# Improving Automatic Verification of Security Protocols with XOR

Xihui Chen[1,2], Ton van Deursen[1⋆], and Jun Pang[1]

[1] Faculty of Sciences, Technology and Communication
University of Luxembourg, 6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg
[2] School of Computer Science and Technology
Shandong University, Jinan, 250101 China

**Abstract.** Küsters and Truderung recently proposed an automatic verification method for security protocols with exclusive or (XOR). Their method reduces protocols with XOR to their XOR-free equivalents, enabling efficient verification by tools such as ProVerif. Although the proposed method works efficiently for verifying secrecy, verification of authentication properties is inefficient and sometimes impossible.
In this paper, we improve the work by Küsters and Truderung in two ways. First, we extend their method for authentication verification to a richer class of XOR-protocols by automatically introducing bounded verification. Second, we improve the efficiency of their approach by developing a number of dedicated optimizations. We show the applicability of our work by implementing a prototype and applying it to both existing benchmarks and RFID protocols. The experiments show promising results and uncover a flaw in a recently proposed RFID protocol.

## 1 Introduction

Cryptographic security protocols typically consists of a series of message exchanges among two or more agents over a hostile network. They aim to achieve various security goals such as authentication, secrecy, key agreement, privacy, and anonymity. However, designing secure protocols is an error-prone task and incorrect protocols may become ideal entry points for various attacks. Starting from the seminal work by Lowe [1], automated symbolic verification methods for security protocols have shown their strength in finding attacks and proving correctness of security protocols.

As attacks that rely on cryptographic primitives are hard to prove and difficult to be automatically checked, cryptographic primitives are usually treated as functions without any algebraic properties in symbolic methods. This is called the perfect cryptography assumption [2], namely no cryptographic message can be opened without the correct key. Based on this assumption, many automatic tools have been designed and implemented, among which ProVerif [3] is considered as the state of the art [4]. However, ProVerif cannot uncover attacks that

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

make use of certain algebraic properties of cryptographic primitives. Cortier, Delaune and Lafourcade give a survey on algebraic properties of common cryptographic primitives and attacks making use of them [5]. Therefore, some relaxation of the perfect assumption needs to be investigated. Exclusive or (XOR) is one binary operator with typical algebraic properties that has drawn a lot of interest. For example, XOR is often used in radio frequency identification (RFID) systems, which have become popular in recent years.

We call security protocols employing the exclusive or operator ($\oplus$) XOR-protocols. The $\oplus$-operator has the following four properties.

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z \qquad \text{(associativity)} \qquad (1)$$

$$x \oplus y = y \oplus x \qquad \text{(commutativity)} \qquad (2)$$

$$x \oplus 0 = x \qquad \text{(neutral element)} \qquad (3)$$

$$x \oplus x = 0 \qquad \text{(nilpotence)} \qquad (4)$$

In order to detect attacks on XOR-protocols, we need to model intruders with the ability of exploring the above algebraic properties, in addition to the perfect cryptography assumption.

**Related work.** In the literature, several approaches have been proposed to deal with the verification of XOR-protocols [6–8], but few of them are practical to implement. A few tools can cope with a certain class of XOR-protocols [8, 9], all of them have strict restrictions on the range of protocols they can be applied to. For example, the tool of Cortier, Keighren, and Steel can only handle protocols with the $\oplus$-operator and symmetric encryption. More recently, Küsters and Truderung proposed a more general approach [10] to automatic verification of cryptographic XOR-protocols based on ProVerif. Their main idea is to reduce protocol analysis with XOR to the XOR-free case. The XOR-reduction step transforms Horn theories modeling XOR-protocols to the ones free from algebraic properties of the $\oplus$-operator, by computing a family of legal substitutions for terms containing $\oplus$. Thus, verification is reduced to a syntactic derivation problem. They implement their transformation step in a tool called XorProverif [10]. The use of ProVerif allows the modeling of essential cryptographic primitives and the verification of security protocols with an unbounded number of sessions. However, there are still a few limitations of this XOR-reduction approach – only $\oplus$-linear protocols can be handled (see Sect. 2 for the definition of $\oplus$-linearity) and it is likely to suffer from exponential blow up of the number of substitutions (Lem. 12, [10]). In this paper, we develop several methods to tackle these restrictions of the XOR-reduction approach, and implement a prototype to evaluate and illustrate our methods by experiments on existing benchmarks and recent RFID protocols.

**Our main contribution.** One goal of this research is to develop a systematic method to improve efficiency of the XOR-reduction approach. Our first idea is to reduce the number of substitutions during the transformation, by exploring the freshness of nonces generated during the executions of the XOR-protocols. By

this further reduction, the time taken by ProVerif for verification is decreased and some false attacks can be removed as well.

We also propose a new approach to use *bounded verification* to make the XOR-reduction approach available to verify authentication of more protocols which violate $\oplus$-linearity. In this approach, session identifiers are considered as constants instead of variables [10] and we can verify protocols using models with a certain bounded number of sessions. Our bounded verification can be further optimized by restricting the order between sessions and by checking the secrecy property first. RFID protocols are a special class of protocols that require authentication. They often use the $\oplus$-operator to build protocol messages. In terms of the characteristics of RFID protocols, more optimizations can be introduced and their protocol models could be much more simplified.

We implement a prototype to evaluate and illustrate our methods: it first automatically transforms the original Horn theory of an XOR-protocol to a multi-session model, then it reduces the model XOR-free and performs the introduced optimizations when necessary. In the end, ProVerif is applied to the final result of the transformations. A number of XOR-protocols including RFID protocols have been analyzed and experimental results show that our approach is effective and improves the verification of XOR-protocols based on the XOR-reduction approach. In one case, a new attack is detected on a RFID protocol in its multi-session model.

**Structure of this paper.** In Sect. 2, we present the main idea of the XOR-reduction approach with a running example. The concepts of bounded verification are introduced in Sect. 3. Several different ways to do optimizations are presented in Sect. 4. We discuss our implementation and experimental results in Sect. 5. We conclude the paper in Sect. 6.

## 2 Preliminaries

In this section, we illustrate how security protocols with $\oplus$ can be modeled by Horn theories and explain the main ideas behind the reduction process proposed by Küsters and Truderung. More details can be found in the original paper [10].

### 2.1 Basic Concepts

We use $\Sigma$ to denote a finite signature containing the binary function symbol $\oplus$ and $V$ to denote a set of variables. The set of terms is defined as usual over $\Sigma$ and $V$. We use $s \sqsubseteq t$ to denote that $s$ is a subterm of $t$. Terms containing no variables are *ground* and are also called *messages*. For a unary predicate $q$ and a (ground) term $t$, we call $q(t)$ a (ground) *atom*. A *substitution* $\sigma$ is a set of pairs $\{t_1/x_1, \ldots, t_n/x_n\}$, where $t_1, \ldots, t_n$ are terms and $x_1, \ldots, x_n$ are variables. We use $dom(\sigma)$ to denote the domain of $\sigma$, which contains the variables $x_1 \ldots x_n$. A term is *standard* if its top symbol is not $\oplus$, otherwise it is called *non-standard*. Equations (1)-(4) define a congruence relation $\sim$ on terms. A term is in *reduced*

*form* if equations (1)-(2) and equations (3)-(4), when interpreted as reductions from left to right, can no longer be applied.

A Horn clause is of the form of $a_1, \ldots, a_n \rightarrow a_0$ where $a_0, \ldots, a_n$ are atoms. A set of Horn clauses constitutes a Horn theory. Given a ground atom $a$, we use $T \vdash a$ to denote that there is a derivation $\pi$ for $a$ from the Horn theory $T$. A derivation $\pi$ is a sequence of ground atoms $b_1, \ldots, b_\ell$ with $b_\ell = a$. For each $b_i$ there exists a substitution $\sigma$ of a Horn clause $a_1, \ldots, a_n \rightarrow a_0$ in $T$, we have $a_1\sigma, \ldots, a_n\sigma \rightarrow a_0\sigma$ where $a_0\sigma = b_i$ and for every $j \in \{1, \ldots, n\}$ there exists $k \in \{1, \ldots, i-1\}$ with $a_j\sigma = b_k$. Similarly, if the congruence relation $\sim$ is used instead of syntactic equality $=$, we can say $a$ can be derived from $T$ modulo $\oplus$, denoted by $T \vdash_\oplus a$.

