threats, such as cloning, spoofing or eavesdropping [14].

This pharmaceutical supply chain scenario highlights the risks of counterfeits and the need for product protection. We agree that a high level of supply chain integrity is the basis for reliable product tracking to reduce counterfeits. We consider event data as sensitive that need to be protected since it may expose business relations.

III. RELATED WORK

Fabian et al. were the first to investigate security aspects of EPCglobal components first in 2006. They discussed various drawbacks and highlighted possible threats for Object Naming Services (ONS) [15]. We consider their work as the basis for our research activities on security extensions for EPCglobal networks. Transparent security extensions for existing ONS installations are proven to work without adapting existing implementation while keeping response times and processing load at a moderate level [16]. In the given work, we focus on how to introduce security extensions in a transparent way.

Jing Sun et al. performed research on securing EPC in ONS data based on an adapted Public Key Infrastructure (PKI) [17]. This requires set-up and administration of a hierarchical chain of trust which issues certificates via Certificate Authorities (CAs) [18], [19]. We agree on using PKI as the basis for identification of supply chain participants, since it proved to work even in high-scaled environments, such as user certificates for signing digital messages. From a business perspective, supply chain participants can be forced to use a common PKI, e.g. by requiring a special accreditation before working with new suppliers. Thus, we make use of a central PKI and use X.509 certificates for identification purposes of inquirers in the given work [20].

Evdokimov et al. propose access control for RFID data on a high level of granularity, i.e. their approach provides only one generic key per data source [21]. If the key is exposed, all RFID events for this source are exposed at once. From the author’s knowledge, this is the first contribution focusing on fine-grained access control and individual keys per inquirer and data source. It comes with high demands regarding key management since each data sources and each client identity uses specific encryption keys. Encryption keys need to be maintained and accessed in real-time, which is addressed by in-memory database in our research prototype to keep lookup time low.
IV. Security Components of EPCglobal Networks

In the following, we introduce our experiences with our security extension prototype for EPCIS repositories. The following components are part of our architecture:

- **Querying Party**: client asking for details about a certain good,
- **EPCIS Repository**: contains captured RFID events and processes incoming queries for certain EPCs,
- **Access Control Server**: responsible for key management and logging history of queries,
- **Access Control Client**: responsible for encryption/decryption of communication data and filtering of event data,
- **License Server**: responsible for issuing access rights and applying them to client queries,
- **Trust Relationship Server**: derives client-specific trust score from local and global trust information,
- **Certificate Authority**: serves and validates trusted client certificates,

The detailed functionality of all components in context of the proposed architecture is described in the following.

A. Querying Party

The querying party \( A \) can either be an individual, e.g. a customer, or a business partner, which is possibly unknown. They are interested in gathering details about a certain product identified by its EPC.

B. EPCIS Repository

The EPCIS repository contains location-based event data that is captured by a certain company or department, i.e. the event owner \( B \). The EPCIS offers standardized interfaces for capturing and querying events. We consider EPCIS repositories as standard software, such as FOSTRAK [22], and do not consider its implementation in further details.

C. Access Control Server

The Access Control Server (ACS) keeps track of queries. It extends the functionality of EPCIS repositories by certain logging and key management functionality without modifying the EPCIS repository itself.

D. Access Control Client

The Access Control Client (ACC) is the pendant of the ACS and is responsible for enforcement of access rights. The ACC performs the following tasks:

- transparent encryption of client queries,
- reception of encrypted result sets from the ACS,
- requesting access rights from the license server,
- decryption of result sets, and
- creating a client-specific view on the result set by filtering it accordingly to the retrieved access rights.

E. License Server

The license server extends transparently the EPCIS functionality by issuing access rights while decoupling access control from data contained within the EPCIS repository. As a result, it is possible to change access rights at any time without involving EPC events.

The database of the license server contains the following entities: rules, user groups, users, and identities. A rule describes either a permission or a prohibition. One or multiple rules are assigned to user groups, whereas users are assigned to one or more groups. A user describes an inquirer of the EPCIS identified by its certificate subject. We refer to a concrete installation of client software identified by its certificate fingerprint as identity. One or multiple identities are assigned to a single user.

