Algebraic Attacks on RFID Protocols

Ton van Deursen* and Saša Radomirović

University of Luxembourg
{ton.vandeursen,sasa.radomirovic}@uni.lu

Abstract. This work aims to identify the algebraic problems which enable many attacks on RFID protocols. Toward this goal, three emerging types of attacks on RFID protocols, concerning authentication, untraceability, and secrecy are discussed. We demonstrate the types of attacks by exhibiting previously unpublished vulnerabilities in several protocols and referring to various other flawed protocols.

The common theme in these attacks is the fact that the algebraic properties of operators employed by the protocols are abused. While the methodology is applicable to any operator with algebraic properties, the protocols considered in this paper make use of *xor*, modular addition, and elliptic curve point addition.

Keywords: Formal verification, algebraic methods, RFID, security protocols, attacks.

1 Introduction

There are two main approaches to prove cryptographic protocols secure. The approach based on formal languages considers protocol messages on a high abstraction level and misses implementation details, but is therefore automatable. The computational approach is more accurate but also much more difficult due to the necessity of manual proofs. This work deals with algebraic verification methods which we consider to be a combination of the two mentioned approaches in the following sense. As in formal methods, we evaluate the security of protocols by considering the free term algebra generated by the messages exchanged between principals of the protocols and acted on by the standard Dolev–Yao adversary [1]. We also consider cryptographic primitives, such as hash functions and encryptions to be *perfect*. As in the computational approach, we study how much information is being leaked through terms to which operators with algebraic properties are applied. We are not aiming to prove protocols secure, but rather to understand how algebraic properties of operators and functions used in communication protocols can make these protocols fail to achieve security goals. Towards this goal, we present three emerging types of vulnerabilities discovered by analyzing recently published RFID protocols.

The investigation of algebraic properties is a particularly useful tool for the discovery of vulnerabilities in RFID protocols. The resource constraints imposed

* Ton van Deursen was supported by a grant from the Fonds National de la Recherche (Luxembourg).

O. Markowitch et al. (Eds.): WISTP 2009, LNCS 5746, pp. 38–51, 2009.

on RFID tags have led to a plethora of proposals for protocols employing *xor*, modular addition, cyclic redundancy check functions, and custom-made hash-like functions. Attempting to prove all such protocols secure in a computational security model is tedious and overkill, since a significant number of the proposed protocols turn out to be flawed. Automated tools based on formal methods approaches currently fail to verify the security of most of these protocols, because they cannot verify some of the desired security properties, such as untraceability of tags, or do not consider flaws related to partial leakage of keys. While our approach is not automatable in general, we do expect that for the types of vulnerabilities described in this paper, the automatic detection of attacks ought to become possible in the foreseeable future.

The types of attacks we present are what we call *algebraic replay attacks* targeting the challenge-response mechanism in authentication protocols in Section 3, *attribute acquisition attacks* on untraceability of tags in Section 4, and cryptanalytic attacks on secrecy of keys and tag identities in Section 5.

2 Preliminaries

2.1 Terminology, Notation, and Conventions

A *reader* refers to the actual RFID reader as well as a potential database or server communicating with the reader, since in all protocols considered this communication takes place over a secure channel. An *agent* can be a tag or a reader, while a *role* refers to the protocol steps a tag or reader is expected to carry out. A *run* is the execution of a role by an agent.

For convenience and intuition, we will refer to certain attacks on protocols as *quality-time* attacks. These are attacks in which the adversary interacts with a tag in absence of an honest or trusted RFID reader. The point of such an attack is to send carefully designed challenges to the tag in order to obtain information which can later be used to impersonate a reader or the tag, trace the tag, or attack any other security requirement of a protocol. Quality-time attacks are facilitated by the mobile and wireless nature of RFID tags. The attacks can be carried out on tags that happen to be in the vicinity of an adversary for a short period of time or on tags the attacker is able to isolate from their environment for an extended period of time.

We simplify the presented protocols whenever possible by leaving out irrelevant steps, communications, and terms. The description given suffices to reconstruct the attacks on the original protocols. When referring to the *untraceability* property of a protocol, we mean the *tag's* untraceability.

For the reader's convenience, when describing a protocol, we consistently use k for a shared secret key, h for hash functions, $r_1, \ldots, r_n$ for nonces, and ID for the tag's ID. Whenever additional functions and variables are needed we use the notation that was originally chosen by the authors of the protocol. When an attack consists of several runs, the terms used in a second run are primed.

