AnSta: Anonymous Statistics using RFID tags

Erik-Oliver Blass Kaoutar Elkhiyaoui Refik Molva
EURECOM, Sophia Antipolis, France

ABSTRACT

Current work in RFID security focuses mainly on authentication and privacy preserving identification. In this paper, we discuss the possibility of widening the scope of RFID security by introducing a new application scenario. The application we propose aims at collecting statistics on some attributes. The main requirement is to perform this operation without violating the privacy of the holders of tags. In order to do so, we combine homomorphic encryption and aggregation at the readers to ensure the privacy of the data stored on tags and re-encryption technique to prevent tracking. AnSta is the scheme we propose to implement such an application. In AnSta, RFID tags store an encrypted form of the values of the targeted attributes. The readers scan tags and forward the aggregate of their encrypted readings to the back-end server. The back-end server then decrypts the aggregates it receives and updates the global statistics accordingly. AnSta is provably privacy-preserving. Moreover, tags can be very simple, they are not required to perform any kind of computation, but only to assure the storage of a few short messages.

1. INTRODUCTION

In Radio Frequency IDentification (RFID), tags are transponders that reply to reader queries and send their identifiers, they are mainly used for identification of goods and even individuals in some cases. RFID tags are very cost effective which makes them appealing for large scale deployment, however, such deployment comes with new security and privacy threats such as cloning, impersonation, tracking, etc. The cost effectiveness on the other hand comes with strong limitation of computational capabilities of the tags. Current passive RFID tags can hardly afford for security mechanisms relying on complex cryptographic operations to counter the security and privacy threats.

Revisiting security problems such as authentication and privacy preserving identification in the highly constrained setting of RFID tags has given rise to a large number of research activities, focusing on lightweight authentication, identification schemes, and formal security and privacy properties thereof, e.g., see Bringer and Chabanne [2], Bringer et al. [3], Dimitrou [7], Pietro and Molva [13], Tsudik [14], Vaudenay [15], Weis et al. [16]. As a result, the basic authentication and identification problems with RFID tags hardly offers any security or privacy problems that have not been tackled by a number of researchers.

In an attempt to explore new security and privacy problems with RFID tags, we introduce a new application scenario, raising new requirements beyond the classical authentication and identification issues. The target scenario is the collection of statistics over private attributes called *properties* of a large population of individuals, while preserving the privacy of these individuals with respect to their attributes.

Addressing this scenario with RFID tags, each tag would contain the attributes of its holder in an encrypted form. The ultimate goal would be to allow a centralized party, such as a server, to compute global statistics: for example, the distribution over the properties held by a group of individuals without disclosing the attributes of individuals to any party involved in the collection of these statistics.

Hence, we suggest a scheme called *AnSta* that assures privacy of individual attributes in this scenario. In AnSta, intermediate parties called *readers* collect encrypted properties from *tags*, compute aggregates over encrypted readings without decrypting them, and periodically forward the result of such aggregation operation to the *back-end server*. The server is then able to compute a global aggregate in cleartext based on the aggregates of the encrypted readings transmitted by readers.

The main challenge in this scenario is to allow the readers to perform the aggregation over encrypted attribute values from which the server can derive global statistics in cleartext. We address this problem through homomorphic encryption. Another threat to the privacy of the tag holders is the tracing of tags by readers. In order to circumvent this threat, we use re-encryption mechanisms. Moreover, the scarcity of resources in tags prohibits the assignment of complex operations to tags, let alone encryption operations. In AnSta, tag thus do not have to perform any complex operations such as encryption and hashing.

In conclusion, the **major contributions** of AnSta are:

- AnSta provides an RFID-based mechanism to enable privacy-preserving statistics over a set of properties.
- We formally prove AnSta's *privacy* and *unlinkability* against external eavesdroppers, malicious readers, and curious back-end servers.

- AnSta does not require tags to do any computation, let alone complex cryptography. Instead, tags are simple, can be passive, i.e., battery-less, and only feature read/write memory. Similar to standard EPC Class 1 Generation 2 tags, this results in low production costs.

The sequel of this paper is organized as follows. In Section 2, we present a typical scenario for our application, we state the problem, and we derive the requirements for the solution. In Section 3, we present the building blocks of AnSta. In Section 4, we define the notion of privacy and unlinkability in the context of our application, and we present the adversary model. Section 5 gives the formal analysis of the protocol. Finally, related work is presented in Section 6, and we conclude.

2. PROBLEM STATEMENT

In this section, we introduce a typical scenario for AnSta, and we present a system model and the requirements AnSta should fulfill.

2.1 Application scenario

The solution we propose targets applications involving a central organization that wants to collect statistics on a given population. This population will be equipped with RFID tags that will be read by readers managed by intermediary entities independent from the central organization.

We can imagine a scenario where the ministry of culture wants to come up with statistics about the attendance of cultural events in order to determine properly her funding policy. To that effect, the ministry will deploy readers at the entry of venues where cultural events take place such as cinemas, theaters, museums, etc. In this scenario, each potential attendee of cultural events will be equipped with an RFID tag that encodes the private attributes of the tag's holder, for example gender, age, profession etc. When the holder of a tag enters the venue of a cultural event, the encrypted attributes on the tag will be scanned by a reader. Each readers will aggregate the encrypted data it collects during a period such as a day. At the end of each period each reader will forward the aggregate data to a server managed by the ministry of culture. The server will then update the overall statistics based on the aggregate values sent by all the readers.

The *key requirement* in this scenario is thus to allow the server to compute global statistics over private attributes of visitors while assuring the privacy of individual attribute values with respect to the readers and the server.