F. Trust Relationship Server

The Trust Relationship Server (TRS) is an extension that shows how to integrate third-party data into the access control process. It implements exemplarily a scoring algorithm for clients to quantify the reputation of unknown inquirers with the help of indicators of trusted partners in the supply chain. Due to page limitations, we refer to existing related work dealing with trust assignment in supply chains [23], [18], [24].

G. Certificate Authority

The CA is required for establishing a PKI. It issues X.509 certificates for trusted clients and maintains Certificate Revocation Lists (CRLs) when the private key of certificate has been compromised. A valid certificate is required to contact the ACS and the license server to proof the client’s identity. Both actors verify the identity of a connected client by contacting the CA. The identity is used to log client queries and to implement history-based access control [25].

V. Communication Protocol

In the following, we share details about the designed communication protocol, which consists of event data and license data communication.
A. Event Data Communication

When a client contacts the EPCIS to retrieve a set of events for a certain QueryQ, we refer to it as event data communication. It follows a query response conversation as depicted in Fig. 3.

For event data communication, we incorporate SSL to secure exchanged data against third parties eavesdropping or attacking the plain text communication. In addition, the X.509 certificate CertA of client A is required to establish the encrypted communication channel and to identify itself against the EPCIS repository. The client’s certificate is verified by contacting the responsible CA during connection setup. The client can perform the same steps to verify the EPCIS’ identity since the latter’s public certificate is also exchanged during handshake.

After the successful verification of the client, the EPCIS fetches the result set of EPC data RspR from the event repository. The event set is symmetrically encrypted by an AES 256 CBC SymKeyR and sent via the SSL connection to the client. The interested reader may ask, why the event data is encrypted even though it is sent through a SSL connection. The certificate’s owner encrypts it to prevent the client from reading the data, i.e. the symmetric key is not shared with the client itself. The key is only shared with the ACC for decrypting the result set. In other words, the client holding the valid certificate is able to decrypt the response and ends up with the encrypted data set RspR.

The design decision to use a symmetric encryption method was taken, because of performance considerations [26]. The symmetric key is identity-specific to prevent exposure of mass data once the shared key has been exposed. In other words, any client – and also any of its identities – having access to multiple EPCIS repositories receives each event data set encrypted by a specific key. If there is no key for a certain identity, a new offset for the AES cipher is generated and stored in the mapping table of the ACS. As a result, the number of individual encryption keys is increased dramatically while in-memory technology is used to keep the additional key lookup latency low. For any subsequent encryption attempts the ACS needs to be contacted with a client’s identity to provide the latest symmetric key for a certain identity. Further, the ACS enforces access key replacement after a specific time interval, e.g. one day.

B. License Data Communication

License data communication refers to the enforcement of access rights and event filtering. Fig. 4 depicts the extended communication protocol to retrieve a license for decryption of result sets.

The ACC specifies an ODRL 2.0 request including a list of desired permissions and sends them via a SSL connection [27]. These permissions can be understood as a set of rules that contain permissions or prohibitions for an entity of the result set to perform certain operations. It describes the subset of data the client is allowed to access, which needs to be extracted from the encrypted event set RspR. After performing authenticity checks of the client, the license server contacts the TRS to evaluate the trust level between client A and manufacturer B. Based on the retrieved scoring access rights are derived with the help of access rules. The ACS responds with an ODRL response including the retrieved symmetric key for the client’s identity via the SSL connection.

Once the ACC received the details for the symmetric key, it will derive SymKeyR. With the help of SymKeyR, it decrypts the RspR. At this point the ACC has full access to the encrypted event set. Before returning the data to the querying party A, the received ODRL set is applied and prohibited rows, attributes, or records are filtered from the result set, i.e. only the filtered event set is returned.

VI. Enforcement of Access Rights

We propose to separate grant and enforcement of access rights in RFID-aided supply chains. Our architecture incorporates the functionality of a license server, which is responsible for granting access rights, which are enforced at the latest possible stage, on client site by the ACC. Although, this introduces potential threats, e.g. attacking the enforcement process, the separation comes with major advantages. For example, once new access rights are populated to the ACS, they are enforced the next time the ACC queries the license server. In other words, if a client retrieves a result set, the access rights can be modified until the client wants to view them. Even when the event set was retrieved before the client’s access rights were updated the client only receives the filtered event set accordingly to the latest access rights.

