Credential Management for Automatic Identification Solutions in Supply Chain Management

Marcel Henseler and Michael Rossberg and Guenter Schaefer
Telematics and Computer Networks Group
Technische Universität Ilmenau
[marcel.henseler, michael.rossberg, guenter.schaefer][at]tu-ilmenau.de

***Abstract*—Current systems for automatic identification of goods presume a single administrative domain. However, in supply chain management systems temporary cooperations of multiple companies exist, and the usage of one identification device, such as a radio-frequency identification (RFID) tag, per company is infeasible for reasons of costs, space requirements, traceability, and higher collision rate. This article analyzes the security requirements resulting from the usage of a single tag for multiple companies and proposes a novel system architecture and accompanying cryptographic protocols that address the security objectives entity authentication, controlled access, data confidentiality and integrity, as well as untraceability of RFID tags. The architecture is designed to provide high availability and graceful degradation in case of compromise of system parts. The results of an implementation and simulation study give insights on appropriate data structures for realizing key functionality, and demonstrate that the approach can be deployed with commercial off-the-shelf hardware.**

***Index Terms*—Data Security, Access Control, Identification, Production Management, Architecture**

I. Introduction

THE term automatic identification refers to a comprehensive set of technologies for autonomous recognition of goods, such as radio-frequency identification (RFID) tags, smart cards, and sensor networks. This common target currently promises a significant reduction of costs created by manual object identification and resulting delays. Hence, especially in the RFID sector a fast market growth has been the consequence and the development is expected to carry on further [1].

Such a major success can also be expected to happen in the supply chain management sector, where RFID tags may help [2], [3] to defeat the growing complexity of inter-company relationships. However, current RFID solutions usually assume a single security domain, that is only a single company accesses the tags. In contrast, supply chains require multiple (partially competing) companies to access the tags. In order to avoid the usage of multiple RFID tags (e.g. one per company) per item, a security architecture is needed that allows multiple companies to access only specific data records on the same tag. While on the one hand a complex set of security requirements has to be satisfied for such an approach, some major benefits can be expected, such as reduced costs and space requirements due to use of a single tag per item, improved flexibility in case of changes in supply chain partners, as well as fewer collisions on the media access control layer when accessing tags. The security objectives to be achieved in such a scenario are anonymity, availability, access control, as well as data confidentiality, and integrity. Furthermore, the system needs to scale with respect to the amount of tags and the count of supply chain members, for example.

This paper contains the following contributions: It analyzes the security objectives and other requirements, revealing that a novel architecture is needed. Based on this analysis, it proposes an Automatic Identification Management System for Multiple Security Domains (AIMS4MSD) that accomplishes the goals by introducing a distributed system with central authentication and authorization servers that are operated by a trusted third party (TTP) and preserve the untraceability of RFID tags. The TTP only provides necessary symmetric keys and never gets in contact with the actual data, reducing the misuse potential in case of a TTP compromise. Furthermore, the RFID readers authenticate with the TTP, but they are untrusted in a way that they may only access approved data sets and are not able to compromise the anonymity of the tags. The RFID tags are not assumed to be tamper-proof, and carry only encrypted data and no shared group-keys. In order to enable a memory efficient implementation of the data-on-tag-scheme, a novel integrity mechanism is proposed that requires significantly less authentication data to be stored. For entity authentication and data access a novel two-phase protocol is proposed that is tightly integrated into the AIMS4MSD architecture. To evaluate the feasibility of the approach with current hardware, a simulation study with several different implementation variants of key functionality is presented.

The rest of the article is organized as follows: in section II the background and assumptions on the environment and attackers are given. It is followed by section III, covering objectives for an automatic identification management system, and the related work section IV. The AIMS4MSD approach is described in section V. An evaluation section VI compares the objectives and the achieved results in an analysis and a simulation study. A conclusion in section VII finalizes this work.

II. Background

This background section is split into three parts. In the first part of this section, a short introduction to current automatic identification technologies is given. It is followed by a short description of supply chains, particularly in the context of

the automotive industry. The section closes with an attacker model.

A. Automatic Identification Technologies

In modern supply chains, different types of automatic identification technologies are envisioned to securely identify goods. While simple smart cards are a widely accepted electronic technology to securely identify and store the record of highly priced products, they require direct electric contact for communication, and the contact area needs to be securely mounted at the outside of the product.

These disadvantages are eliminated, if RFID technology can be used. While the most basic RFID tags can only transmit an identification number (class 1), more complex RFID tags (class 2 and 3) compliant to the ISO 14443 [4] standards become available. The latter ones are basically contactless smart cards, which can even be programmed to some degree [5]. Class 3 RFID tags additionally contain a small battery, for example to achieve a larger communication range. The radio interface of all RFID classes allows embedding the tag into the automotive part itself, protecting it from environmental influences. Furthermore, the communication can take place within a given range between tag and reader, easing the communication setup. The major drawback of RFID tags is their limited amount of available energy, which is obtained wirelessly from the reader (class 1 and 2) or from a battery (class 3). This fact leads to three effects:

- The communication range of tags that have a lot of logic resources, thus high power requirements, is fairly limited.
- Asymmetric cryptography, in its currently accepted form, is infeasible as it would require too much calculation effort. Novel approaches are more energy efficient [6], but cannot considered to be secure for the next years as cryptographic algorithms have to indicate their security by not being broken for a period of time.
- As a last effect, tags cannot permanently power a clock. Hence, they cannot directly keep track of time.

Even though primarily designed for measuring and coordination tasks, wireless sensor nodes [7] may also be used for automatic identification and are called class 4 RFID tags in this context. Their advantages compared to passive RFID tags are the higher transmission range, permanently powered clocks, the use of environment sensors and a possible multi-hop communication. However, serious disadvantages exist, too. Wireless sensor nodes are more expensive and their batteries drain within years. As products can stay in the supply chain cycle for many years, wireless sensor nodes, as well as class 3 tags, are currently no adequate choice for a supply chain management solution. Therefore, for the rest of the paper a passive RFID class 2 solution is presumed, but future upgrades to wireless sensor nodes are taken into consideration.

This technology leads to three main tasks that an automatic identification system has to accomplish:

1) *Secure and anonymous localization of relevant tags:* Before any other communication, this service allows the communication partners to present their legitimate system affiliation without requiring the tag to reveal its identity.
2) *Authentication & access control:* All participating RFID readers and tags need to be authenticated in a second step and access control mechanisms must assure that no data is accessed for which the particular instance does not have sufficient access rights.
3) *Data management on tags:* Any data must be stored securely and the generated data blocks need to be organized to save memory in order to comply with the restricted resources of class 2 tags.

B. Supply chains in the Exemplary Context of the Automotive Industry

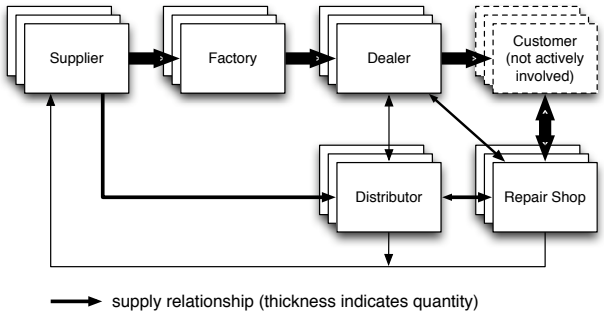

Fig. 1. Simplified structure of supply chains in the automotive industry

Modern supply chains involve complex relationships between companies in order to minimize costs. Figure 1, derived from [8], illustrates a simplified flow of goods within automotive industry, which is one of most progressive ones in this context. Hence, it serves as an example for this article, but statements and conclusions are applicable to any supply chain with any kind of valuable products and the potentially given backward flow of parts that are in need of repair. Examples of other applicable supply chains include aerospace, computer and heavy equipment industries. The given approach is unlikely to be applied by food or clothing industries as the required class 2 RFID tags are too expensive, and returned products do not re-enter the supply chain. In the figure, edges of different width indicate supply relationships of different scales between the different classes of stakeholders. The figure simplifies a real supply chain in two ways: On the one hand, every class represents a nearly arbitrary amount of instances, such as manufacturing sites. On the other hand, each supply relationship can be performed by multiple transport companies, having again multiple clients including competitive automotive companies.