One crucial notion in [10] is $\oplus$-*linearity*. A term is $\oplus$-linear if for each of its subterms of the form $t \oplus s$, $t$ or $s$ is ground. For example, $a \oplus x$ is $\oplus$-linear while $a \oplus x \oplus y$ is not, where $x$, $y$ are variables and $a$ is a constant. The concept of $\oplus$-linearity extends to Horn theories and derivations in a straightforward way. Küsters and Truderung also define the notion of $C$-*domination* [10]. Let $C$ denote a finite set of standard reduced ground terms such that $C$ does not contain two terms $m$ and $m'$ such that $m \neq m'$ and $m \sim m'$. We use $C^\oplus$ to denote the $\oplus$-closure of $C$, that is,

$$C^\oplus = \{t \mid \text{there exist } c_1, \ldots, c_n \in C \text{ s.t. } t \sim c_1 \oplus \cdots \oplus c_n\}.$$

A term is $C$-*dominated* if for each of its subterms of the form $t \oplus s$, it is true that either $t$ or $s$ is in $C^\oplus$. The concept of $C$-domination extends to Horn clauses and derivations. A Horn theory is called $C$-dominated if each clause in $T$ is $C$-dominated, except for the rule $I(x), I(y) \rightarrow I(x \oplus y)$ which models the intruder's ability to perform XOR operations. The set $C$ is always finite and must be chosen as small as possible in order to make the XOR-reduction efficient (see Lem. 2 and Lem. 12 in [10]).

### 2.2 Modeling Protocols by Horn Theories

A Horn theory modeling security protocols contains three parts: *initial intruder facts*, *intruder rules*, and *protocol rules*. It uses the predicate $I$. A fact $I(t)$ means that the intruder can obtain the term $t$. The initial intruder facts represent the initial intruder knowledge, typically names of principals and public keys, for instance, $I(a)$ denotes that the intruder knows the name $a$ and $I(pub(sk_a))$ denotes that the intruder knows the public key of $a$ where $sk_a$ represents its private key. The set of Dolev-Yao intruder [2] rules representing the ability to derive new messages can be found in [10], where a special clause $I(x), I(y) \rightarrow I(x \oplus y)$, called the $\oplus$-*rule*, is used to allow the intruder to perform the XOR operation on arbitrary messages. The protocol rules represent the actions performed in a protocol. Each rule $i$ is of the form $I(t_1), \ldots, I(t_i) \rightarrow I(s_i)$ where $t_1, \ldots, t_i$ describe messages the principal has received up to step $i$ and $I(s_i)$ describes the message the principal will send out at step $i$.

The secrecy property of a term $t$ can be formulated as the fact that $I(t)$ cannot be derived from the set of clauses, while authentication properties are

often expressed as correspondence assertions of the form $end(x) \rightarrow begin(x)$, where $x$ describes the value on which both agents agree [11]. Due to the difference, we give the Horn theories for secrecy and authentication verification of our running example $NSL'_{\oplus}$ separately. Fig. 1(a) depicts the $NSL'_{\oplus}$ protocol, which is a variation of the protocol by Lowe [1] that fixes a vulnerability in the Needham-Schroeder protocol [12].

In this paper, we use *role* to refer to the protocol steps an agent expects to carry out, for instance $A$ and $B$ in Fig. 1(a)[3]. For example, agent $a$ playing role $A$ has two steps. To start, $a$ generates a nonce and sends the first message to the agent playing role $B$. Then upon receiving the second message and checking its correctness, $a$ sends back the last message. A *run* is the execution of a role by an agent. Several runs can be executed at the same time. By *session*, we mean a (prefix of a) complete run of an agent. Let $P$ denote the sets of participants and $H$ be the set of honest agents. The notations $sk_a$ and $pub(sk_a)$ represent the private and the corresponding public key of $a \in P$. Comon-Lundh and Cortier prove that for secrecy (authentication properties), only two (three) participants [13] need to be considered. Therefore, we have $P = \{a, b\}$, $H = \{a\}$ for $NSL'_{\oplus}$-sec and $P = \{a, b, c\}$, $H = \{a, b\}$ for $NSL'_{\oplus}$-auth. We use $n(a, b)$ to denote the nonce in the first message in which $a \in P$ is the generator and $b \in P$ is the receiver, and $m(b, a)$ in the second message to denote the nonce sent from $b$ to $a$. Encryption of a term $t$ under a key $k$ is denoted by $\{t\}_k$.

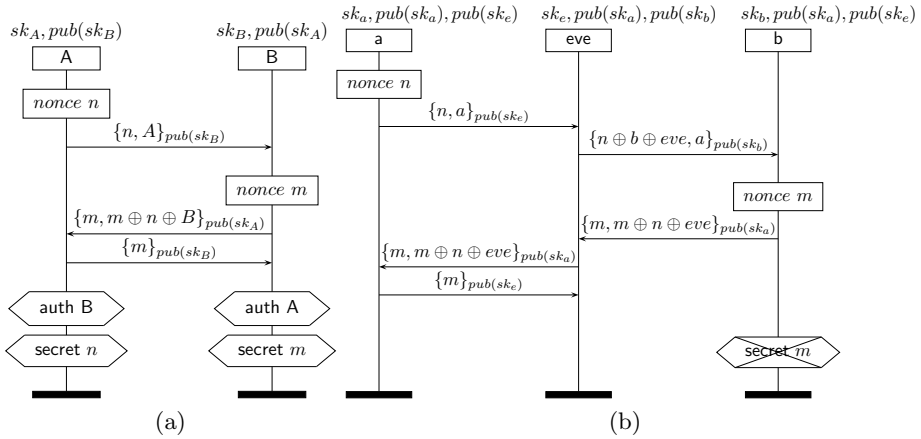


Fig. 1: Description of the $NSL'_{\oplus}$ protocol (a) and one of its attacks (b).

[3] We use message sequence charts for the descriptions of protocols and/or their possible attacks, where capital letters represent roles and small letters are used to represent agents.

*Model for secrecy verification $NSL'_{\oplus}$-sec.* We model the protocol using the following clauses:

$$I(\{n(a,b),a\}_{pub(sk_b)}) \text{ for } a \in H, b \in P \quad (5)$$
$$I(\{x,a\}_{pub(sk_b)}) \rightarrow I(\{m(b,a), m(b,a) \oplus x \oplus b\}_{pub(sk_a)}) \text{ for } b \in H, a \in P \quad (6)$$
$$I(\{y, y \oplus n(a,b) \oplus b\}_{pub(sk_a)}) \rightarrow I(\{y\}_{pub(sk_b)}) \text{ for } a \in H, b \in P \quad (7)$$

We denote the set of clauses above by $NSL'_{\oplus}$-sec. One attack breaking the secrecy claim of $m(b,a)$ is described in Fig. 1(b) where the adversary impersonates $a$ to $b$. After receiving the message $\{n(a,eve),a\}_{pub(sk_e)}$, it makes use of the algebraic properties of XOR and its knowledge of the protocol to send out a message $\{n(a,eve) \oplus b \oplus eve, a\}_{pub(sk_b)}$ to $b$. It replays the response from $b$, $\{m(b,a), m(b,a) \oplus n(a,eve) \oplus eve\}_{pub(sk_a)}$, to $a$. In the end, the adversary can obtain $m(b,a)$ by decrypting the last message from $a$.