We represent protocols graphically using message sequence charts, such as in Figure 1. Every message sequence chart shows the role names, framed, near

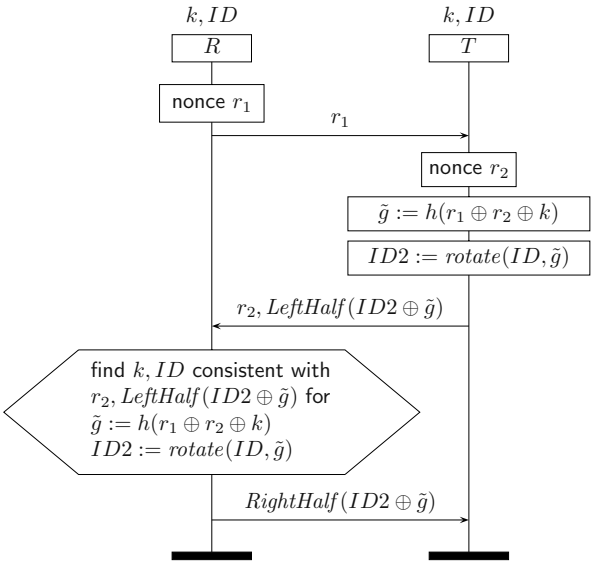

Fig. 1. Flawed authentication protocol

the top of the chart. Above the role names, the role's secret terms are shown. Actions, such as nonce generation, computation, and assignments are shown in boxes. Messages to be sent and expected to be received are specified above arrows connecting the roles. It is assumed that an agent continues the execution of its run only if it receives a message conforming to the specification. Other conditions that need to be satisfied are shown in diamond boxes. For instance, in Figure 1, the role names are R and T, both know the secret terms k and ID. R generates the nonce r_1 before sending the first message. After reception of the first message, T generates a nonce and computes the response. The reader accepts the response only if it can find a pair k, ID which produces the same term when the computation shown is applied to it. If the response is accepted, the reader continues by computing and sending the last message.

2.2 Security Properties and Adversary Models

In terms of Lowe's authentication hierarchy [2], we consider *recent aliveness* to be the most appropriate authentication requirement for RFID protocols. Recent aliveness captures the fact that the tag needs to have generated a message as a consequence of a reader's query. We consider the notion of *untraceability* as defined by Van Deursen et al. [3] which captures the intuition that a tag is untraceable if, for any two protocol runs, an adversary cannot tell whether the same tag was executing both runs or two different tags were executing the runs. Finally, terms that are not in the adversary's knowledge are said to be *secret*.

We perform our security analyses in the Dolev–Yao intruder model [1]. In this model, the adversary may eavesdrop on any message exchanged between tag and

reader, modify or block any message sent from tag to reader or vice versa, and may inject his own messages making them look like they were sent by tag or reader. The models by Avoine [4], Juels and Weis [5], Vaudenay [6], Damgård and Pedersen [7], and Paise and Vaudenay [8] extend the adversary's power with capabilities specifically tailored to the RFID setting. These capabilities will however not be necessary for the attacks presented in this paper.

3 Algebraic Replay Attacks on Authentication

A common way to authenticate RFID tags is by means of the following challenge-response mechanism. The RFID reader challenges the tag with a nonce r_1 to which the tag replies with a term derived from the nonce r_1, some information s identifying the tag, and potentially a nonce r_2 generated by the tag. If present, the nonce r_2 serves as the tag's challenge to the reader in mutual authentication protocols or as a "blinding term" to achieve tag untraceability. We can thus represent the tag's reply to the reader's challenge as the term $r_2, g(r_1, r_2, s)$ with the understanding that r_2 may be constant or empty. The reader verifies the authenticity by applying the inverse of the function g to the term and checking whether the response contains r_1 and a valid s. If g is a one-way function then the reader verifies the authenticity of the tag by computing the function $g(r_1, r_2, s)$ and comparing it to the received value. The reader can compute this function, since it generated the value r_1 itself, the value r_2 is supplied by the tag, and the reader has a database with values of s for every tag it may authenticate.

We now argue that the following two properties are necessary in order for the challenge-response mechanism to guarantee recent aliveness of the tag.

Freshness. For fixed r_2 and s the range of the function $r_1 \to g(r_1, r_2, s)$ must be large. More precisely, given r_2, s, the adversary's advantage in guessing $g(r_1, r_2, s)$ correctly for an unknown, randomly chosen r_1 must be negligible.

ARR. Let $O_s(x)$ be an oracle which upon input x randomly chooses y and returns y and $g(x, y, s)$. If s is unknown, then given access to a polynomial number of queries $O_s(x_1), \ldots, O_s(x_l)$ to the oracle, it is infeasible to compute $g(r_1, r_2, s)$ for a given $r_1 \notin \{x_1, \ldots, x_l\}$ and any r_2.

If the freshness property is satisfied, then as stated, the probability of the adversary guessing $g(r_1, r_2, s)$ is negligible. Thus with overwhelming probability, a response $r_2, g(r_1, r_2, s)$, to the reader's challenge r_1 must have been generated *after* the challenge was sent. This property is obviously necessary for recent aliveness and in particular excludes classic replay attacks.

The ARR (algebraic replay resistance) property guarantees that there is no efficient algorithm to compute a response $r_2, g(r_1, r_2, s)$ to the challenge r_1 even after having observed previous challenge-response pairs. Clearly, an attacker's ability to compute such a response violates recent aliveness and this property is thus necessary for recent aliveness. Such an attack generalizes replay attacks in that instead of merely replaying previously observed information, the attacker combines previously obtained challenge-response pairs to compute the response

to a fresh challenge. Hence, we refer to attacks on challenge-response authentication protocols exploiting the lack of the ARR property as *algebraic replay attacks*.