To quantify the scalability requirements of an automatic identification system, the number of items in the supply chain system must be estimated. This number depends on the number of compound products in the system and the number of tags per product. Just for the sake of an example: if a country like Germany, having about 55 million licensed vehicles and trailers [9], introduced such a system by law in order to record the history of car parts for liability reasons and a number of

200 valuable parts existed per vehicle, about 11 billion items would exist in the chain.

C. Supply Chain Participants & Attacker Model

Within the supply chain scenario, several major parties with different interests exist:

- **Suppliers and manufacturers** want to securely store and share data on the RFID tags and perhaps keep the identities of subcontractors secret.
- **Transport companies** need temporary read/write access to RFID tags in order to organize logistics.
- **Trusted third parties** implement authentication and key management services, but will not access the actual data stored on the tags.
- **Customers** have a major interest in their anonymity and the secure handling of data belonging to them. However, they only have a passive role as they do not access the tag directly.
- **Outsiders** represent the most diverse party. However, advanced attackers with access to sophisticated technical equipment and a strong motivation are assumed to exist.
- **Governments** are a special external party, with an interest to monitor the actions of participants and the power to force TTPs to cooperate by law, for example.

Given these participants, an automatic identification system needs to face two levels of attackers:

1) **External Attackers** are the weakest attacker type. They can try to observe tag movement or to passively listen to communication between tag and reader. A more advanced external attacker might be able to actively interfere this communication, construct own tags, or manipulate original tags. It could also try to inject malicious data into the tags [10].
2) **Internal Attackers** are considered to be adversaries, who possess a legitimate RFID reader. For example, participating companies may aim at the identification of subcontractors, or at obtaining or modifying private data from other companies, like the identities of subcontractors or the period of warranty. Also, a combination of attack strategies involving physical tampering of tags by internals is conceivable.
   Despite its name even a TTP is vulnerable, for example to governmental surveillance or compromise.

These attacker models lead to a set of security objectives that TTP, readers, and tags must comply with. They will be presented, among others, in the next section.

III. Objectives for Automatic Identification in Supply Chains

Within the given scenario the following objectives have to be fulfilled:

1) The **Security** of the overall system is a premise for its success. This complex objective is subdivided into the following parts:
   a) As the detailed structure of supply chains is often considered to be secret, the **Anonymity**, that is the confidentiality of the identities of stakeholders, must be protected by the system. Furthermore, customers should not be traceable when using a car with built-in RFID tags. The stakeholders do not trust each other transitively. Hence, they must not be able to identify, for example, suppliers of suppliers and shall only be able to identify tags based on their own identification numbers, not a global one.
   b) **Data Confidentiality** is a second security goal. Every stakeholder needs to be able to store data with the confidence that only designated parties can read it. A direct consequence is the necessity for the encrypted storage of data on the tag itself. This principle ensures that no single superior entity controls all data and assures that a system compromise will only affect data on tags that are within range of a (compromised) reader. Consequently customers can be assured that their data cannot be accessed as long as they are not in the range of an adversary.
   c) **Authentication and Integrity Protection** ensures that data on the tags is only modified by authorized entities. Any unauthorized modifications must be detected so that countermeasures can be taken.
   d) **Graceful Degradation** in the case of a compromise, meaning robustness towards disclosure of secrets, has to be ensured. A disclosure must not have other than temporary or spatially bounded effects on the security of the system.
   e) **Access Control** must be handled by a role-based access control mechanism [11] within the complex scenario. This ensures that parties may only access data relevant to their role. Furthermore, access control has to be done online as rights must be revocable at any time, for example, if a transport company is changed.
   f) The **Availability** of the RFID system has to be ensured as otherwise the supply chain would be severely disturbed.
2) The **Scalability** of the system needs to be guaranteed in multiple dimensions. First, the number of objects in the supply chain leads to a huge amount of active tags. Second, the system must scale over the stakeholder count. Furthermore, the number of readers can become very large as well as the number of access actions on tags.
3) **Practicability:** The system must be constructible with current state-of-the-art RFID tags. This leads to four subgoals:
   a) **Cost efficiency** is an important property for the participating suppliers, manufacturers, and transport companies.
   b) A **low memory footprint** in the tags must be achieved in order to keep cost and power consumption low.
   c) The system shall use **symmetric cryptography only** as conventional asymmetric cryptography is

considered infeasible for reasons of computing power, and novel asymmetric approaches are not yet established in the security community.

d) Potentially **tampered tags** must be considered as the costs for tamper-proof tags are considered to be too high. This implies, for example, that the overall system must not depend on the intactness of a single tag and that data, which is stored on a tag, must be encrypted and integrity protected. However, some physical attacks on non-tamper-proof devices, such as destruction and cloning, cannot be prevented by software measures and are not in the scope of this article.

IV. Related Work

The related work section consists of two topics: First, an overview of other security-aware automatic identification systems is given. In the second part, different anonymization techniques for RFID systems are addressed.

A system for authorization and access control in the context of automatic identification systems is MASC (Monitoring and Security of Containers) [12]. Its primary goal is the secure identification of ship transport containers with the help of sensor nodes. The energy saving requirements as well as the involvement of multiple parties make it somewhat comparable to the given supply chain scenario. The approach relies on a single central server that is queried by every stakeholder. However, due to the fact that this server contains all data as well as all key material, it offers an enormous potential for misuse. A compromise of this single entity allows an adversary to access and modify previously stored data, and to insert arbitrary new data. Therefore, the data confidentiality and graceful degradation requirement is violated. Furthermore, the system offers insufficient anonymization capabilities for use in supply chains that involve activated tags in the vicinity of customers.

The security framework proposed in [13] is based on the EPCglobal Network [14]. It offers a multi-security-domain authorization and access control system that is based on direct exchanges between backend servers of two parties for every access. The major problem in the approach is a violation of the anonymity requirement as every involved tag is mapped to an owning stakeholder without further authentication. Another problem is that there is no data-on-tag concept present. Threrefore, in case of a compromised third party a graceful degradation cannot be guaranteed.

Generic systems for authorization and access control exist for some time already, and with Kerberos [15] as the probably most prominent solution that only uses symmetric cryptography, they have been deployed widely. However, they are not applicable to automatic identification systems as anonymity was not a design objective and is therefore not preserved. More specialized secure data management systems exist with SELMA [16] and REMPLI [17], which acquire energy measurement data in a distributed way over public networks. Nonetheless, they require tamper-proof devices and do not preserve the customers anonymity as both is not a problem in power-line networks. Another specialized approach is PABADIS'PROMISE [18], which is also aiming at factory automatization with RFID technology and providing a security infrastructure. But due to a slight differences in the assumed scenario, the extensive research results of the project are not applicable to our approach as RFID tags are only used within the protected factory floor environment. Therefore, security considerations such as anonymity and tampered tags are not addressed.

On the other hand, several anonymization techniques exist for RFID systems, usually aiming at an obfuscation of the tag identification number, in a way that only authorized parties can identify it. Such obfuscated identification numbers are called pseudo IDs hereafter. Ari Juels proposed an one-time-pad-like scheme [19] in this context. Every participating tag has a set of multiple pseudonyms, which are changed regularly and updated with every successful authentication. In order to protect the tag from being traced, the number of available pseudonyms needs to be large as an attacker that knows a certain part of the pseudonyms of a tag is again able to trace it. Therefore, the number of pseudonyms need to be very large within the given scenario since automotive parts may stay in cars for a long period of time without being read by a legitimate reader, for example like in a repair shop, leading to a large amount of required storage space on tags and backend servers. Furthermore, the updates of many pseudonyms require too much bandwidth to be feasible.