*Model for authentication $NSL'_{\oplus}$-auth.* For authentication verification, nonces generated in a session are typically chosen as the parameter $x$ in the events $begin(x)$ and $end(x)$. To guarantee their freshness and prevent replay attacks, session identifiers need to be added to nonces to make expressing correspondence of sessions possible. The following Horn theory models the protocol rules to verify if role $A$ can be authenticated by $B$.

$$I(\{n(a,b,sid),a\}_{pub(sk_b)}) \text{ for } a \in H, b \in P \quad (8)$$
$$I(\{x,a\}_{pub(sk_b)}) \rightarrow I(\{m(b,a,sid), m(b,a,sid) \oplus x \oplus b\}_{pub(sk_a)}) \text{ for } b \in H, a \in P \quad (9)$$
$$begin(a,b,y), I(\{y, y \oplus n(a,b,sid) \oplus b\}_{pub(sk_a)}) \rightarrow I(\{y\}_{pub(sk_b)}) \text{ for } a \in H, b \in P \quad (10)$$
$$I(\{m(b,a,sid)\}_{pub(sk_b)}) \rightarrow end(a,b,m(b,a,sid)) \text{ for } b \in H, a \in P \quad (11)$$

The set of clauses we defined above is denoted by $NSL'_{\oplus}$-auth.

### 2.3 The XOR-reduction Process

We refer to the process of reducing the deduction problem modulo XOR to the one without XOR for $C$-dominated theories as XOR-reduction. XOR-reduction aims to construct a Horn theory that can be analyzed by ProVerif and makes sure that any derivation obtained from the theory modulo XOR can also be derived from the constructed one.

Each $C$-dominated term can be turned into normal form after fixing a linear ordering on $C$. The operator $\ulcorner \cdot \urcorner$ denotes this operation. Any two $C$-dominated terms $t$ and $t'$ such that $t \sim t'$ have the same normal form, that is $\ulcorner t \urcorner = \ulcorner t' \urcorner$. If all terms in $C^{\oplus}$ are in normal form, we have the set $C^{\oplus}_{norm}$. A *fragile subterm* $t'$ of a $C$-dominated term $t$ is a non-ground, standard term occurring in a subterm of $t$ of the form $t' \oplus s$ or $s \oplus t'$. We use $\mathcal{F}(t)$ to represent the set of all fragile subterms of $t$. The concept of fragile subterms extends to Horn clauses.

For example, the dominating set for $NSL'_{\oplus}$-sec is

$$\{m(a,b), m(a,a), n(a,b), n(a,a), a, b\}.$$

Considering the term $m(b,a) \oplus x \oplus b$ in rule (6), its fragile subterm is $x$.

**Definition 1 (Def. 4 in [10]).** *Let $t$ be a $C$-dominated term. The family of substitutions $\sum(t)$ for $t$ with respect to $\mathcal{F}(t)$ is defined as follows. The domain of every substitution in $\sum(t)$ is the set of all variables which occur in some $s \in \mathcal{F}(t)$. Consider a substitution $\sigma \in \sum(t)$. For each $x \in dom(\sigma)$ one of the following three cases holds: (i) $\sigma(x) = x$; (ii) $x \in \mathcal{F}(t)$ and $\sigma(x) = c \oplus x$ for some $c \in C^{\oplus}_{norm}$, $c \neq 0$; (iii) $x$ occurs in a fragile subterm $s$ and there exists a substitution $\sigma'$ in normal form satisfying $s\sigma' \in C^{\oplus}$ then $\sigma(x) = \sigma'(x)$.*

Now given a Horn theory modulo XOR, $T$, we can reduce it to an XOR-free one $T^+$ as follows

$$\ulcorner r_1\sigma\urcorner, \cdots, \ulcorner r_n\sigma\urcorner \rightarrow \ulcorner r_0\sigma\urcorner \quad \text{for each } \sigma \in \sum(\langle r_0, \ldots, r_n\rangle) \tag{12}$$

$$I(c), I(c') \rightarrow I(\ulcorner c \oplus c'\urcorner) \quad \text{for each } c, c' \in C^{\oplus}_{norm} \tag{13}$$

$$I(c), I(x) \rightarrow I(c \oplus x) \quad \text{for each } c \in C^{\oplus}_{norm} \tag{14}$$

$$I(c), I(c' \oplus x) \rightarrow I(\ulcorner c \oplus c'\urcorner \oplus x) \quad \text{for each } c, c' \in C^{\oplus}_{norm} \tag{15}$$

$$I(c \oplus x), I(c' \oplus x) \rightarrow I(\ulcorner c \oplus c'\urcorner) \quad \text{for each } c, c' \in C^{\oplus}_{norm} \tag{16}$$

where rule (12) is applied to each rule $r_1, \ldots, r_n \rightarrow r_0$ of $T$. The Horn clauses except for the $\oplus$-rule can be simulated by the rules in (12). The rules (13)-(16) are used to simulate the $\oplus$-rule. Küsters and Truderung prove that a message can be derived from $T$ modulo XOR if and only if it can be derived from $T^+$ only with a syntactic derivation, that is, no algebraic properties of XOR need to be considered.

We take $NSL'_{\oplus}$-sec as an example to show how the reduction works. It is $\oplus$-linear with dominating set $C = \{m(a,b), m(a,a), n(a,b), n(a,a), a, b\}$. We suppose the order on $C$ is how they are listed. The set $C^{\oplus}_{norm}$ can also be computed. Since only the Horn clauses in (6) and (7) have a fragile subterm $x$, we need to compute its set of substitutions whose domain is $\{x\}$. Other clauses should be included in the new theory unchanged.

Consider an instantiated clause of rule (6)

$$I(\{x, a\}_{pub(sk_a)}) \rightarrow I(\{m(a,a), m(a,a) \oplus x \oplus a\}_{pub(sk_a)}) \tag{17}$$

According to Def. 1, case (i) always holds so it gives $\sigma_1(x) = x$. Case (ii) gives 63 substitutions such as $\sigma_i(x) = m(a,a) \oplus n(a,a) \oplus x$. For case (iii), we have another 64 substitutions. For instance, $\sigma_j(x) = m(a,b) \oplus n(a,a)$ will be included. In the end, we have 128 substitutions in total. For each of them, we obtain an instance of rule (6). For example, after applying $\sigma_i$ we have:

$$I(\{m(a,a) \oplus n(a,a) \oplus x, a\}_{pub(sk_a)}) \rightarrow I(\{m(a,a), n(a,a) \oplus a \oplus x\}_{pub(sk_a)})$$

We can obtain the reduced Horn clauses for other instantiated clauses in a similar way. The clauses (13)-(16) model the $\oplus$-rule. In our running example, for instance, $I(m(a,a) \oplus a), I(a \oplus x) \rightarrow I(m(a,a) \oplus x)$ will be an instance of clause (15).

## 3 Bounded Verification of Authentication Protocols

In the Horn theory based approach new protocol runs do not necessarily use fresh nonces [10]. Therefore, nonces from different runs need to be disambiguated. The standard solution is to add a special *session identifier variable* (*sid*) to terms representing nonces. During verification, *sid* is automatically instantiated by a fresh random value. Freshness of nonces is only required when verifying security properties that need correspondence at run level. Note the difference between $m(a, b, sid)$ and $m(a, b)$ in $NSL'_{\oplus}$-auth and $NSL'_{\oplus}$-sec, respectively.

As a consequence, Horn theories that are $\oplus$-linear when verifying secrecy can become non-$\oplus$-linear when verifying authentication properties. For instance, $NSL'_{\oplus}$-auth is not $\oplus$-linear since it contains a term $m(b, a, sid) \oplus x \oplus b$, where both $m(b, a, sid)$ and $x$ are non-ground. As observed by Küsters and Truderung [10], *sid* is a special variable, because it cannot be substituted by $C$-dominated terms. In the sequel, we call variables that can be substituted by $C$-dominated terms *$C$-variables*. Protocol models that are not $\oplus$-linear solely because of the introduction of session identifiers form a special class of XOR-protocols, which we call *nonce-$\oplus$-linear*.

**Definition 2 (Nonce-$\oplus$-linear).** *A term is* nonce-$\oplus$-linear *if for each of its subterms of the form $s \oplus t$, $s$ or $t$ contains no $C$-variables.*

For example, the term $h(n(a, b, sid)) \oplus x$ is nonce-$\oplus$-linear while $h(n(a, b, sid) \oplus x) \oplus y$ is not, where $n(a, b, sid)$ is a nonce and $x$, $y$ are variables. The concept of nonce-$\oplus$-linearity extends to Horn clauses and theories in a similar fashion to $\oplus$-linearity.

