Formal Verification of e-Auction protocols

Jannik Dreier, Pascal Lafourcade, and Yassine Lakhnech

Université Grenoble 1, CNRS, Verimag, FRANCE
firstname.lastname@imag.fr

Abstract. Auctions have a long history, having been recorded as early as 500 B.C.. With the rise of Internet, electronic auctions have been a great success and are increasingly used. Many cryptographic protocols have been proposed to address the various security requirements of these electronic transactions. We propose a formal framework to analyze and verify security properties of e-Auction protocols. We model protocols in the Applied Pi-Calculus and define privacy notions, which include secrecy of bids, anonymity of the participants, receipt-freeness and coercion-resistance. We also discuss fairness, non-repudiation and non-cancellation. Additionally we show on two case studies how these properties can be verified automatically using ProVerif, and discover several attacks.

1 Introduction

Auctions are a simple method to sell goods and services. Typically a *seller* offers a good or a service, and the *bidders* make offers. Depending on the type of auction, the offers might be sent using sealed envelopes which are opened at the same time to determine the winner (the "sealed-bid" auction), or an *auctioneer* could announce prices decreasingly until one bidder is willing to pay the announced price (the "dutch auction"). Additionally there might be several rounds, or offers might be announced publicly directly (the "English" or "shout-out" auction). The winner usually is the bidder submitting the highest bid, but in some cases he might only have to pay the second highest offer as a price (the "second-price"- or "Vickrey"-Auction). In general a bidder wants to win the auction at the lowest possible price, and the seller wants to sell his good at the highest possible price. For more information on different auction methods see [**?**].

Depending on the type of auction and the application different security properties might be interesting to realize in auction protocol. We identify the following main security properties of auction protocols:

- **Fairness:** We propose the two following fairness properties: Firstly a fair auction protocol should not *leak* any information about the other participants and their offers until the bidding phase is over (so as to prohibit unfair tactics based on leaked information). Secondly it should not allow anybody to execute a *"default winning strategy"*, i.e. a strategy that allows a malicious participant to win at a chosen price independently from the other bidders' offers.
- **Authentication**: For the seller it is crucial to ensure *Non-Repudiation*, i.e. that – after the winner has been announced – the winning bidder cannot claim that he did not submit the winning bid. Additionally we might want to ensure *Non-Cancellation*, i.e. that a bidder cannot cancel a submitted offer before the winner is announced.

– **Privacy:** We distinguish several different notions: *Secrecy of Bids*, *Anonymity of Bidders*, *Receipt-Freeness* and *Coercion-Resistance*. Secrecy of Bids guarantees that the losing bids remain secret, or at least cannot be linked to the participants. Anonymity of Bidders means that the participants, in particular the winner, remains anonymous. Receipt-Freeness ensures that bidders are unable to prove to an attacker (which might be another bidder trying to force them to submit a low bid so that he wins) that they bid a certain offer, and Coercion-Resistance means that even when interacting with a coercer, the bidders can still bid a price of their choice.
– **Verifiability**: A verifiable protocol should allow the bidders to verify that the winner was correctly determined, in particular if they lost. Additionally it might be desirable to give the bidders the ability to contest if they think that their offers was not taken into account correctly.

Related Work. Many electronic auction (e-Auction) protocols have been proposed in the literature (see e.g. [**?**,**?**,**?**,**?**] for an overview). As case studies, we use the protocol by Curtis et al. [**?**], which uses a trusted registrar and pseudonyms, and the protocol by Brandt [**?**], which is entirely distributed using secure multi-party computation.

Although there has been much work on developing auction protocols, there is considerably less work on their formal analysis. Subramanian [**?**] proposed an auction protocol and analyzed it using a BAN-style logic to show some security properties. In particular he showed the atomicity of the transaction, weak secrecy of private keys and a form of anonymity modeled as weak secrecy of the public key of the bidder. More recently Dong et al. [**?**] analyzed a receipt-free auction protocol in the Applied π-Calculus. They only considered privacy, in particular secrecy of the bidding price and receipt-freeness, but only for losing bidders.