Another approach [20] uses hash chains to generate new IDs with every query by a reader. Central authentication servers can generate these hash chains as well, and therewith identify the particular tag. External attackers are insufficiently addressed in this approach as they can let the tags generate an arbitrary amount of hashes. By doing so, the authentication servers will become a bottleneck if a large number of tags is in use. The servers have to generate hashes for every potential tag for every authentication. As a result, the required calculation effort per single authentication grows linearly with the total number of tags in the system. There is no upper limit of how many calculations are required. Even with the optimizations proposed in [21], the system is infeasible for supply chains with billions of tags.

The YA-TRAP protocol [22] introduced the transmission of time stamps, which are protected by a keyed-hash message authentication code (HMAC), to be used as identification numbers for RFID sensors. This mechanism ensures that the static identity of a tag, which is the HMAC key in this case, is never transmitted. However, the particular implementation has problems with unauthenticated time stamps as attackers can transmit time stamps of future points in time, leading to an effective denial-of-service attack. This unauthenticated time stamp problem was eliminated in [23], but the introduced hash signatures require group keys between all tags and readers. Hence, it is infeasible without tamper-proof tags and readers.

In summary, existing authorization systems do not regard anonymization sufficiently. On the other hand, existing anonymization techniques have major limitations with regards to scalability and robustness against denial-of-service attacks, or make use of group keys. Hence, an approach is required

that combines both: anonymization and access control over multiple security domains.

V. The Automatic Identification Management System for Multiple Security Domains

This section presents the novel AIMS4MSD approach in three parts. An overview of the general architecture is followed by the description of an anonymization system. The section closes with the specifications of an authentication and a data access protocol.

A. System Architecture

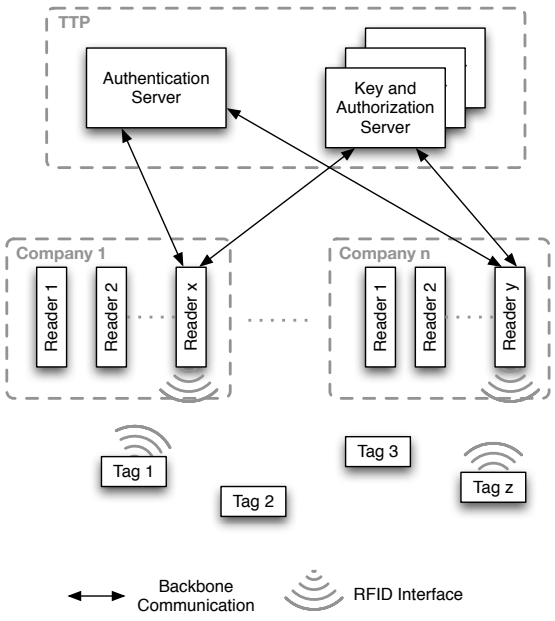

Fig. 2. Architecture of AIMS4MSD

The suggested novel approach consists of four different entities as shown in figure 2.

1) Anonymity Preserving Authentication Server: This single logical instance is operated by a trusted third party. Its primary task is to provide a mapping between the dynamically changing pseudonyms of tags to internal static identities, which are never revealed to the RFID readers. Hence, it is responsible for the anonymity of the tags.

The pseudonyms of tags must contain as little structured information as possible as such information, including the name of the associated authentication server, may make the tag traceable. Therefore in the best case for anonymity, only a single logic authentication server exists. For reasons of availability there may be a replication to multiple redundant physical servers. After obtaining the unique permanent identity of a tag, the authentication server delegates further decisions of access control and key negotiation to an authorization and key server that is responsible for the particular tag.

2) Key and Authorization Servers: These instances are operated by the TTP as well. Their task is the negotiation of keys between readers and tags if the access control mechanism allows a communication between them. The communication between key server and authentication server is not done directly, instead tickets are used that are redirected by the readers. This property, as well as their assignment to check authorizations, makes the key and authorization servers somewhat comparable to the ticket granting servers in Kerberos.

Multiple logical instances of the key and authorization servers are supported for reasons of availability and each key server is responsible for a certain amount of tags. The mapping of tags to key servers is done stochastically, in order to ensure the anonymity of the tags. Otherwise, the change of one stakeholder in the system, e.g., a supplier, could be detected by other parties as it would lead to a sudden change of the used key servers.

3) Readers: Every participating company in the supply chain has one or more readers. These readers are not further trusted. They may only access certain data structures on certain tags if they have the corresponding read, modification, and appending rights. Especially, they must not get to know the static inner identity of any tag. The secure uplink to the authentication and the key servers is provided either by transport layer security (TLS), IPsec, or comparable mechanisms. The wireless link to the tags does not need a further protection as this is provided by the authentication and data access protocols.

4) Tags: The tags carrying identification and data of a part in the supply chain are initialized by the TTP with pre-shared keys for communication with the authentication server. This deployment scenario is somewhat comparable to the hand out of subscriber identity cards (SIMs) in the Global System for Mobile Communications (GSM)/Universal Mobile Telecommunications System (UMTS) [24]. As the tags are assumed to be not tamper-proof, the keys are individual for every tag. The stored information on the tags is encrypted and authenticated by a different set of keys to prevent unauthorized access by outsiders, even if the tag is tampered.

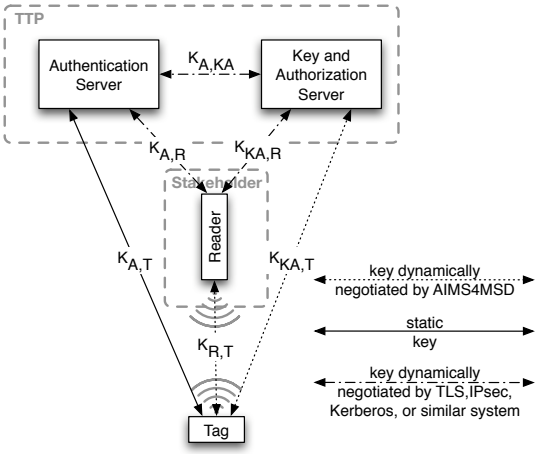

Fig. 3. Trust and Key Distribution Model in AIMS4MSD

Figure 3 illustrates the resulting trust and key distribution model in a simplified form that shows only single instances of the entities. There is only one static key per tag within AIMS4MSD that is individually shared between tag and authentication server ($K_{A,T}$). The readers communicate with

the authentication server, as well as with the key and authorization servers, over an encrypted tunnel using standard techniques. The key $K_{A,KA}$, between key and authorization servers and the authentication server, can either be statically configured, or periodically changed by Kerberos exchanges, for example. The two remaining keys $K_{R,T}$ and $K_{KA,T}$ involve tag communication and are dynamically generated by AIMS4MSD with every authentication.

B. Anonymity Mechanism

Within the given architecture the different services of the automatic identification system are realized. The most important mechanism of the system ensures the anonymity of customers and ensures that supplier relationships remain secret. As existing protocols have been revealed to be insufficient in section IV, a new two phase protocol is proposed as illustrated in figure 4.

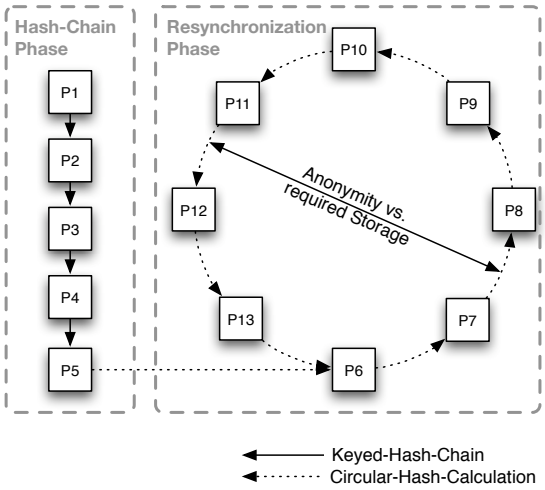

Fig. 4. Pseudonym Calculation

The approach is also based on pseudonyms, and in the first phase tags generate hash-chains that conform to the algorithm given in [20]. Given a value of a hash chain, generated by an arbitrary cryptographic hash function H, a new pseudo ID is derived by hashing the current value with a second cryptographic hash function G. For a start value S the first pseudo ID P1 is G(S), the second pseudo ID P2 is G(H(S)), P3 is G(H(H(S))), and so on. A clear advantage of this method is the constant, low amount of required storage space on the tags (for example 80 bits). However, if the number of pseudo IDs generated by the tag exceeds the number of pseudo IDs precomputed by the server, no further communication is possible as the server would have to calculate the hash chains for all participating tags. Therefore, it is suggested in this article that the tag will switch into a resynchronization phase after a fixed number of pseudo IDs were generated.