It is obvious that for a function $g(r_1, r_2, s)$ to have the ARR property, it must preserve the secrecy of s. Indeed, cryptographic hash functions are frequently used for the type of challenge-response mechanism considered here. Since the collision resistance property of cryptographic hash functions does not seem necessary for the challenge-response mechanism, the question arises whether all one-way functions satisfy the ARR property and the answer is negative. It is certainly false for all homomorphic one-way functions. Consider, for instance, the Rabin function, defined by $x \to x^2 \bmod N$ for certain composite integers N. If $(r_1, r_2, s) \to g(r_1, r_2, s) = (r_1 r_2 s)^2 \bmod N$ is a Rabin function, then given only one challenge-response pair, $r_1, g(r_1, r_2, s)$ it is easy to compute responses for any challenge r_1', since $g(r_1', r_2, s) = g(r_1, r_2, s) \cdot (r_1'/r_1)^2$.

Furthermore, even non-homomorphic one-way functions will in general not have the ARR property if their *argument* has algebraic properties. As demonstrated in the examples below, there are several protocols that fail to achieve recent aliveness for this very reason. In these protocols the challenge-response construction can typically be represented as $g(r_1, r_2, s) = f(r_1 \circ r_2, s)$, where f is a (non-homomorphic) cryptographic hash function and $\circ$ denotes an operator with the following algebraic property. Given a, b, and c, it is easy to find d with $a \circ b = c \circ d$. This construction clearly does not have the ARR property, regardless of the properties of f. The algebraic replay attack on such a protocol works as follows. An adversary observing one execution of the protocol learns r_1, r_2, and $f(r_1 \circ r_2, s)$. When challenged with r_1', the adversary finds r_2' such that $r_1 \circ r_2 = r_1' \circ r_2'$ and replies with $r_2', f(r_1 \circ r_2, s)$. The attack succeeds because $f(r_1 \circ r_2, s) = f(r_1' \circ r_2', s)$.

Examples of operators $\circ$ for which this type of attack succeeds are *xor*, modular addition, and any associative operator for which it is easy to compute left inverses.

3.1 Examples

We highlight two recent examples of algebraic replay attacks and present several new attacks.

- Chien and Chen [9] implement the challenge-response mechanism by composing the cyclic redundancy check (CRC) function with *xor*. To a challenge r_1, the tag responds with $r_2, CRC(EPC, r_1, r_2) \oplus k$, where EPC is a constant representing the identity of the tag. The attack on this protocol has been first reported by Peris-Lopez et al. [10, §4.2]. It uses the fact that CRC is a homomorphism, i.e. $CRC(a) \oplus CRC(b) = CRC(a \oplus b)$.

  To attack the protocol, the adversary observes one protocol execution. When challenged with r_1' the adversary computes the *xor* of the observed response $CRC(EPC, r_1, r_2) \oplus k$ with $CRC(\mathbf{0}_{EPC}, r_1, \mathbf{0}_{r_2}) \oplus CRC(\mathbf{0}_{EPC}, r_1', \mathbf{0}_{r_2})$. The terms $\mathbf{0}_{EPC}$ and $\mathbf{0}_{r_2}$ are 0-bit strings of length

equal to the length of EPC and r_2, respectively. Because CRC is a homomorphism, the computation will result in a correct response $CRC(EPC, r'_1, r_2)$ to the challenge r'_1.
- The protocol proposed by Lee et al. [11], described in detail in Section 4, is vulnerable to an algebraic replay attack in which the adversary needs to observe three protocol executions or perform a quality-time attack consisting of three queries. The algebraic replay attack can then be executed by solving a small system of equations yielding a constant particular to the tag. While this constant does not reveal the tag's secret information, it can still be used to compute the correct response to a reader's challenge. This attack has been first described by Bringer et al. [12].

The protocols by Chien and Huang [13], Kim et al. [14], Lee et al. [15], and Song and Mitchell [16], are vulnerable to algebraic replay attacks abusing the fact that a hash-like function or a cryptographic hash function is composed with *xor* and fit into the challenge-response construction with the function $f(r_1 \circ r_2, s)$ shown above.

We illustrate a complete attack on the protocol proposed by Chien and Huang [13], depicted in Figure 1 above. The reader is referred to the full version of the paper [17] for detailed attacks on the other protocols. The reader R and tag T share secrets k and ID. The reader starts by sending a random bit string r_1. The tag generates a random string r_2 and hashes the *xor* of r_1, r_2, and the secret k. This hash and ID are used as input for a function in which the ID is rotated by a value depending on the hash. The tag computes the *xor* of the rotated ID and the hash, before sending the left half of the resulting bits and r_2 to the reader. The reader performs the same operations on every pair of ID and k until it finds the corresponding tag. It then sends the right half of the corresponding bits to the tag.