In the context of electronic voting there has been much more work on formal verification, in particular in the area of privacy [**?**,**?**,**?**,**?**,**?**,**?**,**?**]. Some notions are similar, yet there are some fundamental differences to auctions: In the case of voting the published result is the sum of all votes, hence there is a certain leakage of information about all voters. For example if a candidate received no votes at all, this increases the attackers knowledge about the voters' votes as he can exclude this previously possible option. Yet ideally there should always be some uncertainty about the votes, i.e. no voter's privacy should entirely compromised (apart from pathological cases such as an unanimous vote). In the case of auctions, the public outcome is the winning bid(er), who loses all privacy. In some cases he might stay anonymous, e.g. the well known "bidder on the phone", but at least the winning price will be public. The other bid(er)s however can remain completely private/anonymous – we only know that the offers are inferior. Fairness also is a requirement in electronic voting as well as e-Auctions, but properties such as Non-Repudiation and Non-Cancellation are specific to e-Auctions.

There has been a lot of work on Non-Repudiation in the context of contract signing protocols (e.g. [**?**,**?**]). We rely on the work by Klay et al. [**?**] who propose many different flavors of non-repudiation based on agent knowledge or authentication. We only consider "Non-Repudiation of Origin", i.e. that the bidder cannot deny that he made an offer, implemented as a form of authentication.

Contributions. We provide the following main contributions:

- We give a formal framework in the Applied π-Calculus [**?**] to model and analyze e-Auction protocols.
- We define the discussed fairness, privacy and authentication properties in our model and analyze their relationship.
- We provide two case studies: The protocol by Curtis et al. [**?**] and a protocol by Brandt [**?**]. We show how both can be modeled in the Applied π-Calculus and verified using Proverif [**?**,**?**,**?**]. We discover several flaws on these protocols and explain how some of their shortcomings can be addressed.

Due to the space limitations we cannot give the full proofs here, they are available in [**?**], and the ProVerif code used in the case studies is available in [**?**].

Outline. In Section 2, we recall the Applied π-Calculus and model auction protocols. In Section 3, we formally define the security properties. In Section 4, we analyze two protocols in our model before concluding in the last section.

2 Preliminaries

We recall the Applied π-Calculus and introduce our model of auction protocols.

2.1 Applied π-Calculus

The Applied π-Calculus [**?**] is a formal language to describe concurrent processes. The calculus consists of *names* (which typically correspond to data or channels), *variables*, and a *signature* Σ of *function symbols* which can be used to build *terms*. Functions typically include encryption and decryption – for example $\mathtt{enc}(message, key)$, $\mathtt{dec}(message, key)$ – hashing, signing etc. Terms are correct (i.e. respecting arity and sorts) combinations of names and functions. We distinguish the type "channel" from other *base* types. To model equalities we use an equational theory E which defines a relation $=_E$. A classical example which describes the correctness of symmetric encryption is $\mathtt{dec}(\mathtt{enc}(message, key), key) =_E message$. Processes are constructed using the grammars detailed in Figure 1.

The substitution $\{^M/_x\}$ replaces the variable x with term M. We denote by $fv(A)$, $bv(A)$, $fn(A)$, $bn(A)$ the free variables, bound variables, free names or bound names respectively. A process is *closed* if all variables are bound or defined by an active substitution. The *frame* $\Phi(A)$ of an active process A is obtained by replacing all plain processes in A by 0. This frame can be seen as a representation of what is statically known to the exterior about a process. The domain $\mathtt{dom}(\Phi)$ of a frame Φ is the set of variables for which Φ defines a substitution. An evaluation context $C[\_]$ denotes an active process with a hole for an active process that is not under replication, a conditional, an input or an output. In the rest of the paper we use the following usual notions of equivalence and bisimilarity based on the original semantics [**?**].