2.2 System model

As in the scenario above, the solution we propose involves the following entities:

- Issuer I: The issuer initializes each tag by writing into the tag's memory an encrypted representation of the properties of the tag holder.

- Tags T_i: Each tag stores an *encryption* of the properties of the tag holder. The tag contains a total of p properties $P_i, 1 \leq i \leq p$. Each property P_i is set to "true" when the tag holder satisfies the property. The properties represent the values of the attributes of the tag holder. For instance, the attribute gender could be represented by two properties, $(P_1, P_2) = (male, female)$. When the tag holder is male, P_1 will be set to true, and P_2 will be set to false. We can basically represent any attributes, e.g., non-boolean ones like age (with arbitrary granularity). The properties in such a case could be (P_1, P_2, P_3, P_4) = (under 12, over 12, under 25, over 25). When the tag holder is of age 18 years, then (P_1, P_2, P_3, P_4) = (false, true, true, false) etc.

- Readers R_i: Readers are in charge of collecting properties stored on tags. They read and forward the result of these readings to the back-end server.

- A back-end server S: S processes the aggregate data received from readers and then derives some global statistics such as distribution of attendance rate with respect to event types and population characteristics.

Generally, the issuer and the back-end server can be managed by independent parties. Health care agencies could act as issuer, whereas the back-end server would be managed by the ministry of culture.

2.3 Requirements: Privacy & Unlinkability

The basic requirement for S is to count the number of tag holders satisfying any property P_i. The main concern is to gather statistics such as counts about each property P_i, while preserving the *privacy* of tag holders. Neither readers nor the back-end server should be able to reveal any individual's properties. To ensure privacy in our scheme, we propose a solution that combines encryption and aggregation. A list ω_i of properties of each tag i will be encrypted to $E(\omega_i)$ and stored on the tags. A reader computes the aggregate of the ciphertexts received from the tags in its range, $\sum E(\omega_i)$, and periodically forwards the encrypted aggregate to the back-end server as shown in Figure 1.

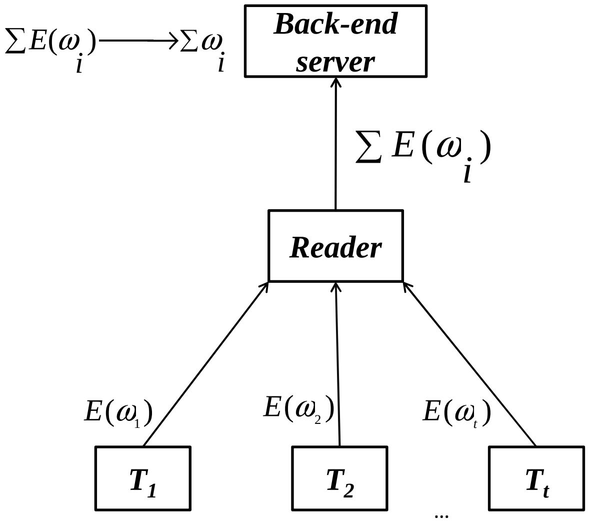

Figure 1: *Aggregation in an RFID-based system*

The back-end server S is the only entity that can decrypt ciphertexts. To enforce privacy against S, readers must aggregate the ciphertexts received from the tags in their range before forwarding the encrypted data to S. If the readers forward data without aggregation to the back-end server S, the latter can always tell which properties the tag holders satisfy. Nonetheless, forwarding each individual reading to the server would strongly overload typically embedded, low capacity readers.

Even though the privacy of properties is assured through encryption, *unlinkability* of tags, see Chatmon et al. [5], has to be assured, too. An adversary should never be able to link two different tags and therewith individuals to each other. In order to assure unlinkability, the encrypted property values sent by the same tag should be different for each reading. Re-encryption is used to that effect.

3. ANSTA

Plain encryption of the properties of the tag holders ensures privacy of the data sent to readers. However, it does not allow aggregation without decryption. If readers decipher the data sent by the tag every time, the privacy of the tag holder against readers cannot be ensured. Therefore, a homomorphic encryption is used in order to allow aggregation without decryption. Homomorphic encryption allows the back-end server to derive the value $\sum \omega_i$ in cleartext from the aggregate of encrypted values $\sum E(\omega_i)$.

Even though the privacy of properties is met through homomorphic encryption and aggregation of encrypted readings, these two mechanisms do not ensure the unlinkability of tags. Unlinkability of tags is required in order to prevent the readers or eavesdropping adversaries from tracking tags over different sessions. A basic solution for unlinkability can be provided by re-encryption, cf., Golle et al. [11]. Re-encryption cannot be performed on tags, as they are completely passive. Therefore, re-encryption will be performed by readers. The readers however should not be able to decrypt the ciphertexts they receive, otherwise, they can always learn the properties a tag satisfies. To tackle this problem, we use an asymmetric encryption that is homomorphic.

As a well studied homomorphic asymmetric encryption scheme, Elgamal [9] meets the requirements of our application, and we use it as the underlying technique. In addition to its homomorphism, Elgamal supports re-encryption. The target scenario for our application calls for an additive homomorphism. However, Elgamal is multiplicatively homomorphic and thus falls short of suiting the target application. The last component of the solution we propose is a special property encoding technique based on *Gödel encoding* [10].