By instantiating the variable *sid* with a fixed finite set $S = \{s_1, \ldots, s_n\}$ of session identifiers, nonce-$\oplus$-linear protocols can be transformed into $\oplus$-linear protocols. Note that $S$ must not intersect with $T$. We then obtain the multi-session Horn theory $T_n$ by replacing *sid* with each $s_i \in S$.

**Definition 3 (Multi-session Horn Theory).** *Let $T$ be a Horn theory, and let $\sigma_i$ $(1 \leq i \leq n)$ be the substitutions mapping sid to $s_i$ and the identity map for other terms. Then multi-session Horn theory of $T$ is defined by*

$$T_n = \bigcup_{1 \leq i \leq n} \sigma_i(T)$$

Clearly, transforming a nonce-$\oplus$-linear Horn theory into a multi-session Horn theory as in Def. 3 makes it $\oplus$-linear.

We now give a theorem about the correctness of our multi-session transformation. Suppose there is a $C$-dominated message using at most $n$ sessions of any agent to derive. We can derive it from $T$ if and only if it can also be derived from $T_n^{\oplus}$ through syntactic derivations. Since $T_n^{\oplus}$ is XOR-free, ProVerif can be used to analyze it.

**Theorem 1.** *Given a nonce-$\oplus$-linear Horn theory $T$, the corresponding multi-session XOR-free Horn theory $T_n^{\oplus}$ and a $C$-dominated message $f$ which can be derived using at most $n$ sessions of participating agents, $T \vdash_{\oplus} f$ iff $T_n^{\oplus} \vdash f$.*

In the sequel, let $T_n$ be the $n$-session model transformed from $T$. We prove the theorem by proving the following two lemmas.

**Lemma 1.** *If $\pi$ is a syntactic derivation for $f$ from $T_n^{\oplus}$, then $\pi$ is a derivation for $f$ from $T$ modulo XOR.*

*Proof.* From Lem. 13 in [10], if there is a derivation $\pi$ for $f$ from $T_n^{\oplus}$, then $\pi$ is also a derivation for $f$ from $T_n$ modulo XOR. Therefore, to prove this lemma it suffices to prove that if $\pi$ is a derivation for $f$ from $T_n$ modulo XOR, then it is also a derivation for $f$ from $T$. Thus, we need to prove each $\pi(i)$ can be obtained by a derivation modulo XOR from $T$ and $\pi_{<i}$. (We use $\pi(i)$ to denote the $i$-th atom in $\pi$, and $\pi_{<i}$ to denote those atoms $\pi(j)$ with $j < i$.)

Suppose $\pi(i)$ is obtained using a protocol rule $r_1, \ldots, r_m \rightarrow r_0$ in $T_n$. There exists a substitution $\theta$ with $r_0\theta \sim \pi(i)$ and for each $k \in \{1, \ldots, m\}$, we have $j < i$ such that $r_k\theta \sim \pi(j)$. By Def. 3, there must be a rule $r'_1, \ldots, r'_m \rightarrow r'_0$ in $T$ and a substitution $\sigma$ such that for each $\ell \in \{0, \ldots, m\}$, $r_\ell = r'_\ell\sigma$. Thus for each $k \in \{1, \ldots, m\}$, we have $j < i$ such that $r'_k(\sigma\theta) = (r'_k\sigma)\theta = r_k\theta \sim \pi(j)$. Thus we obtain $r'_0(\sigma\theta) = r_0\theta \sim \pi(i)$ using the rule $r'_1, \ldots, r'_m \rightarrow r'_0$.

**Lemma 2.** *If $\pi$ is a derivation for $f$ from $T$ modulo XOR, then $\ulcorner\pi\urcorner$ is a derivation for $f$ from $T_n^{\oplus}$.*

*Proof.* Let $S$ be the set of session identifiers occurring in $\pi$ and suppose its size is $n$. By Def. 3, we obtain a multi-session theory $T_n$ using $S$. From Lem. 15 in [10], we know if $\pi'$ is a derivation for $f$ from $T_n$ modulo XOR, then $\ulcorner\pi'\urcorner$ is a syntactic derivation for $f$ from $T_n^{\oplus}$. Thus, to prove this lemma, it suffices to prove $\pi$ is also a derivation from $T_n$. Now, we have to prove each $\pi(i)$ is obtained by a derivation modulo XOR from $T_n$ and $\pi_{<i}$.

Suppose $\pi(i)$ is obtained from a rule $r_1, \ldots, r_m \rightarrow r_0$ in $T$. Then there exists a substitution $\theta$ with $r_0\theta = \pi(i)$ such that for each $k \in \{1, \ldots, m\}$, we have $j < i$ and $r_k\sigma = \pi(j)$. The domain of $\theta$ can be divided into two parts; session identifiers $V_1$ and $C$-variables $V_2$. It is clear that there exist two substitutions $\sigma$ and $\theta'$ such that $r_j\theta = r_j\sigma\theta'$ where $dom(\sigma) = V_1$ and $dom(\theta') = V_2$. From Def. 3, there exists a rule $r'_1, \ldots, r'_m \rightarrow r'_0$ in $T_n$ such that for each $\ell \in \{0, \ldots, m\}$ $r'_\ell = r_\ell\sigma$. Thus we obtain $\pi(i) = (r_0\sigma)\theta' = r'_0\theta'$ from $r'_1, \ldots, r'_m \rightarrow r'_0$.

From the above two lemmas, we immediately obtain that $T \vdash_{\oplus} f$ iff $T_n^{\oplus} \vdash f$.

## 4 Optimizations of XOR-reduction

### 4.1 Optimization Based on Nonce Freshness

Recall that a protocol model in a Horn theory $T$ consists of a set of rules $r_i$ ($i \in \{1, \ldots n\}$) of the form $I(t_1), \ldots, I(t_i) \rightarrow I(s_i)$. Such rules should be read as "after receiving the messages $t_1, \ldots, t_i$ the agent sends $s_i$". The terms on both sides may contain $C$-variables to which substitutions are applied in the XOR-reduction process. Consider a rule $r_i$ in which some $t_j$ ($1 < j \leq i$) and $s_i$ contain

a variable $x$. If $r_i$ generates a nonce $m$, substituting $m$ for $x$ may lead to false attacks. For example, applying substitution $\sigma(x) = m(b,a) \oplus x$ to rule (6) gives

$$\{m(b,a) \oplus x, a\}_{pub(sk_b)} \rightarrow \{m(b,a), b \oplus x\}_{pub(sk_a)},$$

indicating a pre-play of the nonce $m(b,a)$ by the adversary, contradicting freshness of nonces. We call rules that are vulnerable to this type of illegal substitutions *challenging rules*. To identify challenging rules we assume a strict total order $\prec$ on protocol rules of a role according to the execution order of the protocol steps, and use $t \sqsubseteq r$ to denote that a term $t$ appears in the Horn clause $r$ (formally $t$ is a subterm of the left-hand side or right-hand side of the rule $r$).

**Definition 4 (Challenging Rule).** *Let $M$ be the set of nonces occurring in a Horn theory and $R = \{r_1, \cdots, r_n\}$ the corresponding set of protocol rules. We say $r_i$ is a challenging rule if there exists $m \in M$ such that $m \sqsubseteq r_i$ and for each $r_j \in R$ such that $r_j \prec r_i$, $m \not\sqsubseteq r_j$.*

We now define which terms in a clause can be cancelled by applying a substitution to them.

**Definition 5 (Cancelling Term Set).** *Let $t$ be a $C$-dominated term and $s \sqsubseteq t$ be a fragile term. We define the* set of cancelling terms *$\mathcal{N}(s,t)$ to be a set of terms such that there exists a substitution for $s$ resulting in cancellation of another subterm of $t$:*

$$\mathcal{N}(s,t) = \{s' | \exists u \text{ s.t. } s \oplus u \oplus s' \sqsubseteq t \vee s' \oplus u \oplus s \sqsubseteq t\}.$$

For example, the cancelling term set $\mathcal{N}(x,t)$ for $t = m(a,b) \oplus x \oplus a$ is $\{m(a,b), a\}$.