To impersonate a tag, it suffices to notice that the tag's response to the reader's challenge only depends on $r_1 \oplus r_2$ and a shared secret. The composition of functions applied to the *xor* and shared secret can be represented by the function f, defined above. Thus, the adversary can carry out a quality-time attack by challenging a tag with any r_1 to obtain a valid combination of r_1, r_2, and $\mathit{Left}(ID2 \oplus \tilde{g})$. This information suffices for the adversary to be able to respond to any future challenge r'_1 received from a reader. When challenged, the adversary sets $r'_2 = r'_1 \oplus r_1 \oplus r_2$ and sends $r'_2, \mathit{Left}(ID2 \oplus \tilde{g})$.

4 Attribute Acquisition Attacks on Untraceability

A simple, necessary condition for tag untraceability is that an adversary, which has observed a particular tag once, must not be able to recognize the tag as being the same tag in the future. To make this more precise, we call a term, which the adversary can derive from one or more runs of a tag and which identifies the tag to the adversary, a *unique attribute* of the tag. The necessary condition for a tag to be untraceable then is that the adversary must not be able to derive a unique attribute for the tag. Should the adversary be able to compute a unique

attribute, then we refer to the adversary's steps to arrive at such a term as the *attribute acquisition attack*.

A simple unique attribute can be found in protocols where the tag's answer to a challenge c is merely a function $f(c, k)$ of the challenge and a secret (or collection of secrets) k and does not involve any nonce created by the tag. In this case, c is under the adversary's control, k is unique to the tag, and the adversary learns $f(c, k)$ after one round of communication with the tag. Thus for constant c chosen by the adversary, $f(c, k)$ is a unique attribute of the tag whose secret is k.

To prevent long-term traceability in protocols that employ the challenge-response mechanism described, the tag typically updates its secret k at the end of a run. The secret k must therefore also be updated by the reader and in order to avoid desynchronization attacks, the tag needs to authenticate the communicating reader before updating k. Yet, a tag following such a protocol can still be traced by an adversary between two updates by querying the tag and then aborting the protocol. Furthermore, if the update of the secret k at the end of the protocol involves operators with algebraic properties, it is frequently possible for the adversary to compute a unique attribute for the tag which will be valid after the update. References to such protocols are given in the examples section below.

To find unique attributes in general, consider a given RFID protocol in a formal trace model such as the one proposed by Cremers and Mauw [18] or the strand spaces model of Thayer Fàbrega et al. [19]. Then the unique attribute for the tag role can be obtained, if it exists, by computing the intersection of the adversary's knowledge with the set of terms which can be constructed from constants that are unique to the tag and terms that are under the adversary's control. Such a term can be found effectively, provided that the intersection is non-empty.

To find a term in the intersection for the special class of challenge-response protocols in which the tag includes a fresh nonce r in its reply $f(c, k, r)$ to a challenge c, the adversary needs to find challenges $c_1, \ldots, c_l$ and an efficiently computable function $g(x_1, \ldots, x_l)$, such that

$$g(f(c_1, k, r_1), \ldots, f(c_l, k, r_l)) = \tilde{g}(c_1, \ldots, c_l, k)$$

does not depend on the tag's nonces $r_1, \ldots, r_l$. In this case $\tilde{g}(c_1, \ldots, c_l, k)$ is the unique attribute. The attribute acquisition problem displayed in this form is more amenable to solutions by algebraic methods, as the following examples show.

4.1 Examples

We give three examples of attacks that have not been reported in literature. The first two examples are described in more detail in the full version of this paper [17].

1. A simple attribute acquisition attack exists on the protocol proposed by Kim et al. [20]. In this protocol, the tag's response can be represented by

$f(c, k, r) = k_1 \oplus r, h(c, k_2) \oplus r$, where $k = k_1, k_2$ is the tag's secret, c the reader's challenge and r the tag's nonce. To find a unique attribute, the attacker challenges the tag with a constant c_1 and computes the unique attribute by taking the *xor* of the two terms in the response: $k_1 \oplus r \oplus h(c_1, k_2) \oplus r = k_1 \oplus h(c_1, k_2) = \tilde{g}(c_1, k)$.