$P, Q, R :=$	plain processes
0	null process
$P \mid Q$	parallel composition
$!P$	replication
$\nu n.P$	name restriction ("new")
$\texttt{if } M = N \texttt{ then } P \texttt{ else } Q$	conditional
$\texttt{in}(u, x).P$	message input
$\texttt{out}(u, x).P$	message output

(a) Plain process

$A, B, C :=$	active processes
P	plain process
$A \mid B$	parallel composition
$\nu n.A$	name restriction
$\nu x.A$	variable restriction
$\{^M/_x\}$	active substitution

(b) Extended process

Fig. 1: Grammars for *plain* and *extended* or *active processes*

Definition 1 (Equivalence in a Frame [?]). *Two terms M and N are equal in the frame ϕ, written $(M = N)\phi$, if and only if $\phi \equiv \nu\tilde{n}.\sigma$, $M\sigma = N\sigma$, and $\{\tilde{n}\} \cap (fn(M) \cup fn(N)) = \emptyset$ for some names $\tilde{n}$ and some substitution σ.*

Definition 2 (Static Equivalence ($\approx_s$) [?]). *Two closed frames ϕ and ψ are statically equivalent, written $\phi \approx_s \psi$, when dom(ϕ) = dom(ψ) and when for all terms M and N we have $(M = N)\phi$ if and only if $(M = N)\psi$. Two extended processes A and B are statically equivalent ($A \approx_s B$) if their frames are statically equivalent.*

The intuition behind this definition is that two processes are statically equivalent if the messages exchanged with the environment cannot be distinguished by an attacker (i.e. all operations on both sides give the same results). This idea can be extended to *labeled bisimilarity*.

Definition 3 (Labeled Bisimilarity ($\approx_l$) [?]). *Labeled bisimilarity is the largest symmetric relation $\mathcal{R}$ on closed active processes, such that $A \mathcal{R} B$ implies:*

1. *$A \approx_s B$,*
2. *if $A \rightarrow A'$, then $B \rightarrow^* B'$ and $A' \mathcal{R} B'$ for some B',*
3. *if $A \xrightarrow{\alpha} A'$ and $fv(\alpha) \subseteq$ dom(A) and $bn(\alpha) \cap fn(B) = \emptyset$, then $B \rightarrow^* \xrightarrow{\alpha} \rightarrow^* B'$ and $A' \mathcal{R} B'$ for some B'.*

In this case each interaction on one side can be simulated by the other side, and the processes are statically equivalent at each step during the execution, thus an attacker cannot distinguish both sides.

2.2 Modeling Auction Protocols

We model auction protocols in the Applied π-Calculus as follows.

Definition 4 (Auction Protocol). *An auction protocol is defined by a tuple $(B, S, A_1, \ldots, A_m, \tilde{n})$ where B is the process that is executed by the bidders, S is the process executed by the seller, and the A_j's are the processes executed by the authorities (for example an auctioneer, a registrar etc.), and $\tilde{n}$ is a set of private channels. We also assume the existence of a particular public channel res that is only used to publish the winning bid(er).*

Note that we have only one process for the bidders. This means that different bidders will execute the same process, but with different variable values (e.g. the keys, the bids etc.). To reason about privacy, we talk about instances of an auction protocol, which we call *auction processes*.

Definition 5 (Auction Process). *An instance of an auction protocol* $(B, S, A_1, \ldots, A_m, \tilde{n})$ *is called an auction process, which is a closed process*

$$\nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_1}|\ldots|B\sigma_{id_k}\sigma_{b_k}|S|A_1|\ldots|A_l),$$

where $l \leq m$, $\tilde{n}'$ *includes the secret channel names* $\tilde{n}$, $B\sigma_{id_i}\sigma_{b_i}$ *are the processes executed by the* k *bidders,* σ_{id_i} *is a substitution assigning the identity to the* i*-th bidder (this determines for example the secret keys),* σ_{b_i} *specifies the* i*-th bid and* A_j*'s are the auction authorities which are required to be honest.*

In our definitions we use the context $AP'[\cdot]$ *which allows us to reason about bidders inside the auction process* AP*, for example if we want to explicit bidders* l *and* o*, we rewrite* AP *as* $AP'[B\sigma_{id_l}\sigma_{b_l}|B\sigma_{id_o}\sigma_{b_o}]$.

The restricted channel names model private channels. Note that we only model the honest authorities as unspecified parties are subsumed by the attacker.

By abuse of notation we write $b_l > b_o$ to express that the bidding price determined by the substitution σ_{b_l} is greater than the one assigned by σ_{b_o}, and $\max_i\{b_i\}$ denotes the maximal price assigned by any substitution σ_{b_i}.