We assume a global RFID-aided supply chain for the pharmaceutical industry, which comes with a potentially unlimited number of business relationships between its participants. Thus, operators of EPCIS are unable to define individual access rights for any potential participant. We introduced a role-based abstraction that summarizes a huge number of clients to a small number of defined roles [28]. Access rights are assigned to roles rather than to individual users or identities. Users are assigned to one
or more roles. The process of access rights enforcement is two-folded in trusts scoring and history evaluation.

In the following, we distinguish between local and global trust scoring. Based on company-specific experiences, a certain trust score is maintained for each client that queried the EPCIS. By predefining a list of trusted partner EPCIS URLs, the TRS can contact them to retrieve trust scores for unknown clients. The latter is referred to as global trust scoring. Combining both local and global trust scoring by processing company-specific trust rules is used to derive a client-specific trust score.

From our perspective, combining freely available data with semantic information can lead to exposure of sensitive business secrets. For example, a competitor of a pharmaceutical manufacturer performs queries in the EPCIS repository to receive details of fictive goods of one of the major ingredient suppliers. Even if the default rule prohibits returning any details about the product, the fact that you return a message indicating the existence of further details for the given EPC exposes your supplier. Let us assume, the competitor continues with a second and third supplier. The existence of further details in your EPCIS can be combined to derive the ingredients of your pharmaceutical goods. Data exposure and its combination can lead to exposure of business secrets.

The recording of the query history and its analysis is the basis for dynamic adaptation of static access rights to prevent exposure of sensitive business secrets. The license server records the history by logging access request. Minutes, days, or even weeks later, a further query can be used to combine sensitive data, thus, it is mandatory to keep a complete history of queries. New entries are appended to the end of the list. During creation of the ODRL response, the history is evaluated using semantic rules to detect potential exposure of business secrets. Once a rule is violated, access rights are adapted dynamically to prevent further exposure and the incident is logged.

VII. Applicability to Event Repositories

Our proposed architecture is designed to extend existing EPCglobal networks without disruption. We verified our work by applying its implementation to the EPCIS repository of the Free and Open Source Software for Track and Trace (FOSSTRAK) project [22]. FOSSTRAK is EPCglobal-certified open-source development. Its EPCIS repository is developed in Java2Enterprise Edition hosted by an Apache Tomcat Web Application Server [29].

Fig. 1 depicts the integration of our security extension in the FOSSTRAK EPCIS repository. The FOSSTRAK implementation offers a query client, which can be used to specify queries for the EPCIS repository with the help of a graphical user interface (GUI). The URL of the EPCIS repository to query is specified in the client’s GUI. Exchanging the URL of the EPCIS repository by the URL of our FOSSTRAK ACC activates security extensions.

The seamless integration of our security extensions requires the derivation of access control requests from the existing SOAP messages exchanged between querying client and EPCIS repository. Listing 1 shows an excerpt from a Simple Event Query embedded in a SOAP message for a key value pair combined with a matching operator. The result set will consist of events for the given EPC code (MATCH_epc). We defined an eXtensible Stylesheet Language Transformation (XSLT) to extract required access rights information from the SOAP message of the FOSSTRAK client.

VIII. Security Evaluation

A transformation towards an RFID-enabled pharmaceutical industry involves the need to provide open interfaces, e.g. to query EPCIS repositories while coping with vulnerable environments [5]. Related research work elaborated threats for an RFID-aided supply chain [31], [32], [33]. In the following, our proposed architecture and its actors are evaluated regarding their capability to resist against selected threats for RFID-aided supply chains.

1) Theft of data: EPC event data is of high interest for competitors and counterfeiters, since it can be abused to derive business relations or to create counterfeits. We consider it as data that needs to
be protected against unauthorized access [34]. Thus, our security extensions implement access control with user-specific keys to protect data and to control data access for certain users or groups.

2) **Eavesdropping:** Anti-counterfeiting involves the exchange of EPCs between a querying party and the EPCIS. Eavesdropping either the query or the response can be abused to derive details about involved business partners even when exchanged data is encrypted, e.g. fingerprinting [35]. We incorporate SSL to prevent eavesdropping of exchanged messages. Due to asymmetric encryption and the resulting individual ciphers, we consider fingerprinting attacks as an unrealistic attack in this context.