2. The protocols by Li and Ding [21], Osaka et al. [22], and Yang et al. [23] are stateful protocols that update the shared secrets between reader and tag at the end of a successful protocol execution. The updates take the old secret and a fresh value exchanged in the protocol execution, and apply an operator with algebraic properties to obtain the new secret. By observing the messages exchanged in a protocol execution, the attacker can fabricate a challenge to which the tag will respond with the same term: the unique attribute. In other words, the attacker uses his knowledge to "undo" the update of the tag. In the simplest of these, the protocol by Osaka et al. [22], the reader's challenge is c, the tag's response is $f(c, k) = h(k \oplus c)$, where k is the tag's secret. The tag updates its secret by computing the *xor* of it with a third message r it receives from the reader. Disregarding other flaws this protocol suffers from, the attribute acquisition attack consists in challenging the tag the first time with a constant c_1. After an update with message r the tag is challenged with $c_1 \oplus r$. After the next update with message r', the tag is challenged with $c_1 \oplus r \oplus r'$ and so forth. The tag's response to these challenges is each time $h(k \oplus c_1)$.
3. A more challenging example is the authentication protocol proposed by Lee et al. [11] and shown in Figure 2. The protocol is based on a fixed, system-wide elliptic curve over a finite field. The points P, $Y = yP$, x_1P, x_2P on the elliptic curve are publicly known, the scalar y is only known to the reader, and the scalars x_1, x_2 are unique to each tag and only known to the tag. The elliptic curve is assumed to have been chosen such that the computational Diffie-Hellman problem is hard, that is, given only the points xP, yP, and P on the elliptic curve, it is hard to compute xyP.

   In the protocol, the reader challenges the tag with a random number $r_2 \neq 0$ to which the tag responds with two points $T_1 = r_1P$, $T_2 = (r_1 + x_1)Y$ on the elliptic curve and a scalar $v = r_1x_1 + r_2x_2$. Using this information, the reader can infer the tag's identity. Thus, this protocol, too, is a challenge-response protocol with challenge r_2 and a response that can be written as $f(r_2, k, r_1) = r_1P, (r_1 + x_1)yP, r_1x_1 + r_2x_2$, where $k = x_1, x_2$. The points P and yP are constant. To find a unique attribute, the adversary needs to find challenge terms $c_1, \ldots, c_l$ and functions $g, \tilde{g}$ such that $g(f(c_1, k, r_1), \ldots, f(c_l, k, r_1^{(l)})) = \tilde{g}(c_1, \ldots, c_l, k)$, where $\tilde{g}$ does not depend on the tag's random numbers $r_1, \ldots, r_1^{(l)}$.

   If we write $f(c, k, r_1) = T_1, T_2, v$ as in the protocol specification, and recall that primes indicate terms transmitted in the second run, then

$$g(f(c, k, r_1), f(c, k, r_1')) = \frac{T_1 - T_1'}{v - v'} = x_1^{-1}P$$

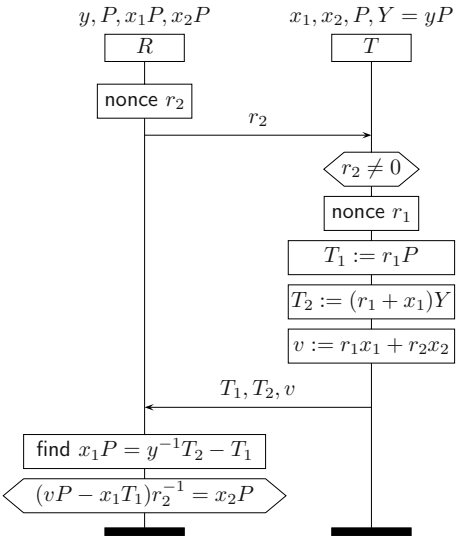

Fig. 2. Protocol with untraceability flaw

depends only on the first part of the secret $k = x_1, x_2$. Thus $\tilde{g}(k) = x_1^{-1}P$ is a unique attribute.

From the definition of the function g, it is now easy to obtain the attribute acquisition attack. By carrying out a quality-time attack, the adversary challenges the tag twice with the same value c. The information received from the tag in the two runs can be used to compute the term $x_1^{-1}P$ as follows. Observe that $v - v' = (r_1 - r_1')x_1$ and $T_1 - T_1' = (r_1 - r_1')P$, thus, multiplying $T_1 - T_1'$ with the inverse of $v - v'$ modulo the order of the elliptic curve, the attacker obtains $x_1^{-1}P$.

Note that after executing this quality-time attack, it suffices for the adversary to challenge any given tag only once with the previously used value c to determine whether the presented tag is equal to the tag identified by $x_1^{-1}P$.

A similar attack on untraceability of the protocol in Figure 2 was independently found by Bringer et al. [12]. The authors observe that for any two protocol executions, the following equations hold:

$$r_2v' - r_2'v = (r_2r_1' - r_2'r_1)x_1$$
$$r_2T_1' - r_2'T_1 = (r_2r_1' - r_2'r_1)P$$

The attacker may then combine these two equations to obtain $x_1^{-1}P$ and proceed as described above.

5 Cryptanalytic Attacks on Secrecy

The authentication and untraceability properties of RFID protocols often rely on the secrecy of shared keys. In some cases, revealing parts of a secret key may already be enough to trace the tag. If sufficiently many bits of a key can be revealed, brute-forcing the remaining bits may become feasible. Formal methods approaches typically do not consider attacks in which an adversary may learn just a few bits of a key, since keys are modeled as atomic terms.