In this case, an arbitrary, but fixed number of pseudo IDs are being rotated, allowing an adjustable degree of anonymity and expenses. On the one hand, the larger the amount of possible identifiers, the harder it is for an attacker to learn all pseudo IDs of a victim and make it traceable. On the other hand, a large amount of IDs requires a lot of calculation effort or storage on the authentication server, which maps the dynamic IDs of tags to their static identifiers. The effort of the authentication server and an attacker is proportional, and the cycle size shall be chosen as large as the infrastructure can cope with. In order to keep the number of required pseudo IDs low and the attackers' effort high, the tags will generate new pseudo IDs only with an exponentially growing *recess period*, if the transmission of a pseudo ID is not followed by a complete authentication. As RFID tags cannot measure time directly, the number of clock cycles that the tag is under power are measured to determine whether the recess period is over. A reader has to power the RFID tag for the whole recess period, otherwise the memory cells for the tick counter and therefore the required time period is reset. This behavior prevents the attacker from tracing tags with loose contact and is applicable to class 2 RFID tags as long as they include a simple high clock frequency sensor to prevent attackers to increase their operating frequency. Furthermore, in order to prevent a denial of-service by creating too long recess periods, a cut-off recess time is proposed. A cut-off time of 10 minutes seems adequate for the automotive scenario as this event is most likely to occur only when a device comes from a customer back into the chain. The cut-off time may be chosen differently for other scenarios.

During the resynchronization period, pseudo IDs are derived by the tags with the function $PseudoID_{Counter} := H(K_{A,T} \mid\mid CycleID \mid\mid Counter)$, where $K_{A,T}$ represents a static key, which is individually shared between authentication server (A) and tag (T). The $CycleID$ is also a shared secret between A and T, but it is changed with every successful resynchronization. Thus, it prevents long-term attacks as the tag generates a new cycle of pseudo IDs. The $CycleID$ is concatenated to a $Counter$ that represents the position of the pseudo ID within the cycle and must be stored persistently on the tag. It is incremented modulo the length of the cycle with every unsuccessful authentication and therefore every pseudo ID generation. This pseudonym construction allows cycles of arbitrary lengths, without requiring more storage space on the tags as the pseudo IDs may be calculated on demand by it. In contrast, the authentication server cannot perform this on-demand calculation as it has no access to the tags' current $Counter$ state and hence limits the cycle length.

In order to guarantee a timely service, the authentication server needs to store all possible pseudo IDs of both phases and of all tags in a large database. Under normal conditions the server looks up the internal ID of the tag in its database by using the received pseudo ID. It will then delete the used pseudo ID and insert a new one by extending the hash chain. In the case of n authentication failures, the authentication server needs to delete $n+1$ pseudonyms and extend the hash chain by $n+1$ entries. Only if the tag was in resynchronization mode, all pseudonyms of the tag would have to be changed as the tag will switch to a new hash chain and a new pseudonym cycle. The fact that the second phase generates more server load for updating pseudonyms is the major reason for the introduction of the first phase that only fails under attack scenarios.

C. Authentication and Data Access Protocols

The proposed communication protocol consists of two phases: an authentication phase, where a pseudo ID of a tag is mapped to its internal static identity, and a data access phase, during which the actual read and write operations take place. Both are described in detail in the following.

1) Authentication Phase: The authentication protocol involves communication between tag (T), reader (R), and authentication server (A). Upon completion of the dialog, the tag is authenticated and the tag and the reader share a session key for further communication. Besides, the reader has obtained all required information for the data transfer protocol. The authentication steps are illustrated in figure 5 and explained in more details in the following:

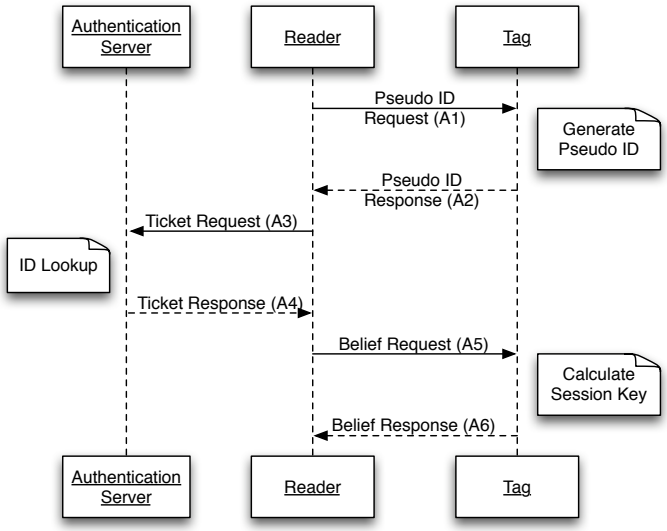

Fig. 5. Sequence Diagram of the Authentication Phase

$$R \rightarrow T: \ \text{authentication request}, \ r_{R1} \tag{A1}$$

$$T \rightarrow R: \ PseudoID_i, \ r_T, \ H(K_{A,T} \ || \ PseudoID_i \ || \ r_T) \tag{A2}$$

Starting with message (A1), R initializes the authentication over the RFID interface by formulating a request to T with the nonce r_{R1}. T answers in message (A2) with a fresh pseudo identity and an own nonce r_T as well as with a hash signature of both fields. The signature is later used by A to verify that the pseudo ID was actually sent by T.

$$R \rightarrow A: \ \Big\{ ID_R, \ t_R, \ r_{R1}, \ PseudoID_i, \ r_T, \ H(K_{A,T} \ || \ PseudoID_i \ || \ r_T) \Big\}_{K_{A,R}} \tag{A3}$$

$$A \rightarrow R: \ \Big\{ ID_A, \ r_{R1}, \ KA, \ PseudoID_i, \ K_{R,T}, \ \{ID_R, \ ID_T, \ KA, \ K_{KA,T}, \ t_A\}_{K_{A,KA}}, \ r_A, \ t_{ticket} \Big\}_{K_{A,R}} \tag{A4}$$

In a third step, R redirects the received message together with its nonce r_{R1} to A (A3). The identity of R and a field (t_R) ensuring the freshness of the message is sent implicitly by TLS or IPsec, which also encrypt the message with a shared key $K_{A,R}$ (represented by curly braces with the index $K_{A,R}$). Then A performs a lookup of the ID of T by using $PseudoID_i$. If the tag transmits a valid pseudonym/signature pair, the server will answer with a ticket for the responsible key and authorization server (KA), which is used for data access later on. The ticket consists of a field encrypted by $K_{A,KA}$ and a time-stamp t_{ticket} that allows KA to lookup the right key. Furthermore, A transmits a session key $K_{R,T}$ (A4) that can be generated by the tag as well as the authentication server by calculating $f(x) := H(K_{A,T} \ || \ ID_T \ || \ r_T \ || \ r_{R1} \ || \ r_A \ || \ x)$ for the constant value $x =$"0". The session key $K_{KA,T}$ is derived by the tag by using a constant "1". If T had been in resynchronization mode, a new $cycleID$ and a new start value of the hash chain would be generated by calculating $f(x)$ with the constants "2" and "3".

$$R \rightarrow T: \ r_A, \{r_{R2}, \ r_T\}_{K_{R,T}} \tag{A5}$$

$$T \rightarrow R: \ \{r_{R2}\}_{K_{R,T}} \tag{A6}$$

With the last two messages (A5 and A6) of the dialog, T learns r_A, calculates the session key $K_{R,T}$, and both entities convey their belief in this key.