In order to reason about reachability and authentication properties we will use *events*. Events are annotations, hence we extend the above plain process grammar as follows: $P = \texttt{event}\ e(M_1, \ldots, M_n).P$ where e is the name of the event, and the terms $M_1, \ldots, M_n$ are parameters. These events do not change the behavior of the processes, but allow us to verify properties such as "event `bad` is unreachable" or "on every trace event `a` is preceded by event `b`". In our definitions we will use the following events:

- `bid(p,id)`: When a bidder `id` bids the price `p` the event `bid(p,id)` is emitted.
- `recBid(p,id)`: When a bid at price `p` by bidder `id` is recorded by the auctioneer/bulletin board[1]/etc. the event `recBid(p,id)` is called. This will be used to model Non-Cancellation, i.e. from this point on a bid is considered binding.
- `won(p,id)`: When a bidder `id` wins the auction at price `p`, the event `won(p,id)` is emitted.

For some of our definitions we need the following two transformations. The first one turns a process P into another process P^{ch} that reveals all its inputs and secret data on the channel ch.

Definition 6 (Process P^{ch} [?]). *Let* P *be a plain process and* ch *be a channel name.* P^{ch} *is defined as follows:*

- $0^{ch} \mathrel{\hat{=}} 0$,
- $(P|Q)^{ch} \mathrel{\hat{=}} P^{ch}|Q^{ch}$,

[1] A bulletin board is a central append-only noticeboard that is often used for communication in protocols.

- $(\nu n.P)^{ch} \mathrel{\hat{=}} \nu n.\mathtt{out}(ch, n).P^{ch}$ *if* n *is a name of base type,* $(\nu n.P)^{ch} \mathrel{\hat{=}} \nu n.P^{ch}$ *otherwise,*
- $(\mathtt{in}(u, x).P)^{ch} \mathrel{\hat{=}} \mathtt{in}(u, x).\mathtt{out}(ch, x).P^{ch}$ *if* x *is a variable of base type,* $(\mathtt{in}(u, x).P)^{ch} \mathrel{\hat{=}} \mathtt{in}(u, x).P^{ch}$ *otherwise,*
- $(\mathtt{out}(u, M).P)^{ch} \mathrel{\hat{=}} \mathtt{out}(u, M).P^{ch}$,
- $(!P)^{ch} \mathrel{\hat{=}} !P^{ch}$,
- $(\mathtt{if}\ M = N\ \mathtt{then}\ P\ \mathtt{else}\ Q)^{ch} \mathrel{\hat{=}} \mathtt{if}\ M = N\ \mathtt{then}\ P^{ch}\ \mathtt{else}\ Q^{ch}$.

In the remainder we assume that $ch \notin fn(P) \cup bn(P)$ before applying the transformation. The second transformation does not only reveal the secret data, but also takes orders from an outsider before sending a message or branching.

Definition 7 (Process P^{c_1,c_2} [?]). *Let P be a plain process and c_1, c_2 be channel names. P^{c_1,c_2} is defined as follows:*

- $0^{c_1,c_2} \mathrel{\hat{=}} 0$,
- $(P|Q)^{c_1,c_2} \mathrel{\hat{=}} P^{c_1,c_2}|Q^{c_1,c_2}$,
- $(\nu n.P)^{c_1,c_2} \mathrel{\hat{=}} \nu n.\mathtt{out}(c_1, n).P^{c_1,c_2}$ *if* n *is a name of base type,* $(\nu n.P)^{c_1,c_2} \mathrel{\hat{=}} \nu n.P^{c_1,c_2}$ *otherwise,*
- $(\mathtt{in}(u, x).P)^{c_1,c_2} \mathrel{\hat{=}} \mathtt{in}(u, x).\mathtt{out}(c_1, x).P^{c_1,c_2}$ *if* x *is a variable of base type,* $(\mathtt{in}(u, x).P)^{c_1,c_2} \mathrel{\hat{=}} \mathtt{in}(u, x).P^{c_1,c_2}$ *otherwise,*
- $(\mathtt{out}(u, M).P)^{c_1,c_2} \mathrel{\hat{=}} \mathtt{in}(c_2, x).\mathtt{out}(u, x).P^{c_1,c_2}$ *where* x *is a fresh variable,*
- $(!P)^{c_1,c_2} \mathrel{\hat{=}} !P^{c_1,c_2}$,
- $(\mathtt{if}\ M = N\ \mathtt{then}\ P\ \mathtt{else}\ Q)^{c_1,c_2} \mathrel{\hat{=}} \mathtt{in}(c_2, x).\mathtt{if}\ x = \mathit{true}\ \mathtt{then}\ P^{c_1,c_2}\ \mathtt{else}\ Q^{c_1,c_2}$ *where* x *is a fresh variable and true is a constant.*

To hide the output of a process, we use the following definition.