3) **ACC attacks:** Attacking the client software for decryption of event sets has become a high risk since it is used in a distributed non-controllable environment. Due to the use of individual client keys, we can revoke a single key of a compromised client without affecting other clients.

4) **Denial of service:** Business processes, e.g. goods receipt processing, are directly coupled via RFID technology with EPCIS repositories and the corresponding business system. They require real-time interaction to leverage performance improvements. As a result, queries need to be performed in sub-second response time while zero downtime is expected. Central architecture components, such as EPCIS repository, should be out of scope for potential attacks since its availability is crucial for business operations in a global RFID-aided supply chain. Our proposed security extensions reduce the probability for denial of service since the exposed data is encrypted. In addition, a valid certificate is required for any query that is forwarded to the EPCIS. Thus, the incorporated PKI might be exposed for attacks, which is still comparable to other PKI scenarios.

4) **Denial of service:** Business processes, e.g. goods receipt processing, are directly coupled via RFID technology with EPCIS repositories and the corresponding business system. They require real-time interaction to leverage performance improvements. As a result, queries need to be performed in sub-second response time while zero downtime is expected. Central architecture components, such as EPCIS repository, should be out of scope for potential attacks since its availability is crucial for business operations in a global RFID-aided supply chain. Our proposed security extensions reduce the probability for denial of service since the exposed data is encrypted. In addition, a valid certificate is required for any query that is forwarded to the EPCIS. Thus, the incorporated PKI might be exposed for attacks, which is still comparable to other PKI scenarios.

**A. Query/Response**

The confidentiality of the EPC needs to be considered in the given scenario. In contrast to bar codes, which identify a group of products, EPCs are used to identify a concrete product instance. This unique identifier can be used to track and trace also individuals and to create customer profiles [34]. For example, if a client asks for details about a certain EPC, one can conclude that the client has direct access to the corresponding pharmaceutical and eavesdropper can create a list of owners of the pharmaceutical. The EPC also contains details about the product class, i.e. eavesdroppers are aware of the category of pharmaceutical owned by the inquirer [36]. These examples outline a small excerpt of possible scenarios in that attackers can derive additional information without gathering details explicitly.

Due to the incorporated Internet connection, the confidentiality of queries and responses needs to be considered. We make use of asymmetric encrypted SSL connections between ACC and ACS and license server respectively. Besides confidentiality SSL connections protect the integrity of data between communication peers. Securing further actors of EPCglobal networks, such as the ONS, is addressed by related work [17].

From the company’s perspective, it is important to know who is requesting information. Firstly, individual querying parties receive specific event sets, e.g. an unknown inquirer will receive only the a minimal set of information about a certain product, whereas well-known business partners will receive details about the manufacturing time, the inventory location or other parties that were involved in the manufacturing process. Secondly, EPCIS operators want to identify inquirers to keep track of shared data and to prevent exposure. From the client’s perspective, it is important to know who will receive the query to ensure the query’s confidentiality as mentioned before.

GS1 defines the use of X.509 certificates for authentication purposes within the EPCglobal network. The only 14 pages of the EPCglobal Certificate Profile Version 2.0 released in June 2010 contains recommendations about how to include global location numbers, device and service information [37].

For authentication purposes, we incorporated a PKI and a trusted CA. Clients need to authenticate its identity using a X.509 certificate containing personal information, such as name, address, etc., which are issued by the CA. X.509 certificates are piggy backed during setup of the SSL connection, i.e. the client, the ACS, and the license server need to hold a valid certificate issued by the trusted CA or any delegate. The X.509 certificate is verified during connection setup, i.e. the CRL of the responsible CA is checked for the certificate and the attributes, such as the validity attributes not-before and not-after. After verifying connection peers exchange of data starts.

**B. Access Rights**

Following the principle to share only the minimum of data that is required to work involves the use of access rights to limit the exposed data, e.g. by filtering [38]. The proposed architecture, the confidentiality of access rights and authenticity of communication partners are protected by design since access rights are exchanged only via SSL connections. The integrity of the message can be considered as protected by using the SSL connection since both communication peers are directly connected.