2) Data Access Phase: After having assured their confidence in the negotiated key, tag and reader may start to transmit the actual data. The involvement of the authentication server is no longer needed. Instead, communication with the TTP is done using the KA that is responsible for the tag. The protocol consists of the messages shown in figure 6.

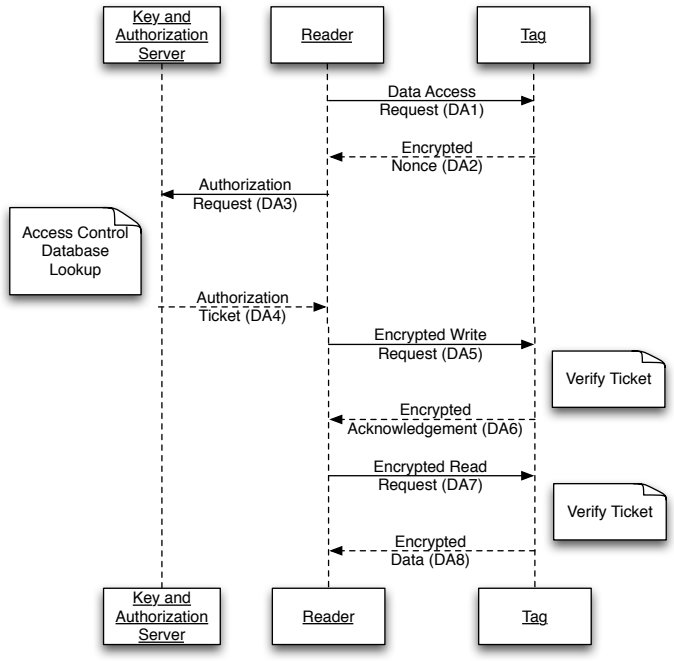

Fig. 6. Sequence Diagram of the Data Access Phase

$$R \rightarrow T: \text{ data access request} \quad \text{(DA1)}$$

$$T \rightarrow R: \{r_T\}_{K_{R,T}} \quad \text{(DA2)}$$

The data access dialog starts with a request (DA1) by the reader. This request is answered by T with a fresh nonce (DA2), identifying all subsequent operations.

$$R \rightarrow KA: \Big\{ID_R,\ r_{R1},\ KA,\ PseudoID_i,\ r_T,\ RWrequests[],\ t_{ticket}, \{ID_R,\ ID_T,\ KA,\ K_{KA,T},\ t_A\}_{K_{A,KA}}\Big\}_{K_{R,KA}} \quad \text{(DA3)}$$

$$KA \rightarrow R: \Big\{r_{R1},\ ID_R,\ PseudoID_i,\ r_T,\ RWresponse[],\ DK[],\ IDK[],\ HMAC(K_{KA,T},\ ID_R \mid\mid PseudoID_i \mid\mid r_T \mid\mid RWresponse[])\Big\}_{K_{R,KA}} \quad \text{(DA4)}$$

Afterwards, R requests data access (DA3) to a specific set of named data fields in T (denoted by $RWrequests[]$). KA performs the authorization by verifying the readers transmitted ticket and an adjacent lookup of R, T, and the data fields in its access control database. A successful verification is followed by message DA4, containing a response $RWresponse[]$ that is used by the tag to unlock access to a set of specific data sets. It also contains data keys $DK[]$ and integrity data keys $IDK[]$ for these sets. The last element of the message is a signature that will be used by the tag to verify the authorization of R later on. Again, the identities of KA and R as well as the nonce may not have to be transmitted explicitly iff the underlying security protocol performs authentication and replay protection. After this initial dialog, the actual read and write operations take place.

$$R \rightarrow T: \Big\{r_{R2},\ ID_R,\ PseudoID_i,\ r_T,\ RWresponse[],\ RWindex[],\ \{\text{newData}\}_{DK[x]},\ MAC(IDK[x], \{\text{newData}\}_{DK[x]}),\ HMAC(K_{KA,T},\ ID_R \mid\mid PseudoID_i \mid\mid r_T \mid\mid RWresponse[])\Big\}_{K_{R,T}} \quad \text{(DA5)}$$

$$T \rightarrow R: \{r_{R2},\ ID_R,\ PseudoID_i\}_{K_{R,T}} \quad \text{(DA6)}$$

As T is not tamper-proof, any permanently stored data is encrypted and authenticated by R before transmission. For these cryptographic operations the keys $DK[]$ and $IDK[]$ are used, respectively. After encrypting and authenticating the data, R will initiate a transfer (DA5) by sending the indices $RWindex[]$, where encrypted data and message authentication code (MAC) shall be written to, as well as the data and the MAC themselves. Also the signature of KA must be included to assure a previous authorization. T starts processing the message by checking whether the accessed indices match values in $RWresponse[]$ and verifying the validity of $RWresponse[]$ by checking the HMAC. This check of the $RWresponse[]$ field prevents readers from writing to areas that they are unauthorized for. However, the mechanism only provides an additional protection against attacks over the air interface. Attackers with physical access may circumvent this protection by directly manipulating the memory state. An acknowledgment (DA6) will be sent if the operation was performed successfully.

$$R \rightarrow T: \Big\{r_{R3},\ ID_R,\ PseudoID_i,\ r_T,\ RWresponse[],\ RWindex[],\ HMAC(K_{KA,T},\ ID_R \mid\mid PseudoID_i \mid\mid r_T \mid\mid RWresponse[])\Big\}_{K_{R,T}} \quad \text{(DA7)}$$

$$T \rightarrow R: \Big\{r_{R3},\ ID_R,\ PseudoID_i,\ \{\text{Data}\}_{DK[x]},\ MAC(IDK[x], \{\text{Data}\}_{DK[x]})\Big\}_{K_{R,T}} \quad \text{(DA8)}$$

R might also need to access the previously stored data. For this task R has to send a request (DA7) with the indices $RWindex[]$ to read, along with a valid signature of KA. As already indicated, T provides read access only to indices that are allowed in the $RWresponse[]$ array. If access is allowed, T will transmit (DA8) the encrypted data along with a MAC to R.

D. Data Organization on Tag

In order to detect data modifications on tampered tags, e.g. by physical actions, MACs can be used. However, in a straightforward approach, one MAC needs to be kept for every data set in order to prevent processing of compromised data. As MACs are currently assumed to require a size of at least 96 bits, the size of the authentication data can easily dominate the actual data, which might be only a few bits. Some space may be saved by aggregating data fields, that are accessed by participants having the same roles and calculating a single MAC over the whole set. However, the scheme requires still a lot of storage, depending on the number of access roles.

Therefore, another scheme with a single MAC over all data sets is proposed. This MAC will secure all data against outside attackers. Authenticated readers are prevented from changing data that they may not access in two ways. First, the tag actively prevents writing to areas that are not covered by the ticket of KA. Second, a message integrity code (MIC) is calculated and stored for every data field that is accessible by a distinct role. This MIC could, for example, consist of the first 8 to 16 bits of a MAC over the data field and is, therefore, severely shorter than a normal MAC. With a 16 bit MIC the probability that an attacker can find a valid MAC by performing one guess is about 0.0015%. The secret MIC key prevents collision attacks. The higher probability in comparison to MACs can still be considered secure as whenever the outer MAC value is correct and the inner MIC verification failed, it is obvious that the tag was tampered by an internal attacker. Hence, an investigation can be made and sanctions may be applied to the last authenticated readers,

limiting the possibility to find a valid MIC by brute force attacks. The key for the MIC is individual for both tag and data field in order to prevent unauthorized copying of valid data.

In case the outer MAC is incorrect, the tag was possibly tampered by an external attacker or the last reader behaved incorrectly. Unfortunately, no sanctions can be carried out as the attacker cannot be securely identified. The resulting data structure is shown in figure 7.