Now, let $M$ be a set of nonces that are freshly generated in rule $r$. We can restrict the set of $C$-dominated substitutions for $r$ to substitutions that do not cancel any term with $m \in M$.

**Definition 6 (Legal Substitution).** *Let $t$ be a $C$-dominated term and $M$ be the set of nonces that are freshly generated. Then $\sigma$ is a* legal substitution *for $t$ if it contains all variables $x$ that occur in $t$ and for each $x$ one of the following three cases holds:*

*i.* $\sigma(x) = x$,
*ii.* $x \in \mathcal{F}(t), \sigma(x) = c \oplus x$ *for some* $c \in C^{\oplus}_{norm}, c \neq 0$ *and for each* $m \in M$*, there does not exist* $n \in \mathcal{N}(x,t)$ *such that* $m \sqsubseteq n \wedge n \sqsubseteq c$.
*iii.* *if $x$ occurs in a fragile subterm $s$ and there exists a substitution $\sigma'$ in normal form satisfying $s\sigma' \in C^{\oplus}$ and for each $m \in M$, there does not exist $n \in \mathcal{N}(s,t)$ such that $m \sqsubseteq n \wedge n \sqsubseteq s\sigma'$, then $\sigma(x) = \sigma'(x)$.*

Recall that there are 128 substitutions for clause (17). Clearly, $\mathcal{N}(x,t)$ is $\{m(a,a), a\}$ where $t = m(a,a) \oplus x \oplus a$. Since $m(a,a)$ is fresh in this challenging rule, $M = \{m(a,a)\}$. According to Def. 6, any substitution in cases (ii) and (iii) having $m(a,a)$ as a subterm is not legal. For instance, the substitutions such as $\sigma(x) = m(a,a) \oplus x$ and $\sigma(x) = m(a,a) \oplus n(a,a)$ are removed. Applying this optimization removes 64 rules.

### 4.2 Optimization Based on Session Ordering

The bounded verification that we have introduced in Sect. 3 extends the class of XOR-protocols that can be automatically verified. However, their verification is often inefficient. Recall that the number of rules of an XOR-reduced protocol grows exponentially in the size of the dominating set. Therefore, in particular the verification of protocols that are nonce-$\oplus$-linear but not $\oplus$-linear becomes less efficient if the number of sessions grows. In this section, we aim to reduce the number of rules obtained from the XOR-reduction process by computing a dominating set for each rule with fragile subterms.

We first observe that the session identifiers we introduced in Sect. 3 are only needed to disambiguate nonces from different sessions. They carry no other information and do not appear anywhere else in the protocol specification. We can therefore enforce an order on the challenging rules that create these nonces. In the following we assume that each role of a protocol contains at most one challenging rule, but we note that our theory can be extended to roles with more than one challenging rule.

Let $Cr(s_i)$ be the challenging rule of an agent in session $s_i \in \{s_1, \ldots, s_n\}$. In these sessions, the agent plays the same role and communicates with the same partner as well. We now extend the order $\prec$ introduced in Sect. 4.1 by defining the order between these challenging rules such that $Cr(s_i) \prec Cr(s_j)$ if and only if $i < j$. The main observation for this optimization is that by fixing an order on the execution of the challenging rules, we can eliminate illegal substitutions. In order to do so, we compute a dominating set for each rule having fragile subterms separately. This dominating set only contains nonces that have been generated in previous sessions (based on $\prec$).

As a starting point we take a dominating set $C$ (see Sect. 2.1). We then eliminate terms that contain subterms that are generated in later challenging rules. Let $Nt(Cr)$ denote the set of nonces generated in challenging rule $Cr$. Then the dominating set $C'$ for rule $r$ is defined by the set $C$ from which any term that depends on a nonce that is generated after or in $r$ is eliminated:

$$C'(r) = \{s \in C | \text{there does not exist } n \in \bigcup_{r \prec r'} Nt(r') \cup Nt(r) \text{ s.t. } n \sqsubseteq s\}.$$

With the size of the dominating set decreasing, the number of substitutions decreases as well. Consider an instance of rule (9) in $NSL'_{\oplus}$-auth. Suppose two sessions $s_1$ and $s_2$ in which agent $b$ plays role $B$ and talks to $a$. Let $r_1$ and $r_2$ represent the rules in session $s_1$ and $s_2$ respectively:

$$I(\{x, a\}_{pub(sk_b)}) \rightarrow I(\{m(b, a, s_1), m(b, a, s_1) \oplus x \oplus b\}_{pub(sk_a)}$$
$$I(\{x, a\}_{pub(sk_b)}) \rightarrow I(\{m(b, a, s_2), m(b, a, s_2) \oplus x \oplus b\}_{pub(sk_a)}$$

Since they are both challenging rules with $Nt(r_1) = \{m(b, a, s_1)\}$ and $Nt(r_2) = \{m(b, a, s_2)\}$, and we also have $r_1 \prec r_2$, the dominating set $C'(r_1)$ cannot contain terms with $m(b, a, s_1)$ and $m(b, a, s_2)$ as subterms.

### 4.3 Secrecy-based Authentication Verification

By the result of Comon-Lundh and Cortier [13], we need one more participant to verify authentication than secrecy (see Sect. 2.2). Therefore, Horn theories for verifying authentication are generally bigger than models of the same protocols for verifying secrecy. The situation becomes worse when bounded verification is applied. We propose to optimize verification of authentication properties by first verifying secrecy for certain terms in the Horn theory.

Consider two nonce-$\oplus$-linear Horn theories $T_{sec}$ and $T_{auth}$. Let $F$ be the set of facts that ProVerif will check for their secrecy when deriving the goals in $T_{auth}$. With the results from the secrecy verification for $F$ using $T_{sec}$, we can prevent ProVerif from deriving these facts during authentication verification.

For the sake of efficiency, $F$ should be carefully chosen. Typically, $F$ contains shared keys and $C$-dominated terms. The observation is that by this choice we can eliminate the rules violating secrecy after reduction. For example, for $NSL'_{\oplus}$ after reduction of its two-session model, we have a rule:

$$I(\{n(a,b,s_1),c\}_{pub(sk_b)}) \rightarrow I(\{m(b,c,s_1), m(b,c,s_1) \oplus n(a,b,s_1) \oplus b\}_{pub(sk_c)}.$$

If we know that $n(a,b,s_1)$ is secret, according to this rule and the $\oplus$-rule the intruder can obtain it after decrypting the message and computing the XOR of $m(b,c,s_1) \oplus n(a,b,s_1) \oplus b$ with $m(b,c,s_1)$ and $b$. This contradicts the secrecy of $n(a,b,s_1)$. To identify these rules, we define *secrecy-violating rules*:

**Definition 7 (Secrecy-violating Rule).** *Let $S$ be a set of verified secrets and $r$ be a reduced rule. We say $r$ is a* secrecy-violating rule *if after repeatedly using the intruder rules, the intruder can obtain a secret $t \in S$.*

This optimization concentrates on finding secrecy-violating rules in order to reduce the size of the resulting Horn theory. Therefore, we can improve the efficiency of verification using ProVerif. We only implemented a light-weight process to remove some of the rules automatically. How to remove all such rules is an interesting research topic.

### 4.4 RFID-based Optimizations

Radio frequency identification (RFID) systems are used to identify tagged objects through wireless channels. Since tags must be manufactured at a very low cost, only simple operations can be performed by the tag. Therefore, XOR is an operator that is often used in RFID protocols. Compared to general security protocols, RFID protocols have their own characteristics that allow optimization of the verification process. In this section, we discuss three characteristics and present their corresponding optimizations.