**C. EPC Event Data**

The EPC event data is exchanged through SSL connections and is additionally encrypted using a symmetric key. In other words, even the ACC receives an encrypted result set and is not able to decrypt it unless the license server returns the client-specific key. Due to the separation of event data and enforcement of access rights, the entire result set needs to be sent to the ACC, which performs rights enforcement. However, to protect confidentiality of
event data, comprising ACC must not result in exposure of event data. Therefore, the symmetric key is only sent after the license server is contacted. Once the key has been sent to the ACC, the event data is decrypted and will be filtered accordingly to access rights. During this phase the result set resides in an unencrypted format in the main memory of the machine executing the ACC.

Attackers can abuse this weakness to derive both: the unfiltered event data and/or the used symmetric key used for encryption of the event data. However, the decision to incorporate symmetric rather than asymmetric encryption was taken with respect to encryption and decryption performance [26]. Nonetheless, we tried to minimize the impact of an exposed symmetric key by using individual keys per identity, which are rotated once an hour. In other words, if a single ACC was compromised the attacker is able to decrypt at best the data that was received within the last hour of operation of this ACC. Using the exposed key to access event data from other ACCs within the same client company fails, since each identity gets a unique key and history assigned. Using separate keys per identity builds the basis to lock a single compromised ACC rather than the entire company and to trace the source of the attack. Once the exposure of data was recognized, the license server or the client resp. notifies the CA to add the certificate to the CRL.

D. Performance Results

In the European pharmaceutical supply chain the ratio of wholesalers and manufacturers is 25:1 resp. 70:1 for retailers and manufacturers. For evaluating the applicability of our security extension, we measured its performance impact. Due to page limitations, we only present the response time behavior of the FOSSTRAK EPCIS repository from a client’s perspective for 10k inquirers for a single supply chain party, i.e. 400 queries per wholesaler resp. 142 per retailer for the given scenario. Our benchmark results were gathered on an IBM Blade HS 21XM Type 7995 running SuSE Linux Enterprise 10 SP2 with the 64-bit kernel 2.6.16.60-0.21-smp. It was equipped with 2 Intel Xeon E5450 CPUs, each running at 3.0 GHz clock speed consisting of four logical cores with 32 kB L1 cache, 12 MB L2 cache. The system was equipped with eight 8 GB DIMM DDR modules to form a total of 64 GB RAM.

We populated 1,032,742 entries of realistic event data of the simulated European pharmaceutical supply chain to the EPCIS repository. Tab. I contains the response time measurements. Instead of using the default MySQL database, we adapted the persistence layer of the EPCIS and ported it to use SAP’s in-memory computing engine as persistency [34]. As a result, our adapted version of the FOSSTRAK EPCIS repository runs on top of a columnar in-memory database, which keeps the data in-memory in a compressed form.

Tab. I shows that the median of the response time for the in-memory FOSSTRAK EPCIS without security extension is 52.4 ms, while its pendant incorporating security extensions has a median of 219.3 ms. In other words, from the perspective of a querying party incorporating our proposed security extensions. Fig. 5 illustrates the performance results as box plots; we concentrate of the measurements within the 25 and 75 percent quartiles and consider the remaining as outliers. We consider the distance between both quartiles in relation to the median as indicator for the derivation from the median. Without security extension the derivation is 31 percent whereas with security extension increased it by 10 percent. We consider the small increase in latency as an indicator for the predictability of the latency and a stable response time behavior, in particular, when considering encryption/decryption and additional communications.

From a security perspective, the performance impact of the proposed security extensions is relatively small. We make use of a steady increasing history, storing all queries for decision making, which need to be scanned for every query. As a result, the response time increases with the length of the history to be scanned. Further, the direct communication between client and EPCIS repository is transformed, encrypted/decrypted, filtered, and two additional actors, the ACS and the license server, need to be contacted when incorporating our security extensions.

Although, the response time increases, it remains below one second. We value the advantages of using our security extension in contrast to the relatively low increase of response time. Ultimately, we consider the performance impact as feasible.
IX. Conclusion and Outlook

The contributions of the given work can be summarized as follows: we defined security extensions to improve resistance of EPCIS repositories for RFID-aided supply chains against common Internet attacks. We shared design decisions and implementation details of our history-based access control and evaluated its applicability with a prototype enhancing the FOSSTRAK EPCIS repository. Future work aims to further improve response times and to gather experiences in real-world scenarios with a huge number of simultaneously querying clients.

REFERENCES


1 All online references were checked on Aug. 11th, 2011.