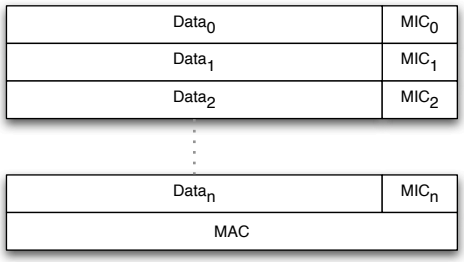

Fig. 7. Data storage scheme on tag

VI. Evaluation

The evaluation of the presented approach is split into two parts. Qualitative objectives of the approach are discussed deductively. However, some quantitative aspects require an implementation and a simulation study, which is discussed in the second part of this section.

A. Analysis

The deductive analysis follows the structure of the objectives for the RFID architecture and starts with the **Security** objectives.

1) Anonymity: The pseudonym system makes it impossible for an external attacker to identify tags over long periods and with the introduction of growing recess intervals the traceability of moving tags is effectively inhibited. The success of the recess intervals depends on the tags' immunity against clocking attacks, which is given for modern cryptographic RFID tags. Nonetheless, stationary tags, such as in parking cars, can be recognized at a later point by an attacker. The system makes a clear trade-off between server load and traceability under these conditions.

Attackers that are more powerful by possessing a valid reader are not able to obtain the static identity of tags either. However, they can by definition write own identifiers onto the tag to allow a later recognition. In order to trace a tag in this way, an attacker must perform a large number of authentication and authorization dialogs for this specific tag. These actions can easily be detected by the key and authorization servers so that the certificate of the reader will be revoked by the trusted third party if a certain access rate is exceeded.

The mightiest attack on anonymity can be performed by the TTP or an attacker that obtained access to some of its data. Not only every tag authentication can be monitored, but as they possess all pseudonyms and keys, they are also able to trace arbitrary tags with own readers. Nonetheless, these attacks require online access to the TTP databases; it is not enough to just copy it once as the state information must be preserved.

2) Confidentiality of Data: Any information that is stored on the tags cannot be obtained as long as the TTP and the cryptographic algorithms are not compromised or broken. Internal attackers can only read the information that they are allowed to access by the key and authorization servers.

A detailed analysis of the confidentiality, authentication, and key management aspects of the proposed protocols has been performed with the help of GNY logic [25]. The evaluation is available [26], but too lengthy to be presented in this article.

3) Authentication and Integrity Protection: The data on the tags is protected by a single MAC against external attackers. But as any internal attacker may access the MAC key, it does not protect the data in this case. To separate the access privileges between different legitimate readers, a MIC protection scheme ensures that an internal attacker will be identified with relatively high probability. The TTP is able to generate arbitrary valid data as it has access to MAC and MIC keys of all tags. However, if the attacker wants to access the data of a certain tag, it must compromise the responsible key and authorization server and must be in vicinity of the tag at the same time.

4) Graceful Degradation: By compromising a tag, an attacker does not obtain access to the required keys to access the information stored on it. A compromised reader is the only entity that can procure key material and information at the same time. Nonetheless, only the information that the reader may access can be read. Compromised TTP servers may give an attacker access to key material and to the identities of tags, but the actual data has to be collected from the tags.

5) Access Control: The proposed system performs access control in two ways. First, the tags allow read and write access to readers, based on a ticket of an authorization server. Second, any data on the tags is encrypted and protected by an authentication code to render tampering attacks useless. Keys and tickets are given out by a single instance only: the responsible key and authorization server, which performs as role-based access control for every reader. Thus, the access control criterion is fulfilled.

6) Availability: The most critical path with regards to availability is the connection between readers and the trusted third party. While this may be a problem in some scenarios, similar limitations exist in all data-on-network systems. Furthermore, the functionality of the TTP was split into key and authorization servers and authentication server to avoid the need for a single monolithic system. These servers can also be implemented on redundant hardware, and rate control mechanisms can keep single RFID readers from exhausting server resources as all readers are authenticated. The authentication server is for reasons of anonymity the only component that has to be replicated on hardware basis. Nonetheless, experiences with the home location registers (HLRs) and authentication centers (AuCs) in GSM/UMTS [24] indicate that a such a replication is technically possible and that a sufficient degree of availability can be achieved. The networks operated by the automotive industry, such as

the European Network Exchange (ENX) and the Automotive Network Exchange (ANX), seem to be an ideal foundation for the required highly available communication infrastructure.

The **Scalability** over the amount of readers and tags was another major development objective. But current approaches of using multiple authentication servers by employing a pseudonym prefix for differentiation [20] may lead to a reduced anonymity as a certain car type can, for example, be identified by having x tags of server 1, y of server 2, and z of server 3. For this reason within the proposed system, the authentication server is the only logical entity that cannot be replicated without degradation of anonymity properties. However, it is possible to implement this server on standard hardware that can handle the projected number of tags and access requests for the automotive scenario. Further details are shown and discussed in the implementation section.

The request for **Practicability** is the third major criterion to discuss. It consists of the following sub-objectives:

1) Cost efficiency: While the additional monetary overhead of operating a TTP and using data-on-tag technology will increase costs for some parts of the system, the total costs are not believed to increase. As multiple companies share the same tag, these costs can be shared and will outperform the purchase of one tag per company. Furthermore, the approach eliminates the need to kill RFIDs after production and retagging the parts in the event of back flow.

2) Low memory footprint: The combination of MACs and MICs represents a novel data saving technique, which allows the system to significantly reduce the amount of saved authentication data. Furthermore, the amount of required security against external and internal attackers can independently be traded against a higher memory footprint, by increasing the size of the MACs and MICs, respectively.

3) Symmetric cryptography only: The proposed system does not utilize any asymmetric cryptography whenever it communicates with a tag.

4) Consideration of tampered tags: All security deliberations take a manipulating of tags by internal or external attackers into account. An attacker who modified a tag cannot insert or modify data without being detected, nor is it able to retrieve additional information from the stored data. Physical access could lead to a compromise of the key $K_{A,T}$ and will lead to a breach of the anonymity objective. However, in this case the attacker could also simply replace the tag by a traceable one to achieve its goal even easier.

B. Implementation

As already identified in the last section, the authentication server is the system that has to deal with the highest work load in the AIMS4MSD approach. Therefore, several different implementation variants of the authentication server have been prepared, in order to evaluate the feasibility of the approach with current hardware in a simulation study. Examined are both: required storage as well as the number of required hard disk accesses. The simulation itself was conducted by using the discrete event simulation framework OMNeT++. Figure 8 illustrates the graphical user interface of the simulation system.

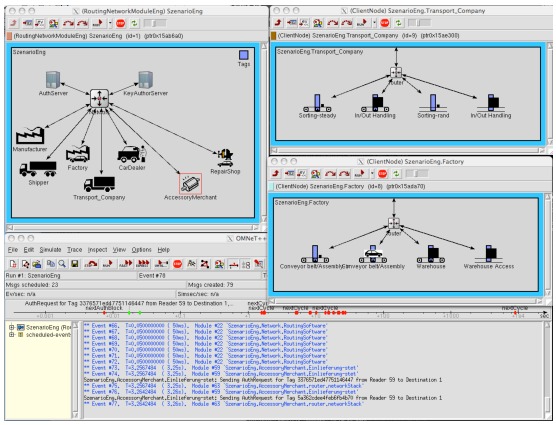

Fig. 8. AIMS4MSD Simulation User Interface

As already described in the background section, a country of the size of Germany is expected to have about 11 billion tags that are built into automotive parts. An expected lifetime of about 15 years, leads to a deployment rate of nearly 750 million new tags per year or 25 tags per second. If the tags are accessed by 20 readers during transportation and assembly, the authentication server must handle an average of 500 requests per second. The frequency of accesses is most likely not uniformly distributed. Therefore, a more detailed load model was developed to obtain more accurate results.