If we assume that operators with algebraic properties are applied to terms sent back and forth between a reader and a tag, then a natural point of attack is to set up equations involving the terms on whose secrecy a protocol depends. Such equations may be obtained by observing several protocol runs, but also by selectively modifying parts of messages. In other words, one may attempt to apply any cryptanalytic method known to mankind. While this is hardly an original strategy, it turns out to be quite successful in the domain of RFID protocols. One reason for this is the popularity of simple operators with algebraic properties. The other reason is due to the simple structure of typical RFID protocols. The reader challenges the tag with a nonce r to which the tag responds with a message involving that nonce and a secret k. This leads to a function $r \mapsto f(k, r)$ which can be compared to a cipher $m \mapsto C(k, m)$ or keyed hash function $x \mapsto h(k, x)$. The tag's response can further be analyzed by forwarding a modified version of it to the reader and checking the reader's response. For RFID protocols with three or more messages, a tag-generated nonce, may frequently be considered as a known plaintext. Finally, stateful RFID protocols, i.e. RFID protocols in which the tag upon successful completion of the protocol updates its secret ID or cryptographic key, can be analyzed by taking advantage of algebraic relations between previous and future ID's or keys.

5.1 Examples

There are several examples of cryptanalytic attacks in the literature.

- In the HB$^+$ protocol of Juels and Weis [24], tags use the binary inner product and *xor* operator to hide their secret keys while proving knowledge of it. The attack by Gilbert et al. [25] breaks secrecy of a tag's key by first modifying the messages exchanged between reader and tag, then observing the reader's behavior, and finally using the observed information to set up and solve a system of linear equations.
- Van Deursen et al. [3] use information obtained through eavesdropping on executions of the Di Pietro and Molva protocol [26] to expose two thirds of the bits of a tag's secret key. In the protocol execution, bits of the tag's secret key are combined with random nonces using *xor* and logical *and* and *or* operators and then sent from the tag to the reader. The attack is carried out by solving a system of linear equations derived from the observed

messages which yields two thirds of the secret key's bits. This is enough to break untraceability. It furthermore permits a brute force attack on the remaining bits in order to break authentication.

- In the protocols of Peris-Lopez et al. [27,28,29], logical *and* and *or* operators are used in addition to *xor* and modular arithmetic leading to information leaks exploited by Alomair et al. [30] and Li and Wang [31].
- Vajda and Buttyán have proposed several lightweight authentication protocols in [32]. Their first protocol uses *xor* and bit permutations to update keys shared between reader and tag. The attack of Alomair et al. [30] correlates keys across updates thereby breaking authentication. Vajda and Buttyán's second protocol is vulnerable to an active attack in which the adversary recovers the shared secret by querying the tag with a challenge of his choice and analyzing the response.

For a concrete, simple example of a hitherto unknown attack, consider the protocol proposed by Kang and Nyang [33]. In this protocol, the tag generates a random value r_0 from a small domain and a random value r_1 of length n. The tag sends the two hashes $h(ID, r_0)$, $h(r_1, k)$ and $ID \oplus r_1$ to the reader. Using $h(ID, r_0)$, the reader finds ID by trying out all combinations of values for ID stored in its database and of all possible values for r_0. This is possible for the reader because r_0 is chosen from a small domain and the number of IDs stored in its database is very small compared to the number of possible IDs. Using ID the reader retrieves k from its database, and using $ID \oplus r_1$ and ID, the reader finds r_1 and may then verify the correctness of the value of $h(r_1, k)$. The reader then generates a random value r_2 of length n and sends $ID \oplus r_2$ and $h(r_1, r_2)$ to the tag. The tag verifies these and sends $r_1 + r_2 \bmod 2^n$ back to the reader. Both tag and reader update the ID by *xor*-ing it with $r_1 \oplus r_2$. The protocol is depicted in Figure 3.

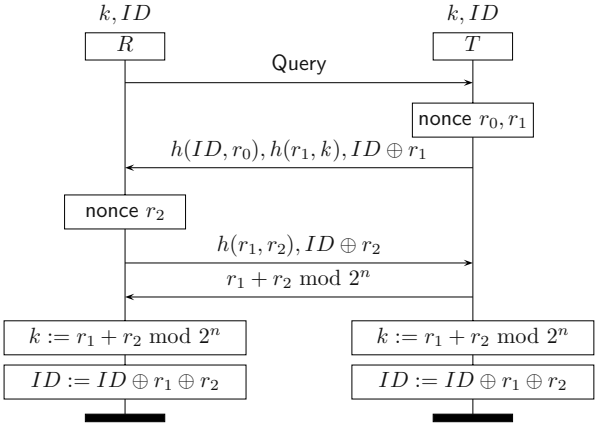

Fig. 3. Several bits of ID leak in every run

Since hash functions are assumed to be perfect, we consider the terms $ID \oplus r_1$, $ID \oplus r_2$, and $r_1 + r_2 \bmod 2^n$, setting up a system of equations involving the variables ID, r_1, r_2, and the values observed during runs of the protocol. A moment's thought shows that we may combine the first two equations to obtain $r_1 \oplus r_2$.