3.1 Elgamal Cryptosystem

- **Setup:** The system outputs two large prime P and Q such that Q divides $(P-1)$ and $|P| = \tau$. Here, τ represents the security parameter of Elgamal. Let $\mathcal{G}$ be the subgroup of $\mathbb{Z}_P^*$ of order Q, and g be a generator of $\mathcal{G}$. All arithmetic operations will be performed mod P.
- **Key generation:** The secret key sk is $x \in \mathbb{Z}_Q$. The corresponding public key pk is $y = g^x$.
- **Encryption:** To encrypt a message $m \in \mathcal{G}$, one randomly selects $r \in \mathbb{Z}_Q$ and computes $(u, v) = (g^r, y^r m)$. The ciphertext is $c = (u, v)$.
- **Decryption:** To decrypt a ciphertext $c = (u, v)$, one computes $m = \frac{v}{u^x}$.

Elgamal encryption is multiplicatively homomorphic:

$$\forall m_1, m_2 \in \mathcal{G}, E(m_1) \cdot E(m_2) = E(m_1 \cdot m_2)$$

To adapt Elgamal to our scheme, we encode the properties using Gödel encoding before encryption as follows.

3.2 Property Encoding

In order to collect statistics on p properties P_i, we assign to each property a prime number p_i. Without loss of generality the first prime number p_1 will correspond to the property P_1, the second prime number p_2 will correspond to P_2 and so on. Both, properties P_i and primes p_i are publicly known. If the holder of a tag satisfies two properties P_i, P_j this will be represented by $(p_i \cdot p_j)$. More formally:

- **Setup:** Let P_i, $1 \leq i \leq p$, be the p properties the back-end server is interested in, and p_i are p primes. Each property P_i will be mapped to prime number p_i.
- **Encoding:** Let m be the vector $(\nu_1, ..., \nu_p)$ such that $\nu_i = 1$, if the tag T fulfills the property P_i, otherwise $\nu_i = 0$. The encoding of the properties of the tag T is defined as $\Omega(m) = \prod_{i=1}^{p} p_i^{\nu_i}$.

3.3 Protocol

In AnSta, the tags are initialized once by the issuer. Whenever a tag T is read by a reader R, the reader aggregates the ciphertext $c = (u, v)$ it receives from T, then it re-encrypts the ciphertext c and writes the new ciphertext into T. Periodically, readers in the system forward their aggregates to the back-end server. The latter decrypts and decodes the aggregates and computes the statistics it is interested in.

We assume that the system comprises, for ease of understanding, a single reader, and it has t tags in its range.

- **System setup:** Let $\mathcal{G}$ be a group in which the discrete logarithm is intractable, g a generator of $\mathcal{G}$, Q the order of $\mathcal{G}$ and P_i the p properties of the system. The output of the setup operation is a pair of keys (pk, sk): $(y = g^x, x)$, $x \in \mathbb{Z}_Q$, and p primes p_i such that the property P_i corresponds to prime number p_i. Elgamal secret key $sk = x$ is known by both the issuer and the back-end server. Generator g, the public key $pk = y$ and the p primes are made public.
- **Tag initialization:** The input comprises the vector $m = (\nu_1, ..., \nu_p)$, the public key y, the secret key x, the p primes p_i, and a random number $r \in \mathbb{Z}_Q$. The issuer of the tag encodes the vector m following the Gödel encoding and computes $\omega = \Omega(m)$. The output of the initialization operation is a ciphertext $(u, v) = (g^r, y^r \omega)$, cf., Figure (2).
- **Aggregation:** The input is a set of t ciphertexts (u_i, v_i), $1 \leq i \leq t$, received by the reader from the tags in its range. The reader outputs the *aggregate*, a new ciphertext $(U, V) = \prod_{i=1}^{p} (u_i, v_i)$ where multiplication is performed component wise, cf., Figure 3.
- **Re-encryption:** The input of re-encryption is a ciphertext $(u, v) = (g^r, y^r \omega)$ received by the reader from a tag T, g the generator of $\mathcal{G}$, the public key y, and

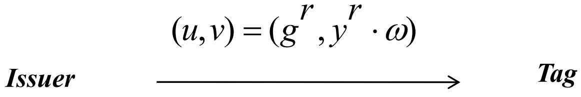

Figure 2: *Issuer initializes a tag*

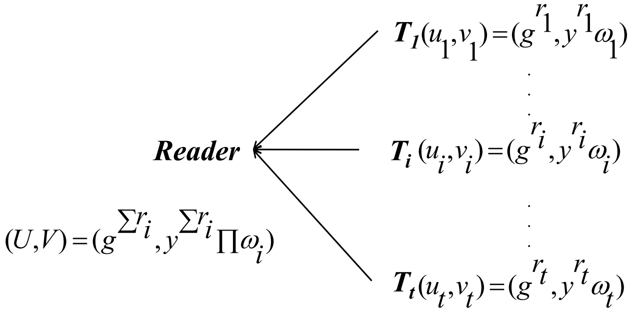
$T_1(u_1,v_1)=(g^{r_1},y^{r_1}\omega_1)$
Reader
$T_i\,(u_i,v_i)=(g^{r_i},y^{r_i}\omega_i)$
$(U,V)=(g^{\sum r_i},y^{\sum r_i}\prod\omega_i)$
$T_t(u_t,v_t)=(g^{r_t},y^{r_t}\omega_t)$

Figure 3: *Readers aggregate ciphertexts from different tags*

a random number $r' \in \mathbb{Z}_Q$. The output is a new ciphertext $(u', v') = (g^{(r+r')}, y^{(r+r')}\omega)$, cf., Figure 4.

- **Decryption and decoding:** The input is an aggregated ciphertext $(U, V) = \prod_{i=1}^{t}(u_i, v_i)$ received from the reader, the secret key x, and the p primes p_i. The back-end server computes $W = \frac{V}{U^x}$ and factorizes W. This factorization is easily feasible, as the back-end server knows the primes p_i. Given that this factorization is unique, the back-end server gets $\Omega^{-1}(W) = (\nu_1, ..., \nu_p)$. The respective ν_i corresponds to the number of tags satisfying the property P_i that have been read by the reader.