Definition 8 (Process $A^{\setminus out(ch,\cdot)}$ [?]). *Let A be an extended process. We define the process $A^{\setminus out(ch,\cdot)}$ as $\nu ch.(A|!\mathtt{in}(ch, x))$.*

3 Security Requirenments

We can now give the formal definitions of our fairness, authentication and privacy properties.

3.1 Fairness Properties

A fair auction protocol should not leak any information about any participant until the bidding phase is over and the winning bid is announced, and hence some information is inevitably leaked. We propose the following two definitions:

Definition 9 (Strong Noninterference (SN)). *An auction protocol ensures* Strong Noninterference (SN) *if for any two auction processes AP_A and AP_B that halt at the end of the bidding phase (i.e. where we remove all code after the last* `recBid` *event) we have $AP_A \approx_l AP_B$.*

This notion is very strong: Any two instances, independently of the participants and their offers, are required to be bisimilar until the end of the bidding phase. This would also require two instances with a different number of participants to be bisimilar, which will probably not hold on many protocols. A more realistic notion is the following:

Definition 10 (Weak Noninterference (WN)). *An auction protocol ensures* Weak Noninterference (WN) *if for any two auction processes* $AP_A = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,A}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,A}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *and* $AP_B = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,B}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,B}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *that halt at the end of the bidding phase (i.e. where we remove all code after the last* `recBid` *event) we have* $AP_A \approx_l AP_B$.

This only requires any two instances with the same participants $B\sigma_{id_i}$ to be bisimilar, however bids may still change. It is easy to see that (SN) implies (WN).

Another important fairness property is that there is no strategy that allows a malicious participant to win the auction at a chosen price, independently of the other bids.

Definition 11 (No Default Winning Strategy (NDWS)). *An auction protocol ensures* No Default Winning Strategy (NDWS) *if for any auction process* AP *we have that for* $AP'[B\sigma_{id_A}\sigma_{b_A} \mid (B\sigma_{id_B}\sigma_{b_B})^{c_1,c_2}]$ *where* b_A *is the highest submitted bid, there is no trace containing the event* `won` *for bidder* id_B *with a lower bid.*

The idea is the following: We have an honest bidder $B\sigma_{id_A}$ who submits the highest bid. The attacker has completely corrupted another bidder $B\sigma_{id_B}$ and should be unable to win the auction on his behalf on a lower bid.

3.2 Authentication Properties

The first authentication property we want to define is *Non-Repudiation*, i.e. that – once the winner has been announced – a winning bidder cannot claim that the winning bid was not send by him. As discussed in [**?**], Non-Repudiation can be expressed as form of authentication.

Definition 12 (Non-Repudiation). *An auction protocol ensures the property of* Non-Repudiation (NR) *if for every auction process* AP *on every possible execution trace the event* `won(p,id)` *is preceded by a corresponding event* `bid(p,id)`.

The intuition is simple: If there was a trace on which a bidder would win without submitting the winning bid, he could try to claim that he did not submit the winning bid even in a case where he rightfully won.

The second authentication property we want to model is *Non-Cancellation*, i.e. that a bidder cannot cancel a submitted bid before the winner is announced.

Definition 13 (Non-Cancellation). *An auction protocol ensures the property of* Non-Cancellation (NC) *if for any auction process* AP *which contains a bidder* $(B\sigma_{id_i}\sigma_{b_i})^{chc}$, *i.e. a bidder which reveals his secret data on channel* `chc` *(see Def. 6), and which submits the highest bid, i.e.* $\forall j \neq i : b_i > b_j$, *there is no trace containing the events* `recBid(b_i,id_i)` *and* `won(b_w,id_w)` *for another, lower bid, i.e.* $b_w < b_i$.