For convenience, we set $V = r_1 + r_2 \bmod 2^n$ and $W = r_1 \oplus r_2$. Let $V[i]$ be the i-th bit of V, and similarly for W, r_1, and r_2. Furthermore, let $V[1]$ be the least significant bit of V. By comparing addition modulo 2^n with *xor* it is easy to see that $V[i+1] \neq W[i+1]$ only if there is a carry bit in the computation of $V[i]$. If this is the case, then $r_1[i] \neq r_2[i] \Leftrightarrow W[i] = 1$ and $r_1[i] = r_2[i] = 1 \Leftrightarrow W[i] = 0$.

Since the latter case determines $r_1[i]$ and $r_2[i]$ uniquely, it follows that it can be used to find the i-th bit of ID. More bits from ID can be obtained by noticing that a carry bit in $V[i]$ followed by no carry bit in $V[i+1]$ implies $r_1[i+1] = r_2[i+1] = 0$.

Since r_1 and r_2 are chosen at random, on average, every communication session leaks roughly $\frac{n-1}{4}$ bits of ID. Revealing all bits of ID, once sufficiently many bits are known, can be achieved with a brute-force search over possible values for ID and r_0 and comparing their hash to $h(ID, r_0)$. Revealing all bits of ID is made a little more complicated by the fact that reader and tag update ID at the end of every protocol execution by setting it to $ID \oplus r_1 \oplus r_2$. The adversary may therefore need to keep track of two or three consecutive protocol executions between the tag and reader before performing the exhaustive search in order to completely reveal the tag's ID. Knowing the ID, the adversary can impersonate both tag and reader and furthermore trace the tag.

6 Conclusion and Future Work

By analyzing simple necessary conditions for authentication and untraceability and studying information leakage in secret terms, we have found three categories of attacks on recently published RFID protocols. The attack methods are particularly suitable for RFID protocols since they take advantage of algebraic properties of operators and functions typically used in these protocols. The methods used to find algebraic replay and attribute acquisition attacks are sufficiently straight-forward that we expect them to be easily implementable in a tool-supported verification framework. The tool-supported verification of secrecy and authentication properties in presence of associative and commutative operators is already a very active research area. The automatic verification of untraceability will be considered in future work following the procedure outlined in Section 4. An indication for how some of the cryptanalytic attacks may be automated can be obtained from the attack presented in Section 5. By representing all atomic terms as bit vectors, the system of equations of atomic terms can be expanded to a larger system over the finite field of two elements involving the bits of the vectors as variables. Such a system can, in principle, be solved using SAT solvers or Gröbner basis algorithms.

References

1. Dolev, D., Yao, A.: On the security of public key protocols. IEEE Transactions on Information Theory IT-29(2), 198–208 (1983)
2. Lowe, G.: A hierarchy of authentication specifications. In: CSFW, pp. 31–44 (1997)
3. van Deursen, T., Mauw, S., Radomirović, S.: Untraceability of RFID Protocols. In: Onieva, J.A., Sauveron, D., Chaumette, S., Gollmann, D., Markantonakis, K. (eds.) WISTP 2008. LNCS, vol. 5019, pp. 1–15. Springer, Heidelberg (2008)
4. Avoine, G.: Adversary model for radio frequency identification. Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC), Lausanne, Switzerland (September 2005)
5. Juels, A., Weis, S.: Defining strong privacy for RFID. In: IEEE International Conference on Pervasive Computing and Communications – PerCom 2007, New York, USA, pp. 342–347. IEEE Computer Society Press, Los Alamitos (2007)
6. Vaudenay, S.: On privacy models for RFID. In: Kurosawa, K. (ed.) ASIACRYPT 2007. LNCS, vol. 4833, pp. 68–87. Springer, Heidelberg (2007)
7. Damgård, I., Pedersen, M.Ø.: RFID security: Tradeoffs between security and efficiency. In: Malkin, T.G. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 318–332. Springer, Heidelberg (2008)
8. Paise, R.I., Vaudenay, S.: Mutual authentication in RFID: Security and privacy. In: ACM Symposium on Information, Computer and Communications Security (ASIACCS 2008), pp. 292–299. ACM Press, New York (2008)
9. Chien, H.Y., Chen, C.H.: Mutual authentication protocol for RFID conforming to EPC class 1 generation 2 standards. Computer Standars & Interfaces, Elsevier Science Publishers 29(2), 254–259 (2007)
10. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J., Ribagorda, A.: Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard (2007)
11. Lee, Y.K., Batina, L., Verbauwhede, I.: EC-RAC (ECDLP based randomized access control): Provably secure RFID authentication protocol. In: Proceedings of the 2008 IEEE International Conference on RFID, pp. 97–104 (2008)
12. Bringer, J., Chabanne, H., Icart, T.: Cryptanalysis of EC-RAC, a RFID identification protocol. In: Franklin, M.K., Hui, L.C.K., Wong, D.S. (eds.) CANS 2008. LNCS, vol. 5339, pp. 149–161. Springer, Heidelberg (2008)
13. Chien, H.Y., Huang, C.W.: A lightweight RFID protocol using substring. In: Kuo, T.-W., Sha, E., Guo, M., Yang, L.T., Shao, Z. (eds.) EUC 2007. LNCS, vol. 4808, pp. 422–431. Springer, Heidelberg (2007)
14. Kim, K.H., Choi, E.Y., Lee, S.M., Lee, D.H.: Secure EPCglobal class-1 gen-2 RFID system against security and privacy problems. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4277, pp. 362–371. Springer, Heidelberg (2006)
15. Lee, S., Asano, T., Kim, K.: RFID mutual authentication scheme based on synchronized secret information. In: Symposium on Cryptography and Information Security, Hiroshima, Japan (January 2006)
16. Song, B., Mitchell, C.J.: RFID authentication protocol for low-cost tags. In: Wireless Network Security (WISEC), pp. 140–147 (2008)
17. van Deursen, T., Radomirović, S.: Attacks on RFID protocols (version 1.0). Cryptology ePrint Archive, Report 2008/310 (July 2008), `http://eprint.iacr.org/2008/310`