To get the total number of tags satisfying a property P_i in the case of multiple readers, the back-end server sums the ν_i for all the readers in the system.

Aggregation under restrictions: In order to ensure the correctness of statistics obtained by the back-end server, we cannot allow the readers to aggregate an infinite number of ciphertexts. They are only allowed to aggregate up to a threshold γ of ciphertexts $c_i = E(\omega_i)$ at a time, such that $\prod_{i=1}^{\gamma}\omega_i < P$. Typically, $|P| = 1024$ bits and $|Q| = 160$ bits.

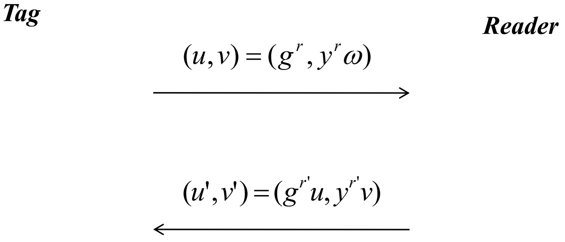

Figure 4: *Readers re-encrypt ciphertexts received from tags*

Let us assume that tags store the following properties: $(P_1,\ P_2,\ P_3,\ P_4,\ P_5,\ P_6)$ = (student, employee, male, female, over 25, under 25). Let $(2, 3, 5, 7, 11, 13)$ be the primes corresponding to these properties. Given that a tag holder cannot be male and female or under 25 and over 25 at the same time, the encoding of properties is upper-bounded by 546, this means following the previous notations, $\forall m = (\nu_1, ..., \nu_6)$, $\Omega(m) \leq 546$. If $|P| = 1024$, we can aggregate up to $\gamma = \frac{1024}{\log_2(546)} = 112$ tags at once.

Generally, given p properties P_i and p prime numbers p_i, the threshold γ could be defined as $\frac{\frac{|P|}{p}}{\log_2(\prod_{i=1}^{p} p_i)}$. If a reader has σ tags in its range, it will aggregate ciphertexts by bunches of size at most γ. The reader then instead of forwarding one aggregate it forwards $\lfloor\frac{\sigma}{\gamma}\rfloor + 1$ aggregates to the back-end server.

4. ADVERSARY & PRIVACY MODELS

In this section, we introduce the adversary model and define the notions of privacy and unlinkability for the proposed application.

4.1 Adversary model

AnSta protects against two different categories of adversaries,

1. $\mathcal{ADV}_1$, external adversaries and malicious readers,
2. $\mathcal{ADV}_2$, a malicious back-end server.

$\mathcal{ADV}_1$ does not collude with $\mathcal{ADV}_2$.

4.1.1 $\mathcal{ADV}_1$

Borrowing notions from Cramer and Damgård [6], we assume a *rushing, active* adversary who has full control over all communication between tags and readers. He can not only eavesdrop messages, but also intercept, modify, and even initiate communication. For example, the adversary might impersonate a tag and communicate with the reader or read-out tags. He might even replace a tag's content by re-writing it. However, re-writing tags has some special implications on AnSta's security and privacy, so we discuss this issue separately in Section 4.3 and only assume read-access to tags in the sequel of this section. Finally, the adversary might compromise readers, read-out and tamper with their memory and program – consequently, malicious readers might not behave in protocol compliant manner.

4.1.2 $\mathcal{ADV}_2$

The back-end server might be under the control of the adversary, e.g., as assumed if the organization collecting the statistics is generally not trusted. The back-end server is not assumed to have full control over the network. The back-end server is passive in the sense that it only receives aggregates from readers. It cannot initiate communication with tags or readers.

We conjecture that there might be scenarios where back-end servers have full control over all communication and might collude with compromised readers, e.g., envisioning an extreme scenario whereby the ministry of culture would also own the readers of all the cultural venues. We clearly state that AnSta will not provide privacy in such scenarios.

As motivated in the introduction, the adversary's primary goal in any case, i.e., $\mathcal{ADV}_1$ or $\mathcal{ADV}_2$, is to gain some knowledge about sensitive information, in this case individual tag holders' properties as formalized in the following privacy models.

Note that AnSta primarily focuses on the tag holders' privacy, and security properties of statistics, such as the integrity of the final outcome of statistics is *not* a key issue. Nevertheless, we will discuss security aspects briefly in Section 4.3.

4.2 Privacy Models

At the end of the protocol execution, AnSta is said to be privacy preserving, if readers, external adversaries, and the back-end server

- cannot decide which properties a given tag (and therewith tag holders) satisfies.
- cannot link tags (and therewith tag holders) to previous protocol executions.

We use experiment-based definitions to formalize RFID privacy, cf., Juels and Weis [12]. In conclusion, the adversary should not have higher chance in breaking privacy or unlinkability than simple guessing. The following oracle-like constructions exist:

$\mathcal{O}_{\text{pick}}$ is an oracle that randomly selects some tags from all the n tags in the system.

$\mathcal{O}_{\text{semantic}}$ is provided with two plaintexts ω_0, ω_1, randomly chooses $b \in \{0,1\}$, encrypts ω_b using Elgamal and public key pk, and returns the resulting ciphertext c_b.

$\mathcal{O}_{\text{reencrypt}}$ is an Elgamal re-encryption oracle that uses public key pk and ciphertext $c = (u, v)$ stored on tag T, and writes a new (re-encrypted) ciphertext $c' = (u', v')$ into T, cf., Section 3.3.