The idea is the following: The bidder id_i submits the highest bid, so he should win. If however there is the possibility that even though his bid was correctly received he did not win, this would mean that the intruder was able to cancel the bidder's bid even after reception. We require the bidder to reveal all his secret data to the intruder to capture the fact that the bidder himself might want to cancel his offer, in which case he could use his private data (keys etc.) to do so.

Both properties are independent: A protocol may implement the cancellation of bids as an official feature, for example after all bids have been submitted, bidders could be allowed to cancel their bids for a certain period of time, before the winner is finally announced. At the same time, such a protocol may ensure non-repudiation of the winner using e.g. signatures.

Similarly a protocol may ensure Non-Cancellation but no Non-Repudiation if the submitted bids cannot be canceled, but are not authenticated, so that the winner can successfully claim not having submitted the winning bid.

3.3 Privacy Properties

As explained in the introduction, we will consider different notions of privacy and analyze their relationship. We consider Privacy, Receipt-Freeness and Coercion-Resistance, and at each level two independent axes:

- the winner may stay anonymous (Strong Anonymity (SA, RF-SA, CR-SA), Weak Anonymity (WA, RF-WA, CR-WA)) or not (Strong Bidding-Price Secrecy (SBPS, RF-BPS, CR-BPS), Bidding-Price Unlinkability (BPU, RF-U, CR-U))
- the bids may stay completely private (Strong Bidding-Price Secrecy (SBPS, RF-BPS, CR-BPS), Strong Anonymity (SA, RF-SA, CR-SA)), or there might be list of all bids, which are however unlinkable to the bidders (Bidding-Price Unlinkability (BPU, RF-U, CR-U), Weak Anonymity (WA, RF-WA, CR-WA))

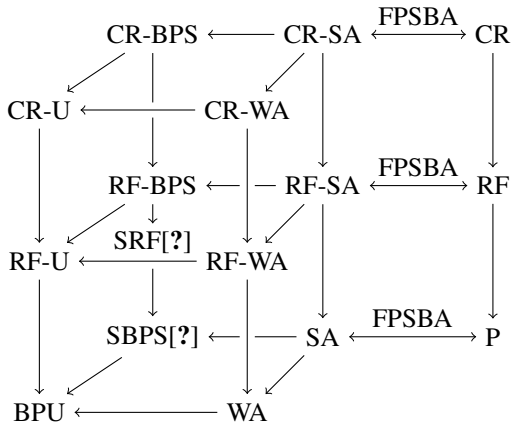

Fig. 2: Relations among the privacy notions. A $\xrightarrow{\text{C}}$ B means that under the assumption C a protocol ensuring A also ensures B.

These definitions are expressed for protocols implementing a first-price sealed-bid auction. We also provide the generalized notions (P), (RF) and (CR), which can also be applied to other types of auctions such as second-price auctions. We show that if a protocol correctly implements a First-Price Sealed-Bid Auction (FPSBA), these notions coincide with the corresponding Strong Anonymity-notions (SA), (RF-SA) and (CR-SA). Figure 2 provides an overview of the different notions.

Privacy. The first privacy notion we consider was proposed by Dong et al. [**?**].

Definition 14 (Strong Bidding-Price Secrecy (SBPS) [?]). *An electronic auction protocol ensures* Strong Bidding-Price Secrecy (SBPS) *if for an auction process AP and any bids $b_A, b_B < b_C$ we have*

$$AP'\left[B\sigma_{id_A}\sigma_{b_A}|B\sigma_{id_C}\sigma_{b_C}\right] \approx_l AP'\left[B\sigma_{id_A}\sigma_{b_B}|B\sigma_{id_C}\sigma_{b_C}\right]$$

The intuition is the following: If the losing bids are private, a losing bidder may change his bid for another losing one without this being noticeable to an attacker. This is expressed as an observational equivalence between two situations where a losing bidder changes his bid. Note that $B\sigma_{id_C}$ does not necessarily win since in AP' there might be a bidder offering a higher price, but $b_A, b_B < b_C$ guarantees that $B\sigma_{id_A}$ loses.

We propose the following, slightly weaker notion of Bidding-Price Unlinkability, which allows the losing bids to be public, however their link to the bidders have to be secret.