During communications, readers are initiators and they aim to authenticate tags. Tags receive challenges and run the steps described by the protocol. For this reason, an agent can only play one role: an agent is either *reader* or *tag*. This allows us to simplify the Horn theories for verification of authentication. For

instance, assume $NSL'_{\oplus}$ is used as an RFID protocol and let the set of protocol participants be $\{tag, reader, intruder\}$. In rule (8) of $NSL'_{\oplus}$-auth, $a$ can only be substituted by *reader* while $b$ can be substituted by either *tag* or *intruder*.

Since information such as keys is embedded in tags, only the readers of the same system can talk to tags. Moreover, tags always belong to one RFID system. There never exist secrets shared between the intruder and tags. We therefore do not model the intruder as an insider, preventing the derivation of insider-attacks.

In particular, we propose to remove the rules in which *tag* believes to be talking to *intruder*. For example, with the assumption that $NSL'_{\oplus}$ is an RFID protocol, in rule (9), we have $a \in \{tag\}, b \in \{reader\}$. In this way, we decrease the number of Horn clauses in the model. In particular, the size of dominating set will be smaller as a number of nonces is removed.

We observe that tags are manufactured in such a way that they can only have one active protocol execution at a time. Therefore, we do not have to model attacks that rely on a parallel execution of two or more runs of one tag. Hence, a tag's runs are completely sequential. For bounded verification, the order $\prec$ introduced in Sect. 4.1 can be extended to all rules of the tag. Suppose there are $\ell$ rules in a session and $n$ sessions are modeled in total. Let $r(i, s_k)$ be the rule that represents the $i$th step of the tag in session $s_k \in \{s_1, \ldots, s_n\}$. Given $i, j \leq \ell, k_1, k_2 \leq n$, we have $r(i, s_{k_1}) \prec r(j, s_{k_2})$ if $(i < j \wedge k_1 = k_2) \vee (k_1 < k_2)$. Now, the optimization in Sect. 4.2 can be applied to the simplified models with the strict order on the tag's rules.

### 4.5 Optimization Based on $\oplus$-rule Reduction

In the implementation of XorProverif, Küsters and Truderung introduce a compact way to represent clauses (13)-(16). They do not keep all the copies for every pair $c, c' \in C^{\oplus}_{norm}$, but rather introduce a function $xtab(c, c', \ulcorner c \oplus c' \urcorner)$ to denote clauses of the form of (13). The Horn clauses (14)-(16) are represented below:

$$xarg(x), I(x), I(y) \rightarrow I(x \oplus y)) \tag{18}$$

$$xarg(x), I(x \oplus y), I(x) \rightarrow I(y) \tag{19}$$

$$xtab(x, y, z), I(x \oplus t), I(y) \rightarrow I(z \oplus t) \tag{20}$$

$$xtab(x, y, z), I(x \oplus t), I(y \oplus t) \rightarrow I(z) \tag{21}$$

where $xarg(x)$ denotes $x \in C^{\oplus}_{norm}$ in the first two clauses and $x, y, z$ are variables in the last two. When instantiating rule (20) with the substitution $\{a/x, b/y, \ulcorner a \oplus b \urcorner / z\}$, we have $xtab(a, b, \ulcorner a \oplus b \urcorner), I(a \oplus t), I(b) \rightarrow I(\ulcorner a \oplus b \urcorner \oplus t)$. Similarly, for substitution $\{b/x, a/y, \ulcorner a \oplus b \urcorner / z\}$ we have $xtab(b, a, \ulcorner a \oplus b \urcorner), I(b \oplus t), I(a) \rightarrow I(\ulcorner a \oplus b \urcorner \oplus t)$. As shown by this example, rule (20) requires both $xtab(a, b, \ulcorner a \oplus b \urcorner)$ and $xtab(b, a, \ulcorner a \oplus b \urcorner)$ existing in the Horn theory to capture both scenarios. By introducing the following symmetric clause to rule (20)

$$xtab(x, y, z), I(y \oplus t), I(x) \rightarrow I(z \oplus t)$$

we can remove $xtab(b, a, \ulcorner a \oplus b \urcorner)$ as long as $xtab(a, b, \ulcorner a \oplus b \urcorner)$ remains in the Horn theory in the previous example, since the second substitution is captured

by the newly introduced clause. In this way, we can remove rules of the form $xtab(a, b, \ulcorner a \oplus b \urcorner)$. With the size of the dominating set $C^{\oplus}_{norm}$ becoming larger, the number of reduced rules also becomes larger.

## 5 Implementation and Experiments

In order to validate our ideas, we have built an implementation [14] of the bounded verification (as descried in Sect. 3) and the optimizations (as described in Sect. 4). In order to check the effects of our improvements, we have compared our implementation with that of XorProverif.

### 5.1 Implementation

We use SWI prolog for our implementation. The input Horn theory consists of three parts: (1) declaration of function symbols that are used in the theory, (2) necessary initial intruder facts, intruder rules, and protocol rules, (3) verification goals, either secrecy or authentication. We introduce a function *nonce* to declare nonces, and an auxiliary function to provide necessary information about a protocol rule including its position and session. The latter is needed in order to implement optimizations in Sect. 4.

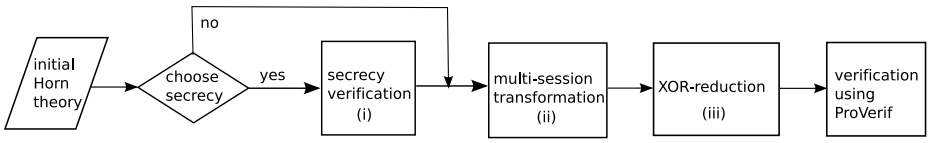


Fig. 2: Structure of the implementation.

As shown in Fig. 2, our implementation mainly performs three steps. Each step takes the output of its previous step as the input Horn theory and outputs a new Horn theory after. The input Horn theory at the very beginning must be nonce-$\oplus$-linear. Step (i) is optional. It can choose a set of terms to check if they are secret, and the results of the secrecy verification are added to the output. Step (ii) transforms its input into a multi-session $\oplus$-linear model, which is necessary for bounded verification (see Sect. 3). Step (iii) checks $\oplus$-linearity and computes $C$-dominating sets as done by XorProverif. It also applies optimizations as described in Sect. 4 whenever possible and reduces the Horn theory to the XOR-free one. In the end, ProVerif performs the last part of the verification.

### 5.2 Experiments

We first present experimental results for secrecy verification with optimizations applied to the XOR-reduction step and compare them with XorProverif (see

Tab. 1). Then we apply bounded verification to a number of nonce-$\oplus$-linear protocols including some RFID protocols to check authentication (see Tab. 2). All experiments are performed on a Dell Latitude E5500 laptop with a 2.26GHz Intel Core$^{\mathbf{TM}}$ 2 Duo P8400 processor and 2GB RAM.

**Secrecy verification.** We first describe the protocols in that we use for our experiments.

The first protocol we consider is our running example $NSL'_{\oplus}$-sec. We propose two fixes to the protocol that counter the attack depicted in Sect. 2.2. In $NSL'_{\oplus}$-fix-0, we replace the message $\{m, m \oplus n \oplus b\}_{pub(sk_a)}$ with $\{m \oplus n, b\}_{pub(sk_a)}$ and in $NSL'_{\oplus}$-fix-1 with $\{m, h(m \oplus n) \oplus b\}_{pub(sk_a)}$, where $h$ denotes a hash function. Note that these protocols are only meant to fix the secrecy flaw.

The protocol $NSL_{\oplus}$ is the example used by Küsters and Truderung [10] where the second message is of the form $\{m, n \oplus b\}_{pub(sk_a)}$. CCA is short for Common Cryptographic Architecture (CCA) API [15], designed by IBM. This series of CCA protocols are also checked by Küsters and Truderung [10].

Inspired by Millen's ffgg protocol [16], we design a family of protocols which we call *fgms*. The family contains protocols that can be attacked in $n$ sessions, but not in $n-1$ sessions, for any $n$. In order to attack the secrecy claim, the algebraic properties of $\oplus$ need to be used.