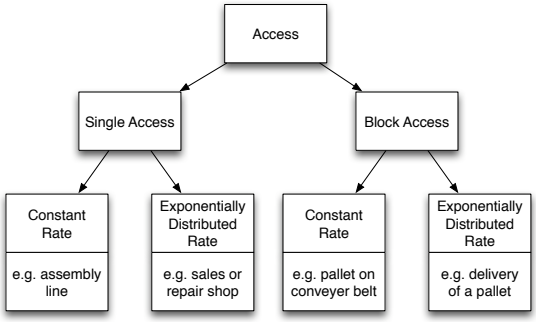

Fig. 9. Assumed Access Schemes for RFID Tags

This load model (see Figure 9) differentiates between uniformly distributed access of entities on a conveyor belt and an exponentially distributed access for entities that are exposed to no-periodic accesses. Furthermore, pallets exist that are accessed in parallel with only $7.5ms$ delay, which is required for data link layer access. The given average of 500 authentication requests per second and the assumption that about half of the accesses happen on pallets were used to conduct a simulation in order to better understand the structure of requests on the server.

Figure 10 shows a typical example of the number of authentication requests that the server has to answer each second in steady state. For the system a complete supply chain with pallets of up to 1000 parts and conveyor belts with up to 25 parts per second were considered. And even though conveyor

belts lead to a uniform background access, the consideration of pallets leads to a highly bursty load behavior. Therefore, authentication, as well as key and authorization servers, need to cope with peaks of twice as many requests as the average load.

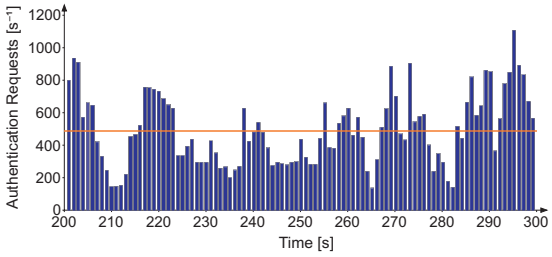

Fig. 10. Assumed load on authentication server

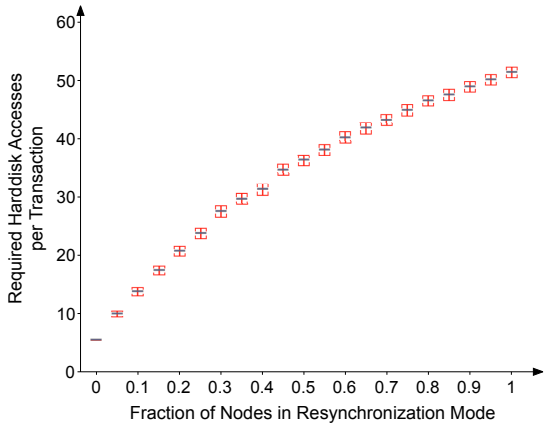

Fig. 11. Influence on resychronization probability on server performance

In order to cope with about 1,000 requests per second, appropriate data structures must be used by the authentication server. More specifically, the number of hard disk accesses in the authentication server must be minimized. For this reason, several tree structures and hash table algorithms were simulated, in order to estimate the required number of hard disk accesses to perform the mapping between pseudo ID and static identifier. Within the simulation, the pseudonym chain of the first phase is assumed to have a length of 10 and the ring cycle consists of 15 entries. Single reading errors lead to the generation of a new pseudonym with a probability of 10%. As result of these simulations, all tree structures were found to create an insufficiently high amount of accesses due to the number of tags in the system. Therefore, they are not further discussed in the following. On the other hand, hash-table-based structures were found to perform very well as collisions usually can be handled without a second access to the hard disk as long as the entries are stored on the same block. Because of this effect, all modeled hash strategies (linear probing, chaining, chaining with sub-buckets) lead to about the same results. All of them are extremely efficient in dense conditions, and up to 80% of the raw memory space can be used.

With an assumed size of 80 bits per pseudo ID and an internal pointer of 64 bit, each entry in the hash table has a size of 18 bytes. For every tag the system has to store 25 pseudo IDs, leading to a total of 450 bytes per tag. Thus, for 11 billion tags an effective storage of 4.5 terabytes (TBs) is required. When the hash table is filled to 50%, 9TBs raw space must be available, a value that modern storage systems easily achieve.

The actual number of required hard disk accesses is more difficult to estimate as it depends on the fraction of tags that are in resynchronization phase, thus influencing how many entries in the hash table have to be refreshed. Again, a simulative evaluation was conducted to examine the influence on the required effort. The authentication server implements a simple hash table with linear probing for this experiment.

The results in figure 11 show the fraction of tags that require resynchronization. There is a significant influence on the number of hard disk accesses as expected. Nonetheless, an upper bound of about 50 accesses can be given with 99% confidence for the given parameter set. This leads to the conclusion that the server system will face a maximum of 50,000 I/O operations per second in the very rare case that all nodes are in resynchronization. Real values are expected at about 10% and therefore 15,000 I/O operations, a value that can easily be handled by enterprise storage systems. Hence, a central authentication instance can be implemented for the given scenario and the scalability criterion is fulfilled.

Another valid concern is the added network overhead by AIMS4MSD. In order to evaluate the amount of generated traffic between reader and TTP, an IPsec Transport Mode communication utilizing Advanced Encryption Standard (AES) in Galois/Counter Mode (GCM) [27], a cryptographic hash function with 16 byte output, a 16 byte nonce generator, and 8 byte time stamps are assumed. This adds a maximal communication overhead of 57 Bytes to every packet as it requires a header (20 Bytes for Internet protocol version 4), an encrypted security payload (ESP) header (8 Bytes), an initialization vector (8 Bytes for AES-GCM), an ESP padding and trailer (maximum 5 Bytes for AES-GCM), and authentication data (maximum 16 bytes for AES-GCM).

For the communication during authentication phase this leads to an outbound communication of 115 bytes per authentication as packet A3 consists of 57 bytes packet overhead, two nonces (32 bytes), one hash (16 bytes), and a pseudo ID of 10 bytes. The inbound communication requires 175 bytes for packet A4, which consists of two nonces (32 bytes), $K_{R,T}$ (16 bytes), a pseudo ID (10 bytes), a KA ID (4 bytes), a time-stamp (8 bytes), and a ticket (48 bytes). This ticket, which is assumed to be padded to full AES block-sizes, contains a reader ID (4 bytes), a tag ID (8 bytes), KA ID (4 bytes), $K_{KA,T}$ (16 bytes), and a time-stamp (8 bytes).

During the data access phase two packets must be transmitted over the network. For packet DA3 171 bytes are required that are 57 bytes communication overhead, a pseudo ID (10 bytes), a nonce (16 bytes), a request vector (implementation depended, but no more than 32 bytes for 128 fields), a time-stamp (8 bytes), and the encrypted ticket (48 bytes). The reply DA4 has a size of 257 bytes, consisting of the 57 bytes general overhead, a pseudo ID (10 bytes), a nonce (16 bytes),

a response vector (32 bytes), keys (e.g., access to four fields $4 \cdot 2 \cdot 16$ bytes = 128 bytes), and a hash (16 bytes).

This leads to a total of 286 bytes outbound and 432 bytes inbound communication per tag with regards to the reader. Already with 1 megabit per second communication capacity a total of 289 tag authentications and authorizations per second is possible. Therefore, the resulting communication overhead should not pose a problem, as this amount of traffic would be comparable to other traffic generated by data warehouse applications as found in logistic systems.

VII. Conclusion and Future Work

In difference to common RFID security approaches, AIMS4MSD can effectively handle a large number of different stakeholders and deploy role based access control to data on large amounts of tags in modern supply chains. The used data-on-tag approach allows for a graceful degradation in the case of server compromises and gives customers the assurance that data can only be accessed by authorized systems in the vicinity of their tags. Both will lead to a greater rate of acceptance. While the identified anonymity requirements led to the decision for a single logic authentication server, it can be replicated onto multiple physical entities and the proposed anonymization technique allows for an effective trade-off between server load and traceability of the tags. Furthermore, the performed simulative study of the authentication server showed that this critical part of the system can be implemented on current enterprise storage systems.