Definition 15 (Bidding-Price Unlinkability (BPU)). *An electronic auction protocol ensures* Bidding-Price Unlinkability (BPU) *if for an auction process AP and any bids $b_A, b_B < b_C$ we have*

$$AP'\left[B\sigma_{id_A}\sigma_{b_A}|B\sigma_{id_B}\sigma_{b_B}|B\sigma_{id_C}\sigma_{b_C}\right] \approx_l AP'\left[B\sigma_{id_A}\sigma_{b_B}|B\sigma_{id_B}\sigma_{b_A}|B\sigma_{id_C}\sigma_{b_C}\right]$$

In this definition we require two situations in which two losing bidders swap their bids to be bisimilar. This might be the case if the bids are public, but the real identity of the bidders is hidden, e.g. through the use of pseudonyms.

Note that the previous two notions only concern the losing bids, yet we might also want to preserve the anonymity of the winning bidder.

Definition 16 (Strong Anonymity (SA)). *An electronic auction protocol ensures* Strong Anonymity (SA) *if for an auction process AP and any bids $b_A, b_B < b_C$ we have*

$$AP'\left[B\sigma_{id_A}\sigma_{b_A}|B\sigma_{id_C}\sigma_{b_C}\right] \approx_l AP'\left[B\sigma_{id_A}\sigma_{b_C}|B\sigma_{id_C}\sigma_{b_B}\right]$$

Here we require two situations two be bisimilar where two different bidders win using the same offer, and the losing bidders may also use different bids in the two cases. This is intuitively stronger than Strong Bidding-Price Secrecy (SBPS).

A slightly weaker notion is Weak Anonymity, which allows the bids to be public, however their link to the bidders have to be secret, even for the winner.

Definition 17 (Weak Anonymity (WA)). *An electronic auction protocol ensures* Weak Anonymity (WA) *if for an auction process AP and any bids $b_A < b_C$ we have*

$$AP'\left[B\sigma_{id_A}\sigma_{b_A}|B\sigma_{id_C}\sigma_{b_C}\right] \approx_l AP'\left[B\sigma_{id_A}\sigma_{b_C}|B\sigma_{id_C}\sigma_{b_A}\right]$$

Here again two different bidders win using the same bid, but the losing bidder cannot choose his bid freely as above - the two bidders swap their bids. This intuitively implies Bidding-Price Unlinkability (BPU).

All these definitions are only meaningful for first-price auctions. To also deal with second-prices auctions, we can use the following generalization based on the published result.

Definition 18 (Privacy (P)). *An electronic auction protocol ensures* Privacy (P) *if for any two auction processes* $AP_A = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,A}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,A}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *and* $AP_B = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,B}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,B}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *we have*

$$AP_1|_{res} \approx_l AP_2|_{res} \Rightarrow AP_1 \approx_l AP_2$$

The intuition is quite simple: any two instances (consisting of the same bidders) which give the same result, i.e. the same winning bid, have to be bisimilar.

It turns out that for a correct first-price sealed-bid auction protocol which only publishes the winning price, this coincides with Strong Anonymity.

Definition 19 (First-Price Sealed-Bid Auction (FPSBA)). *An electronic auction protocol implements a* First-Price Sealed-Bid Auction (FPSBA) *if for any two auction processes* $AP_A = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,A}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,A}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *and* $AP_B = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,B}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,B}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *we have*

$$AP_A|_{res} \approx_l AP_B|_{res} \Leftrightarrow \max_i b_{i,A} = \max_i b_{i,B}$$

This definition requires the protocol to announce the same result if and only if the maximum among the submitted bids is the same, independently of which bidder submitted which bid. It is easy to see that this is true in the case of a correct first-price sealed-bid auction protocol. This allows us to prove the equivalence of (P) and (SA).

Theorem 1. *If an electronic auction protocol implements a First-Price Sealed-Bid Auction (FPSBA), Privacy (P) and Strong Anonymity (SA) are equivalent.*

Proof. Sketch: Assume we have two instances that give the same result, by (FPSBA) they have the same maximal bid. This bid may have been submitted by another bidder, and the other bids might have changed, but this can be proved using successive applications of (SA). Similarly if we assume two instances as in the definition of (SA), it is easy to see that they have the same maximal offer. Hence the result will be the same, and we can apply (P) to conclude. □

Receipt-Freeness. A first Receipt-Freeness definition for auction protocols was proposed by Dong et al. [**?**]. It is a generalization of Strong Bidding-Price Secrecy (SBPS).