The specification of fgms-2, the protocol that can be attacked in two sessions but not in one, is as follows. Role $A$ and $B$ initially share a secret $k$. An agent in role $A$ initiates the protocol by sending $\{n_a, k\}_k$ to $B$. The agent playing role $B$ does not verify the values of $n_a$ and $k$ inside the encryption, but only the encryption key $k$. He then generates a nonce $n_b$ and replies with $\langle x, n_b, \{n_b \oplus y, x\}_k\rangle$. The protocol is shown in Fig. 3.

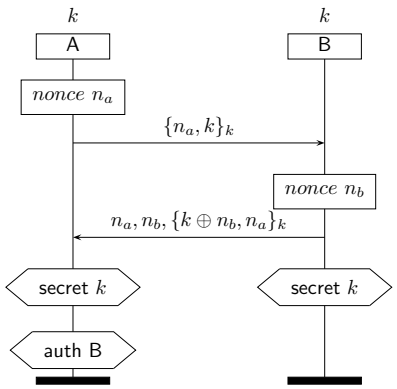


Fig. 3: Description of the fgms-2 protocol.

We can obtain the protocol fgms-3 by adding an extra nonce to both messages. The first message is replaced by $\{n_a, n'_a, k\}_k$ and the second message by $n_a, n_b, \{n'_a \oplus n_b, k, n_a\}_k$. In a similar way fgms-$n$ for any $n > 2$ can be designed.

Tab. 1 gives the reduction time required by XorProverif (referred to as 'XPv') and our implementation (referred to as 'optimized'), and the ProVerif verification time with and without our optimizations. From the results, we observe a big improvement for $NSL'_{\oplus}$ and its fixes, if our optimization for secrecy is applied. For the CCA protocols, due to the optimization in Sect. 4.5, the analysis also becomes more efficient. For the fgms family of protocols, without our optimization ProVerif cannot terminate.

Table 1: Results for secrecy verification (n.t. for non-terminating).

| XOR-protocols | correct | reduction | | ProVerif time | | saved |
|---|---|---|---|---|---|---|
| | | XPv | optimized | - opt. | + opt. | |
| $NSL'_{\oplus}$-sec | no | 0.67s | 0.52s | 16.12s | 7.16s | 55.6% |
| $NSL'_{\oplus}$-fix-0 | yes | 0.13s | 0.12s | 0.14s | 0.08s | 42.9% |
| $NSL'_{\oplus}$-fix-1 | yes | 0.71s | 0.53s | 14.95s | 6.60s | 55.9% |
| $NSL_{\oplus}$ | no | 0.07s | 0.07s | 0.02s | 0.01s | 50% |
| CCA-0 | no | 0.24s | 0.22s | 129s | 117s | 9.3% |
| CCA-1A | yes | 0.09s | 0.09s | 0.69s | 0.64s | 7.2% |
| CCA-1B | yes | 0.12s | 0.11s | 1.17s | 1.11s | 5.1% |
| CCA-2B | yes | 0.20s | 0.18s | 12.7s | 10.4s | 18.1% |
| CCA-2C | yes | 0.25s | 0.22s | 69.60s | 64.34s | 7.6% |
| CCA-2E | yes | 0.09s | 0.09s | 1.48s | 1.34s | 9.5% |
| fgms-2 | no | 0.06s | 0.06s | n.t. | 0.21s | - |
| fgms-3 | no | 0.07s | 0.07s | n.t. | 0.37s | - |
| fgms-4 | no | 0.07s | 0.07s | n.t. | 0.40s | - |
| fgms-5 | no | 0.08s | 0.08s | n.t. | 0.51s | - |

**Bounded verification of authentication properties.** For the analysis of our verification method for authentication we use the following protocols.

The protocols containing $NSL'_{\oplus}$ in their names include our running example and one of its fixes. Lee et al. [17] and Song and Mitchell [18] proposed RFID protocols, which we call LAK06 and SM08 after the last names of the authors. Attacks on both protocols have been reported by Van Deursen and Radomirović [19]. We also analyze a variant of the protocol by Choi et al. [20] (CLL09).

Our final example is the mutual RFID authentication protocol proposed by Cai et al. [21], which is depicted in Fig. 4. In order to comply with the EPCglobal C1G2 specification, the protocol only uses a 16-bit Pseudo-Random Number Generator (PRNG) and a 16-bit Cyclic Redundancy Check (CRC). The reader $R$ and tag $T$ share secrets $TID$ (*Tag Identifier*) and $PWA$ (*Access Password*).

The reader starts by sending a query and a nonce $R_r$. The tag generates a nonce $R_t$ and computes the XOR of $PWA$ and the concatenation of $M_\ell$ and $M_h$, as given in Fig. 4.

The reader checks the the correctness of the received message before sending the response. Burmester et al. give two attacks on the protocol [22], which both rely on the homomorphic properties of CRC functions.

Using our prototype, we find a new attack on tag authentication using bounded verification. To impersonate a tag the intruder proceeds as follows. He challenges the tag with any nonce $R_e$ and obtains the reply $\langle R_t, (CRC(TID_l \oplus R_e \oplus R_t) \parallel CRC(TID_h \oplus R_e \oplus R_t)) \oplus PWA\rangle$. This message suffices for the intruder to respond to any reader challenge $R_r$ by replacing $R_t$ in the message with $R_e \oplus R_r \oplus R_t$. The attack is depicted in Fig. 5.

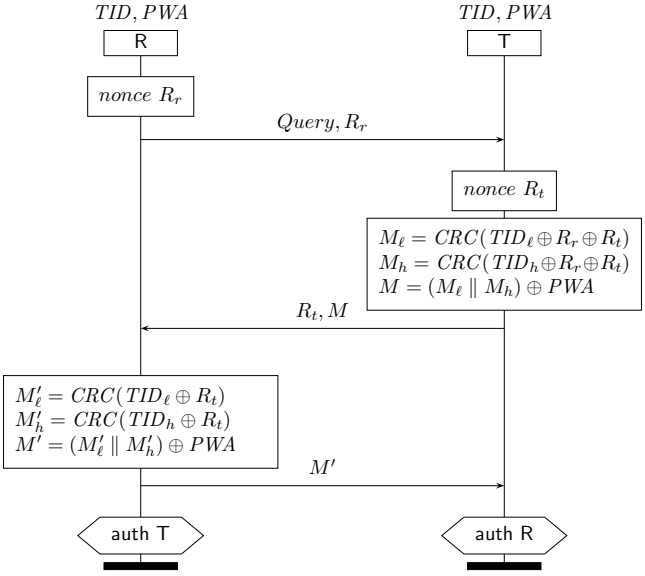


Fig. 4: Description of the CZW08 protocol.

Tab. 2 gives the number of sessions (#sid) used for multi-session transformation, the time used for our optimized XOR-reduction, the verification time taken by ProVerif after the multi-session transformation (without the optimizations) and our bounded verification with optimizations, and the number of generated derivations (#derivations). For general protocols, we apply the optimization in Sect. 4.2. For RFID protocols, the optimization in Sect. 4.4 is also applied. The table clearly shows that our optimizations can reduce both the verification time by ProVerif, and the number of derivations.

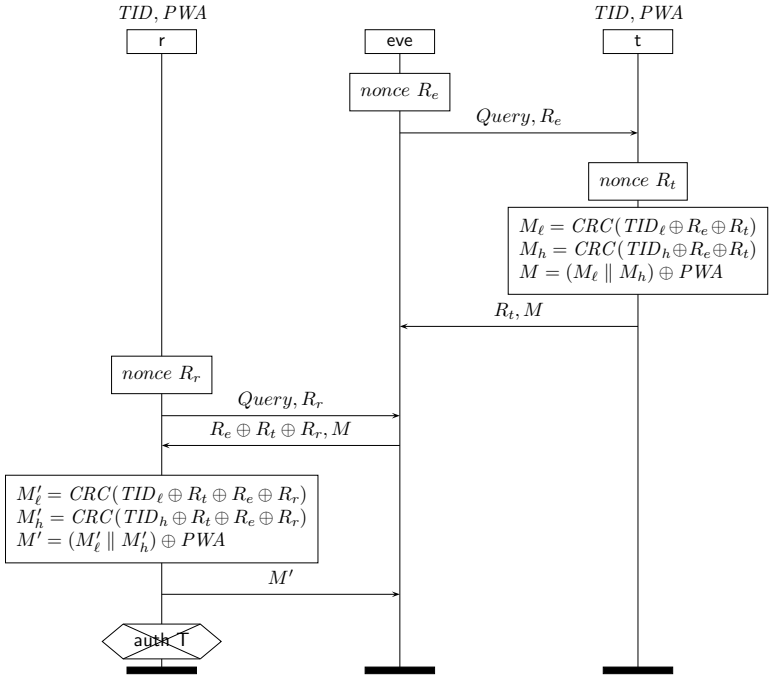


Fig. 5: An attack on the CZW08 protocol.