$\mathcal{O}_{\text{flip}}$ is an oracle that, provided with two tags T_0, T_1, randomly chooses $b \in \{0,1\}$ and re-encrypts the ciphertext stored on T_b using $\mathcal{O}_{\text{reencrypt}}$. It returns T_b with the re-encrypted ciphertext.

$\mathcal{O}_{\text{aggregate}}$ computes a total of s aggregates Agg_1, Agg_2, …, Agg_s, each time by randomly choosing a set of γ tags, as follows: Agg_1 is computed using tags $(T_1^1, T_1^2, \ldots, T_1^\gamma)$, Agg_2 is computed using $(T_2^1, T_2^2, \ldots, T_2^\gamma)$, …, Agg_s is computed using $(T_s^1, T_s^2, \ldots, T_s^\gamma)$. The sets of tags are chosen randomly, but there is at least one tag that is an element of two different sets, i.e., used in the computation of two different aggregates. Finally, $\mathcal{O}_{\text{aggregate}}$ returns Agg_1, Agg_2, …, Agg_s.

4.2.1 Privacy against $\mathcal{ADV}_1$

An adversary breaks the privacy of AnSta, if given the public key pk, a tag T, the ciphertext $c = (u, v)$ stored on the tag T, and a property P_i, he can decide if a tag T satisfies the property P_i.

More formally, for τ the security parameter of Elgamal and $s \in \mathbb{N}$, we define the following privacy experiment:

Experiment $\text{Exp}_{\mathcal{ADV}_1}^{\text{privacy}}[\tau, s]$

1. **Setup:** The issuer initializes n tags with their corresponding ciphertexts using Gödel encoding and Elgamal. It publishes the public key pk. It shares its secret key sk only with the back-end server.
2. **Learning:** $\mathcal{O}_{\text{pick}}$ provides the adversary $\mathcal{ADV}_1$ with a single challenge tag T that he reads for a maximum of s times. After each read, T is given to $\mathcal{O}_{\text{reencrypt}}$ that re-encrypts T's stored ciphertext.
3. **Guess:** $\mathcal{ADV}_1$ selects a property P_i. Given the public key pk and the s intermediate states of tag T, $\mathcal{ADV}_1$ outputs 1, if he guesses that T satisfies P_i, and 0 otherwise. $\mathcal{ADV}_1$ is *successful*, if his guess is right.

Definition 1. *Let $\tau \in \mathbb{N}$ be a security parameter. We consider as negligible in τ any function $\mu : \mathbb{N} \rightarrow [0,1]$ such that $\forall c > 0, \mu(\tau) < \frac{1}{\tau^c}$ for every sufficiently large τ.*

Definition 2. *AnSta is said to be privacy preserving with respect to $\mathcal{ADV}_1$:*

if for all adversaries of category $\mathcal{ADV}_1$,

$$\Pr[\text{Exp}_{\mathcal{ADV}_1}^{\text{privacy}}[\tau, s] \text{ succeeds}] \leq \frac{1}{2} + \mu(\tau),$$

such that $\mu(\tau)$ is a function negligible in τ.

4.2.2 Privacy against $\mathcal{ADV}_2$

Formalizing properties' privacy with respect to $\mathcal{ADV}_2$ is difficult: as assumed in the adversary model of Section 4, $\mathcal{ADV}_2$, i.e., a malicious back-end server, only receives aggregates from readers. In any case, there is no relation between tags, and therewith tag holders, and $\mathcal{ADV}_2$. In conclusion, $\mathcal{ADV}_2$ simply cannot learn anything about properties of tags.

While we do not target a formal proof, privacy against $\mathcal{ADV}_2$ is furthermore discussed and additional reasoning is given in the according security analysis section 5.1.2.

4.2.3 Unlinkability against $\mathcal{ADV}_1$

Neither external adversaries nor readers should be able to link two responses from the same tag once it is re-encrypted outside the range of the adversary.

Experiment $\text{Exp}_{\mathcal{ADV}_1}^{\text{unlinkability}}[\tau, s]$

1. **Setup:** the issuer initializes n tags with their corresponding ciphertexts using Gödel encoding and Elgamal cryptosystem, it publishes its public key pk. It shares its secret key sk only with the back-end server.
2. **Learning:** The oracle $\mathcal{O}_{\text{pick}}$ provides the adversary $\mathcal{ADV}_1$ with two randomly chosen tags T_0, T_1. The adversary is allowed to read tags T_0, T_1 and also any other tag out of all tags in the system for a maximum of s times. However, after each reading of any tag T, T is given to $\mathcal{O}_{\text{reencrypt}}$ to re-encrypt its ciphertext.
3. **Guess:** Using T_0 and T_1, $\mathcal{O}_{\text{flip}}$ provides $\mathcal{ADV}_1$ with a re-encrypted T_b. Given the public key pk, the results of s reads, and the current ciphertext stored on the tag T_b, the adversary $\mathcal{ADV}_1$ guesses the value of b. He succeeds, if his guess is right.

Definition 3. *AnSta is said to provide unlinkability with respect to $\mathcal{ADV}_1$:*

if for all adversaries of category $\mathcal{ADV}_1$,

$$\Pr[\text{Exp}_{\mathcal{ADV}_1}^{\text{unlinkability}}[\tau, s] \text{ succeeds}] \leq \frac{1}{2} + \mu(\tau)$$

such that $\mu(\tau)$ is a function negligible in τ.