For future work, the development of different approaches for resilience improvement are planned. One imaginable way to accomplish a partition of the authentication server would introduce cryptographic watermarks in the pseudo IDs that identify the correct server and are only readable by legitimate readers. Alternatively, two different authentication servers could organize two hash tables, one keeping a large, slowly changing dataset of already deployed tags, and the other one keeping only a small set of data for tags that are currently in the supply chain. Another approach could delay the activation of the anonymization mechanism until the final deployment to the customers as anonymization might not be an explicit objective within the supply chain itself, and therefore reduce authentication server load and relax timing constrains. However, competing companies could then easily reveal supply chain associations. A further line of research could analyze the applicability of rights management technologies from the emerging MPEG-21 standard [28] for specifying access rights on key and authorization servers. Finally, active sensors in the products could store measurement data on the RFID tags and further exploit the data-on-tag storage. These sensors would have to be integrated into the access control scheme.

References

[1] R. Das, "IDTechEx RFID Market projections 2008 to 2018," 2008. [Online]. Available: http://www.idtechex.com/products/en/articles/00000813.asp

[2] S. Chalasani and R. Boppana, "Data Architectures for RFID Transactions," *Industrial Informatics, IEEE Transactions on*, vol. 3, no. 3, pp. 246–257, Aug. 2007.

[3] H.-B. Jun, D. Kiritsis, and P. Xirouchakis, "Product Life-Cycle Metadata Modeling and Its Application with RDF," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, no. 12, pp. 1680–1693, Dec. 2007.

[4] International Organization for Standardization, Geneva, Switzerland, "ISO/IEC Standard 14443," Published, 2001.

[5] D. W. Engels and S. E. Sarma, "Standardization Requirements within the RFID Class Structure Framework," Auto-ID Labs, Massachusetts Institute of Technology, Cambridge, MA USA, Tech. Rep., 2005.

[6] M. McLoone and M. Robshaw, "Public Key Cryptography and RFID Tags," in *Topics in Cryptology – CT–RSA 2007*, 2007, pp. 372–384.

[7] H.-J. Korber, H. Wattar, and G. Scholl, "Modular Wireless Real-Time Sensor/Actuator Network for Factory Automation Applications," *Industrial Informatics, IEEE Transactions on*, vol. 3, no. 2, pp. 111–119, May 2007.

[8] M. Strasser, *RFID im Supply Chain Managment*. Deutscher Universitäts-Verlag, 2005.

[9] Kraftfahrt-Bundesamt, "Statistische Mitteilungen – Fahrzeugzulassungen," 2007. [Online]. Available: http://www.kbashop.de/wcsstore/KBA/Attachment/Kostenlose_Produkte/b_hersteller

[10] M. R. Rieback, B. Crispo, and A. S. Tanenbaum, "Is your cat infected with a computer virus?" *Pervasive Computing and Communications*, pp. 169–179, 2006.

[11] D. Ferraiolo and R. Kuhn, "Role-based access controls," in *15th NIST-NCSC National Computer Security Conference*, 1992, pp. 554–563.

[12] J. O. Lauf and H. Sauff, "Secure Lightweight Tunnel for Monitoring Transport Containers," in *3rd International Conference on Security and Privacy in Communication Networks – SecureComm 2007*, IEEE. Nice, France: IEEE Computer Society Press, September 2007.

[13] D. S. Kim, T.-H. Shin, and J. S. Park, "A Security Framework in RFID Multi-domain System," in *ARES '07: Proceedings of the The Second International Conference on Availability, Reliability and Security*. IEEE Computer Society, 2007, pp. 1227–1234.

[14] EPCglobal Inc., "The EPCglobal Network: Overview of Design, Benefits, & Security," Tech. Rep., 2004.

[15] J. G. Steiner, B. C. Neuman, and J. I. Schiller, "Kerberos: An authentication service for open network systems," in *Proceedings of the USENIX Winter 1988 Technical Conference*. Berkeley, CA: USENIX Association, 1988, pp. 191–202.

[16] N. Z. Luigi Lo Iacono, Christoph Ruland, "Secure transfer of measurement data in open systems," *Computer Standards & Interfaces: Validation of Software in Metrology*, vol. 28, no. 3, pp. 311–326, 2006.

[17] A. Treytl, T. Sauter, and G. Bumille, "Real-time energy management over power-lines and internet," in *International Symposium on Power-Line Communications and its Applications*, 2004, pp. 306 – 311.

[18] B. A. Khan, J. Mad, and A. Treytl, "Security in agent-based automation systems," in *IEEE Conference on Emerging Technologies & Factory Automation*, 2007, pp. 768–771.

[19] A. Juels, "Minimalist cryptography for low-cost RFID tags," in *International Conference on Security in Communication Networks – SCN 2004*, ser. Lecture Notes in Computer Science, C. Blundo and S. Cimato, Eds., vol. 3352. Amalfi, Italia: Springer-Verlag, September 2004, pp. 149–164.

[20] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic approach to "privacy-friendly" tags," in *RFID Privacy Workshop*, MIT, MA, USA, November 2003.

[21] G. Avoine and P. Oechslin, "A scalable and provably secure hash based RFID protocol," in *International Workshop on Pervasive Computing and Communication Security – PerSec 2005*, IEEE. Kauai Island, Hawaii, USA: IEEE Computer Society Press, March 2005, pp. 110–114.

[22] G. Tsudik, "YA-TRAP: Yet another trivial RFID authentication protocol," in *International Conference on Pervasive Computing and Communications – PerCom 2006*, IEEE. Pisa, Italy: IEEE Computer Society Press, March 2006.

[23] Y. Seo and K. Kim, "Scalable and untraceable authentication protocol for RFID," in *International Workshop on Security in Ubiquitous Computing Systems – Secubiq 2006*, ser. Lecture Notes in Computer Science. Seoul, Korea: Springer-Verlag, August 2006.

[24] G. M. Koien, "An introduction to access security in UMTS," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 8–18, Feb 2004.

[25] L. Gong, R. Needham, and R. Yahalom, "Reasoning About Belief in Cryptographic Protocols," in *Proceedings of the IEEE 1990 Symposium on Security and Privacy*, 1990, pp. 234–248.

[26] M. Henseler, "Credential Management für heterogene RFID-Lösungen in Logistik-Anwendungen (Credential Management for hetrogenous RFID Solutions in Logistic Appliances)," Master's thesis, Technische

Universität Ilmenau, Germany, February 2008. [Online]. Available: http://www.tu-ilmenau.de/fakia/Diplomarbeiten.676.0.html

[27] J. Viega and D. McGrew, "The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)," RFC 4106 (Proposed Standard), 2005. [Online]. Available: http://www.ietf.org/rfc/rfc4106.txt

[28] I. Burnett, R. V. de Walle, K. Hill, J. Bormans, and F. Pereira, "MPEG-21: Goals and Achievements," *IEEE MultiMedia*, vol. 10, no. 4, pp. 60–70, 2003.

Marcel Henseler graduated in Computer Science at Technische Universität Ilmenau, Germany, in 2008. He is currently working as systems engineer for Controlware GmbH, where his focus lies on IT security.

Michael Roßberg graduated in Computer Science at Technische Universität Ilmenau, Germany, in 2007, where he is currently working at the Institute of Practical Computer Science. His research interest lies in the area of network security and protection of communication infrastructures.

Günter Schäfer Guenter Schaefer received his diploma and Ph.D in computer science from the University of Karlsruhe, Germany in 1994 and 1998, respectively. Between February 1999 and July 2000 he took a post at the Ecole Nationale Superieure des Telecommunications in Paris, France, where he focused on network security and access network performance of third-generation mobile communication networks. Between August 2000 and March 2005 worked at the Technical University of Berlin, Germany in the areas of network security and advanced mobile communication architectures and services. Since April 2005, he is a full professor of telecommunications/computer networking at the University of Ilmenau, Germany. His main subject areas are network security, protection of communication infrastructures, as well as communication protocols and architectures. He is a member of the Association for Computing Machinery (ACM), the Institute of Electrical and Electronics Engineers (IEEE) and the German Gesellschaft fuer Informatik (Computer Science Society).