Table 2: Results of bounded verification of authentication.

| XOR-protocols | correct | #sid | reduction | ProVerif time | | saved | #derivations | |
|---|---|---|---|---|---|---|---|---|
| | | | | - opt. | + opt. | | - opt. | + opt. |
| $NSL'_{\oplus}$-authA | no | 1 | 4.47s | 17.67s | 7.39s | 58.2% | 2 | 1 |
| $NSL'_{\oplus}$-authA-fix-0 | yes | 1 | 6.50s | 0.132 | 0.072s | 45.5% | 1 | 1 |
| $NSL'_{\oplus}$-authA-fix-0 | yes | 2 | 97.2s | 6916s | 2907s | 58.0% | 2 | 2 |
| $NSL'_{\oplus}$-authB-fix-0 | no | 1 | 3.01s | 0.32s | 0.08s | 75.0% | 1 | 1 |
| LAK06 | no | 1 | 0.152s | 0.012s | 0.004s | 66.7% | 8 | 4 |
| SM08 | no | 1 | 0.128s | 0.036s | 0.016s | 55.6% | 8 | 4 |
| CLL09 | yes | 1 | 0.068s | 0.124s | 0.064s | 48.4% | 13 | 5 |
| CLL09 | no | 2 | 0.62s | 244.4s | 139.4s | 42.9% | 156 | 14 |
| CZW08 | no | 1 | 0.17s | 0.064s | 0.028s | 56.2% | 8 | 4 |

## 6 Conclusion and Future Work

In this paper, we have focused on the verification of security protocols with XOR. We improve the XOR-reduction approach of Küsters and Truderung [10] for the verification of XOR-protocols modeled by Horn theories.

First, we extend their approach for authentication verification to a richer class of XOR protocols using the idea of bounded verification. We consider session identifiers as constants instead of variables [10] and verify protocols using models with a bounded number of sessions. The corresponding transformation process is performed automatically.

Second, we make their approach more efficient by developing a number of dedicated optimizations including the usage of freshness of generated nonces and secrecy of certain terms to reduce the number of substitutions, restricting session order in our bounded verification, and exploring the specific characteristics of RFID protocols. All these ideas have been implemented in a prototype. The experimental results show the feasibility of our methods and the reduction in verification time by ProVerif looks in all respects promising. We also found a new attack on a recently proposed RFID protocol.

We conjecture that our optimizations presented in the current paper do not sacrifice the soundness of Küsters and Truderung's approach. However, their formal correctness proofs are left for the future. There are several ways to proceed. Our implementation is still preliminary, we want to improve it and test it with more experiments. Especially we are interested in bigger examples. We want to extend our work by identifying more optimizations. Küsters and Truderung have extended their reduction approach to protocols with Diffie-Hellman exponentiation [23]. It will be interesting to see to what extent our optimizations can be applied to those protocols as well.

*Acknowledgement.* We thank the anonymous referees for their valuable comments.

## References

1. Lowe, G.: Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In: Proc. 2nd Workshop on Tools and Algorithms for the Construction and Analysis of Systems. Volume 1055 of Lecture Notes in Computer Science., Springer (1996) 147–166
2. Dolev, D., Yao, A.C.C.: On the security of public key protocols. IEEE Transactions on Information Theory **29**(2) (1983) 198–207
3. Blanchet, B.: An efficient cryptographic protocol verifier based on prolog rules. In: Proc. 14th IEEE Computer Security Foundations Workshop, IEEE Computer Society (2001) 82–96
4. Cremers, C., Lafourcade, P., Nadeau, P.: Comparing state spaces in automatic protocol analysis. In: Formal to Practical Security. Volume 5458 of Lecture Notes in Computer Science., Springer (2009) 70–94
5. Cortier, V., Delaune, S., Lafourcade, P.: A survey of algebraic properties used in cryptographic protocols. Journal of Computer Security **14**(1) (2006) 1–43

6. Comon-Lundh, H., Shmatikov, V.: Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. In: Proc. 8th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society (2003) 271–280
7. Comon-Lundh, H., Delaune, S.: The finite variant property: How to get rid of some algebraic properties. In: Proc. 16th Conference Term Rewriting and Applications. Volume 3467 of Lecture Notes in Computer Science., Springer (2005) 294–307
8. Cortier, V., Keighren, G., Steel, G.: Automatic analysis of the security of XOR-based key management schemes. In: Proc. 13th Conference on Tools and Algorithms for the Construction and Analysis of Systems. Volume 4424 of Lecture Notes in Computer Science., Springer (2007) 538–552
9. Lowe, G.: Casper: A compiler for the analysis of security protocols. In: Proc. 10th Computer Security Foundations Workshop, IEEE Computer Society (1997) 18–30
10. Küsters, R., Truderung, T.: Reducing protocol analysis with XOR to the XOR-free case in the horn theory based approach. In: Proc. 15th ACM Conference on Computer and Communications Security, ACM Press (2008) 129–138
11. Blanchet, B.: Automatic verification of correspondences for security protocols. Journal of Computer Security **17**(4) (2009) 363–434
12. Needham, R.M., Schroeder, M.D.: Using encryption for authentication in large networks of computers. Communications of the ACM **21**(12) (1978) 993–999
13. Comon-Lundh, H., Cortier, V.: Security properties: two agents are sufficient. Science of Computer Programming **50**(1-3) (2004) 51–71
14. Chen, X., van Deursen, T., Pang, J.: Improving automatic verification of protocols with XOR (implementation) (2009) Available at `http://satoss.uni.lu/software/`.
15. International Business Machines Corporation: CCA basic services reference and guide. (2003) Available at `http://www-306.ibm.com/security/cryptocards/pdfs/CCA_Basic_Services_241_Revised_20030918.pdf`.
16. Millen, J.K.: A necessarily parallel attack. In: Proc. Workshop on Formal Methods and Security Protocols. (1999)
17. Lee, S., Asano, T., Kim, K.: RFID mutual authentication scheme based on synchronized secret information. In: Proc. Symposium on Cryptography and Information Security. (2006)
18. Song, B., Mitchell, C.J.: RFID authentication protocol for low-cost tags. In: Proc. 2nd ACM Conference on Wireless Network Security, ACM Press (2008) 140–147
19. van Deursen, T., Radomirović, S.: Algebraic attacks on RFID protocols. In: Proc. 3rd Workshop in Information Security Theory and Practices: Smart Devices, Pervasive Systems, and Ubiquitous Networks. Volume 5746 of Lecture Notes in Computer Science., Springer (2009) 38–51
20. Choi, E.Y., Lee, D.H., Lim, J.I.: Anti-cloning protocol suitable to EPCglobal class-1 generation-2 RFID systems. Computer Standards & Interfaces (2009) In press.
21. Cai, Q., Zhan, Y., Wang, Y.: A minimalist mutual authentication protocol for RFID system and BAN logic analysis. In: Proc. ISECS Colloquium on Computing, Communication, Control and Management. (2008) 449–453
22. Burmester, M., Medeiros, B., Munilla, J., Peinado, A.: Secure EPC gen2 compliant radio frequency identification (2009) Available at `http://eprint.iacr.org/`.
23. Küsters, R., Truderung, T.: Using ProVerif to analyze protocols with Diffie-Hellman exponentiation. In: Proc. 22th IEEE Computer Security Foundations Symposium, IEEE Computer Society (2009) 157–171