Note. In the above definition of an unlinkability experiment, we require that after each read of tag T, the ciphertext stored on T is re-encrypted by $\mathcal{O}_{\text{reencrypt}}$. In the real world, this could be achieved, if the adversary is not able to eavesdrop at least one single protocol execution between T and a legitimate reader. As tags in this paper only provide storage, i.e., a state, but no computational capability, they cannot autonomously update their state after a read. Trivially in such a setting, tags would be linkable, if the adversary can always eavesdrop all communication of T. Therefore, closely related to the notion of backward security by Dimitrou [7], we require an "unobserved" re-encryption between two reads, implemented in our experiment by $\mathcal{O}_{\text{reencrypt}}$.

4.2.4 Unlinkability against $\mathcal{ADV}_2$

A malicious back-end server should not be able to link aggregates to aggregates it has received before. More precisely, a malicious back-end server should not tell, whether a received aggregate involves a tag that was involved in another aggregate received earlier. We illustrate the unlinkability against a malicious back-end server ($\mathcal{ADV}_2$) by the following experiment:

Experiment $\text{Exp}^{\text{unlinkability}}_{\mathcal{ADV}_2}[\gamma, s]$

1. **Setup:** the issuer initializes n tags with their corresponding ciphertexts using Gödel encoding and Elgamal cryptosystem, it publishes its public key pk. It shares its secret key sk with the back-end server, i.e., $\mathcal{ADV}_2$.
2. **Learning:** $\mathcal{O}_{\text{aggregate}}$ provides $\mathcal{ADV}_2$ with s aggregates $\text{Agg}_1, \ldots, \text{Agg}_s$.
3. **Guess:** Given the public key pk, the s aggregates $\text{Agg}_1, \ldots, \text{Agg}_s$, $\mathcal{ADV}_2$ guesses a pair $b, b' \in \{1, \ldots, s\}$ and therewith Agg_b and $\text{Agg}_{b'}$. $\mathcal{ADV}_2$ succeeds, if Agg_b and $\text{Agg}_{b'}$ have been computed by $\mathcal{O}_{\text{aggregate}}$ with at least one tag in both aggregates.

Definition 4. *AnSta is said to provide unlinkability with respect to* $\mathcal{ADV}_2$*:*

if for all adversaries of category $\mathcal{ADV}_2$,

$$\Pr[\text{Exp}^{\text{unlinkability}}_{\mathcal{ADV}_2}[\gamma, s] \textit{ succeeds}]$$

$$\leq \frac{1}{s(s-1)} + \mu(\gamma)$$

such that $\mu(\gamma)$ *is a function negligible in* γ.

Untraceability. Note that in this work, we do not focus on untraceability, although this is also being considered in related work. However, the notion of untraceability is *weaker* than unlinkability, cf., Chatmon et al. [5]. Thus, if AnSta provides unlinkability, it also provides untraceability.

4.3 Malicious writing

Tags in our scheme have a writable memory, where the ciphertext is stored every time it is re-encrypted by readers. As there is no access control on tags to check the authenticity of readers, our scheme is vulnerable to "malicious writing". The adversary can write incorrect data into a tag which renders the final result obsolete.

We can divide malicious writing attacks into two categories:

- *Writing an invalid ciphertext into the tag:* this attack can be detected at the back-end server, as the decryption and the Gödel decoding will not succeed.
- *Writing a valid ciphertext into the tag:* the simplest way to implement such an attack is by copying the content of a tag into another one. Moreover, given that Elgamal is malleable and that the adversary knows the public key pk and the primes, the adversary can generate a set of valid ciphertexts from a ciphertext he has seen such that the respective plaintexts are related [8]. Since the ciphertext written into the tag is a valid one, this type of attack cannot be detected at decryption .

Malicious writing affects the correctness of the results obtained at the back-end server. Given that no access control is implemented on the tags, this attack cannot be prevented. A basic solution to ensure the integrity of the global results obtained by the back-end server, could be to detect the ciphertexts written by adversaries at the legitimate readers and discard these ciphertexts.

We can use an HMAC to check the validity of ciphertexts stored on the tags. The (legitimate) readers will share a key K that they will use to *sign* the ciphertexts. A tag will store the ciphertext $c = (u, v)$ along with the $\text{HMAC}(c)$. When a legitimate reader reads the tag, it will firstly check that $(c, \text{HMAC}(c))$ is a valid pair: if it is, the reader will aggregate the ciphertext c, re-encrypt it and compute the HMAC corresponding to the new ciphertext, otherwise, the reader discards the pair $(c, \text{HMAC}(c))$ and writes into the tag a new pair $(c_0 = E(1), \text{HMAC}(c_0))$.

This still does not protect against simple cloning attacks, so additional effort is required. One could envision, e.g., per-tag one-time serial numbers shared between readers, but this is out of scope of this paper. As mentioned above, this paper focuses on privacy. With low cost tags as in the setting of this paper, general security features such as authentication and data integrity are not relevant and cannot be afforded. Yet, even with low cost tags, the *privacy* of tag holders is the crucial issue.

5. SECURITY ANALYSIS

This section provides formal proofs for AnSta's privacy and unlinkability as defined in the models of Section 4.2.

5.1 Privacy

5.1.1 Privacy against $\mathcal{ADV}_1$

Theorem 1. *AnSta is privacy preserving with respect to* $\mathcal{ADV}_1$ *adversaries under the DDH assumption over* $\mathcal{G}$.

Proof. Assume we have an adversary $\mathcal{A} \in \mathcal{ADV}_1$ for which his advantage to break the privacy experiment is not negligible. We build a new adversary $\mathcal{A}' \in \mathcal{ADV}_1$ that uses $\mathcal{A}$ as a subroutine and breaks the semantic security of Elgamal which leads to a contradiction under the DDH assumption. In this proof, we make use of the fact that a tag T satisfies a property P_j, **iff** the corresponding prime number p_j divides the plaintext underlying the ciphertext stored on T.