18. Cremers, C., Mauw, S.: Operational Semantics of Security Protocols. In: Leue, S., Systä, T.J. (eds.) Scenarios: Models, Transformations and Tools. LNCS, vol. 3466, pp. 66–89. Springer, Heidelberg (2005)
19. Thayer Fàbrega, F., Herzog, J., Guttman, J.: Strand spaces: Why is a security protocol correct? In: Proc. 1998 IEEE Symposium on Security and Privacy, Oakland, California, pp. 66–77 (1998)
20. Kim, I.J., Choi, E.Y., Lee, D.H.: Secure mobile RFID system against privacy and security problems. In: SecPerU 2007 (2007)
21. Li, Y., Ding, X.: Protecting RFID communications in supply chains. In: ASIACCS, pp. 234–241 (2007)
22. Osaka, K., Takagi, T., Yamazaki, K., Takahashi, O.: An efficient and secure RFID security method with ownership transfer. In: Wang, Y., Cheung, Y.-m., Liu, H. (eds.) CIS 2006. LNCS, vol. 4456, pp. 778–787. Springer, Heidelberg (2007)
23. Yang, J., Park, J., Lee, H., Ren, K., Kim, K.: Mutual authentication protocol for low-cost RFID. In: Handout of the Ecrypt Workshop on RFID and Lightweight Crypto (July 2005)
24. Juels, A., Weis, S.: Authenticating Pervasive Devices with Human Protocols. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 293–308. Springer, Heidelberg (2005)
25. Gilbert, H., Robshaw, M., Sibert, H.: An active attack against HB^+ – a provably secure lightweight authentication protocol (July 2005) (manuscript)
26. Di Pietro, R., Molva, R.: Information confinement, privacy, and security in RFID systems. In: Biskup, J., López, J. (eds.) ESORICS 2007. LNCS, vol. 4734, pp. 187–202. Springer, Heidelberg (2007)
27. Peris-Lopez, P., Hernandez-Castro, J.C., Estevez-Tapiador, J.M., Ribagorda, A.: M^2AP: A minimalist mutual-authentication protocol for low-cost RFID tags. In: Ma, J., Jin, H., Yang, L.T., Tsai, J.J.-P. (eds.) UIC 2006. LNCS, vol. 4159, pp. 912–923. Springer, Heidelberg (2006)
28. Peris-Lopez, P., Castro, J.C.H., Estévez-Tapiador, J.M., Ribagorda, A.: EMAP: An efficient mutual-authentication protocol for low-cost RFID tags. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2006 Workshops. LNCS, vol. 4277, pp. 352–361. Springer, Heidelberg (2006)
29. Peris-Lopez, P., Castro, J.C.H., Estévez-Tapiador, J.M., Ribagorda, A.: LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In: Printed handout of Workshop on RFID Security – RFIDSec 2006 (July 2006)
30. Alomair, B., Lazos, L., Poovendran, R.: Passive attacks on a class of authentication protocols for RFID. In: Nam, K.-H., Rhee, G. (eds.) ICISC 2007. LNCS, vol. 4817, pp. 102–115. Springer, Heidelberg (2007)
31. Li, T., Wang, G.: Security analysis of two ultra-lightweight RFID authentication protocols. In: IFIP SEC 2007, Sandton, Gauteng, South Africa, IFIP (May 2007)
32. Vajda, I., Buttyán, L.: Lightweight authentication protocols for low-cost RFID tags. In: Second Workshop on Security in Ubiquitous Computing – Ubicomp 2003, Seattle, WA, USA (October 2003)
33. Kang, J., Nyang, D.: RFID authentication protocol with strong resistance against traceability and denial of service attacks. In: Molva, R., Tsudik, G., Westhoff, D. (eds.) ESAS 2005. LNCS, vol. 3813, pp. 164–175. Springer, Heidelberg (2005)