- Given the public key pk, $\mathcal{A}'$ specifies two plaintexts $\omega_0 = \prod p_i^{\nu_{0,i}}$ and $\omega_1 = \prod p_i^{\nu_{1,i}}$, such that $\forall i$, $1 \leq i \leq p$, and $b' \in \{0,1\}$: $\nu_{b',i} \in \{0,1\}$ and $\nu_{0,i} + \nu_{1,i} = 1$. In terms of properties P_i, this means that the tag T_0, storing the plaintext ω_0, and the tag T_1, storing the plaintext ω_1, do not satisfy the same properties.

- $\mathcal{A}'$ transmits these two plaintexts along with the public key to the challenge oracle $\mathcal{O}_{\text{semantic}}$.
- $\mathcal{O}_{\text{semantic}}$ returns the encryption c_b of one of the plaintexts ω_0, ω_1 to the adversary $\mathcal{A}'$.
- $\mathcal{A}'$ writes c_b into a tag T.
- $\mathcal{A}'$ calls the adversary $\mathcal{A}$ and gives it T as the challenge tag. $\mathcal{A}$ picks a property P_i. As $\mathcal{A}$'s advantage in the privacy experiment is not negligible, $\mathcal{A}$ can tell, if T satisfies the property P_i or not. Therefore, $\mathcal{A}'$ can decide which plaintext ω_b corresponds to the ciphertext c_b based on the guess of $\mathcal{A}$. If T satisfies the property P_i, this implies that the corresponding prime number p_i divides ω_b. Using this information, $\mathcal{A}'$ can tell which plaintext ω_b corresponds to the ciphertext c_b – this would break the semantic security of Elgamal ensured under the DDH assumption.

  The adversary $\mathcal{A}'$ must specify ω_0 and ω_1 such that $\nu_{0,i} + \nu_{1,i} = 1$. Otherwise, using the adversary $\mathcal{A}$ as a subroutine will not suffice to break the semantic security of Elgamal, as $\mathcal{A}$ may only learn information about the properties both tags satisfy or do not satisfy.

□

5.1.2 *Privacy against $\mathcal{ADV}_2$*

As stated in Section 4.1, $\mathcal{ADV}_2$ receives only aggregated ciphertexts. Still, given the aggregates, $\mathcal{ADV}_2$ can learn some information about the properties of tags read by readers, but is never able to tell *which* tag, and therewith *which* holder satisfies *which* property.

For instance, if $\mathcal{ADV}_2$ receives an encrypted aggregate and decrypts it to $\text{Agg} = \prod_{i=1}^{p} p_i^{\nu_i}$, and $\exists j$ such that $\nu_j = 0$ after factorization, $\mathcal{ADV}_2$ can learn that all the tags that were read by R do not satisfy the property P_j.

However, as $\mathcal{ADV}_1$ and $\mathcal{ADV}_2$ do not collude, $\mathcal{ADV}_2$ cannot tell *which* tag satisfies or does not satisfy a certain property P_i.

5.2 Unlinkability

5.2.1 *Unlinkability against $\mathcal{ADV}_1$*

Theorem 2. *AnSta provides tag unlinkability against $\mathcal{ADV}_1$ under the DDH assumption over $\mathcal{G}$.*

Proof. Assume we have an adversary $\mathcal{A} \in \mathcal{ADV}_1$ for which his advantage to break the unlinkability experiment is not negligible. We build a new adversary $\mathcal{A}' \in \mathcal{ADV}_1$ that uses $\mathcal{A}$ as a subroutine and breaks the semantic security of Elgamal.

- $\mathcal{A}'$ specifies two plaintexts ω_0 and ω_1, given the public key pk, $\mathcal{A}'$ encrypts ω_0 and ω_1 and writes c_0 and c_1 into the tags T_0 and T_1 respectively.
- $\mathcal{A}'$ submits the two tags T_0 and T_1 to the adversary $\mathcal{A}$ that goes into the learning phase.
- $\mathcal{A}'$ transmits ω_0 and ω_1 along with the public key pk to the challenge oracle $\mathcal{O}_{\text{semantic}}$.
- $\mathcal{O}_{\text{semantic}}$ returns the result c'_b of encrypting one of the two plaintexts ω_0, ω_1 to the adversary $\mathcal{A}'$.
- $\mathcal{A}$ gives c'_b to $\mathcal{A}$. Given that encryption and also re-encryption generate the same distribution of ciphertexts, c'_b is also a re-encryption of c_b. Since $\mathcal{A}$'s advantage in the unlinkability experiment is not negligible, $\mathcal{A}$ can tell which tag corresponds to the new ciphertext c'_b. Therefore, $\mathcal{A}'$ can decide which plaintext ω_b corresponds to the ciphertext c'_b. This breaks the semantic security of Elgamal that is ensured under the DDH assumption, again leading to a contradiction.

□

5.2.2 *Unlinkability against $\mathcal{ADV}_2$*

Theorem 3. *AnSta provides unlinkability of tags against $\mathcal{ADV}_2$ for large γ.*

Proof Sketch. Given a sufficiently large γ, the aggregates received by the back-end server will be such that $\text{Agg} = \prod_{i=1}^{p} p_i^{\nu_i}$, and $\forall i$, $1 \le i \le p$, $\nu_i > 0$ holds with high probability. Therefore, the individual properties of the tags will be completely blinded and the back-end server cannot distinguish between the tags involved in the aggregates. Moreover, using a large s in the learning phase would not give the adversary $\mathcal{ADV}_2$ a greater advantage in guessing (b, b').

Hence, using a large γ does not allow $\mathcal{ADV}_2$ to succeed in the unlinkability experiment with an advantage over a random guess of Definition 4.

□

6. RELATED WORK

Golle et al. [11] introduce *universal re-encryption*, a new approach based on the Elgamal cryptosystem to enhance the privacy of tags. As the name implies, universal re-encryption allows re-encryption without knowing the public key initially used to encrypt the plaintexts. Similar to AnSta, tags are only required to provide read/write into memory. Re-encryption is performed by the readers in order to prevent tracking. However, this protocol fails at preventing tracing through *malicious writings*. An adversary can write into a tag a message m and encrypt it under its public key, by doing so, the adversary can always trace the tag – even after re-encryption; all he needs to do is to decipher the ciphertext stored on the tag using his secret key.

To tackle this problem, Ateniese et al. [1] propose *insubvertible encryption*, which is a universal re-encryption based on bilinear pairings and Elgamal cryptosystem. This scheme makes use of randomizable certificates based on bilinear pairings to check the honesty of the last reader (randomizer) that wrote into the tag. If the certificate is valid, the ciphertext stored on the tag will be re-encrypted. Otherwise, it will be discarded and replaced by a so called *dummy encryption.*

This paper introduces a new possible application for RFID. This application aims at securely collecting statistics over a population of RFID tags. The solution we propose uses re-encryption and secure aggregation to enforce privacy against readers and the back-end server. Unlike Ateniese et al. [1], Golle et al. [11], re-encryption in AnSta is not universal, as we assume that our setup uses one pair of public/secret key.

Camenisch and Groß [4] use Gödel encoding to encode attributes efficiently. Yet, the authors target a different problem which is proving that a credential contains an attribute a_i with a given value v_i; this involves heavy cryptography on the prover side. AnSta, on the other hand, aims at counting the number of tag holders having an attribute a_i with a value v_i – in an RFID setting.

7. CONCLUSION

RFID systems can be used for many applications besides identification and authentication. In this paper, we introduced a new application for RFID that collects statistics over a population of tag holders. We presented AnSta, a protocol to mitigate resulting new privacy problems. AnSta does not require tags to perform any (cryptographic) computation. Instead, tags only need to feature some cheap storage. All computations within AnSta are solely performed by readers. AnSta provably ensures the *privacy* of tag and therewith holders' properties as well as their unlinkability: tag holders can be sure that neither RFID readers, nor a back-end system can reveal the properties stored on their tags. Additionally, if scanned at different readers on different occasions, tag holders can be sure that these occasions cannot be linked.

References

[1] Giuseppe Ateniese, Jan Camenisch, and Breno de Medeiros. Untraceable rfid tags via insubvertible encryption. In *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*, pages 92–101, New York, NY, USA, 2005. ACM. ISBN 1-59593-226-7.

[2] Julien Bringer and Hervé Chabanne. Trusted-HB: A low-cost version of HB $^+$ secure against man-in-the-middle attacks. *IEEE Transactions on Information Theory*, 54(9):4339–4342, 2008.

[3] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. HB^{++}: a lightweight authentication protocol secure against some attacks. In *SecPerU*, pages 28–33, 2006.

[4] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In *CCS '08: Proceedings of the 15th ACM conference on Computer and communications security*, pages 345–356, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-810-7.

[5] Christy Chatmon, Tri van Le, and Mike Burmester. Secure anonymous RFID authentication protcols. Technical report, Florida State University, Department of Computer Science, Tallahassee, Florida, USA, 2006. `http://www.cs.fsu.edu/~burmeste/TR-060112.pdf`.

[6] R. Cramer and I. Damgård. Introduction to secure multi-party computations. In *Contemporary Cryptology: Advanced Courses in Mathematics*, pages 41–87. Birkhauser, 2005. ISBN 3-7643-7294-X.

[7] T. Dimitrou. rfiddot: Rfid delegation and ownership transfer made simple. In *Proceedings of International Conference on Security and privacy in Communication Networks*, Istanbul, Turkey, 2008. ISBN 978-1-60558-241-2.

[8] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-malleable cryptography. In *STOC '91: Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 542–552, New York, NY, USA, 1991. ACM. ISBN 0-89791-397-3.

[9] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc. ISBN 0-387-15658-5.

[10] Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i. *Monatsheft für Mathematik und Physik*, 38:173–198, 1931. ISSN 0026-9255.

[11] Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson. Universal re-encryption for mixnets. In *In Proceedings of the 2004 RSA Conference, Cryptographer's track*, pages 163–178. Springer-Verlag, 2002.

[12] Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. In *PERCOMW '07: Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 342–347, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2788-4.

[13] R. Di Pietro and R. Molva. Information confinement, privacy, and security in RFID systems. In *Lecture Notes in Computer Science, Volume 4734*, pages 187–202, 2007. ISBN 978-3-540-74834-2.

[14] G. Tsudik. Ya-trap: yet another trivial RFID authentication protocol. In *Proceedings of International Conference on Pervasive Computing and Communications Workshops*, Pisa, Italy, 2006. ISBN 0-7695-2520-2.

[15] Serge Vaudenay. On Privacy models for RFID. In *Advances in Cryptology - Asiacrypt 2007*, Lecture Notes in Computer Science, pages 68–87. Springer-Verlag, 2007.

[16] S.A. Weis, S.E. Sarma, R.L. Rivest, and D.W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Security in Pervasive Computing*, pages 201–212, Boppard, Germany, 2003. ISBN 3-540-20887-9.