Definition 20 (Simple Receipt-Freeness (SRF) [?]). *An electronic auction protocol ensures* Simple Receipt-Freeness (SRF) *if for an auction process* AP *and any bids* $b_A, b_B < b_C$ *there exists a process* B' *such that* $B'^{\backslash out(chc,\cdot)} \approx_l B\sigma_{id_A}\sigma_{b_B}$ *and*

$$AP' \left[(B\sigma_{id_A}\sigma_{b_A})^{chc} \mid B\sigma_{id_C}\sigma_{b_C} \right] \approx_l AP' \left[B' \mid B\sigma_{id_C}\sigma_{b_C} \right]$$

The intuition behind this definition is a follows: If the protocol is receipt-free, an attacker cannot distinguish between a situation where a losing bidder bids b_A and reveals all his secret data on a channel chc, and a situation where the bidder bids b_B and only pretends to reveal his secret data (the fake strategy, modeled by process B'). Note that Simple Receipt-Freeness (SRF) implies Strong Bidding-Price Secrecy (SBPS).

This definition has several shortcomings: Firstly, it ensures receipt-freeness only for one losing bidder, whereas in reality several bidders might be under attack. Secondly, consider for example a protocol that allows a losing bidder to create a fake receipt for himself, which however contains all other bids. Such a protocol would be secure according to this definition, but it would imply that creating a receipt violates the privacy of the other participants. To address these issues, we propose the following notions.

Definition 21 (RF-XXX). *A auction protocol ensures* RF-XXX *if for any two auction processes* $AP_A = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,A}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,A}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *and* $AP_B = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,B}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,B}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *such that*

- *if XXX=BPS* (Bidding-Price-Secrecy)*, there exists a* j *with* $b_{j,A} = b_{j,B} = \max_i b_{i,A} = \max_i b_{i,B}$ *and for any subset* $I \subset \{1, \ldots, j-1, j+1, \ldots k\}$*,*
- *if XXX=U* (Unlinkability)*, there exists* j *with* $b_{j,A} = b_{j,B} = \max_i b_{i,A} = \max_i b_{i,B}$ *and a permutation* Π *with* $\forall i : b_{i,B} = b_{\Pi(i),A}$*, and for any subset* $I \subset \{1, \ldots, j-1, j+1, \ldots k\}$*,*
- *if XXX=SA* (Strong Anonymity)*,* $\max_i b_{i,A} = \max_i b_{i,B}$ *and for any subset* $I \subset \{1, \ldots, k\}$*,*
- *if XXX=WA* (Weak Anonymity)*, here exists a permutation* Π *with* $\forall i : b_{i,B} = b_{\Pi(i),A}$*, and for any subset* $I \subset \{1, \ldots, k\}$*,*

there exist processes B'_i *such that we have* $\forall i \in I : B_i'^{\setminus out(chc_i,\cdot)} \approx_l B\sigma_{id_i}\sigma_{b_{i,B}}$ *and*

$$AP'_A\left[\mid_{i\in I} (B\sigma_{id_i}\sigma_{v_{i,A}})^{chc_i}\right] \approx_l AP'_B\left[\mid_{i\in I} B'_i\right]$$

Consider the first case, (RF-BPS): In this definition any subset of losing bidders may create fake receipts at the same time, and the other bidders can also change their bids. It is easy to see that this definition implies Simple Receipt-Freeness (SRF).

Similarly to our privacy definitions, we can also weaken (RF-BPS) and only consider cases where the bids are merely unlinkable to the bidders, by only considering permutations of the bids: We obtain (RF-U).

The third notion (RF-SA) is stronger in the sense that we also allow the winning bidder to be under attack, i.e. a winner needs to be able to create a fake receipt that proves that he lost, and a losing bidder needs to be able to create a fake receipt that proves that he won. Note that an attacker might ask a losing bidder to prove that he bid a certain price before the auction is over. If the bidder decides to bid less and create a fake receipt, the attacker may notice that he got a fake receipt if for example the winning bid is less that the price on the receipt. This is however an inherent problem of auctions, but our definition guarantees that a losing bidder can create a fake receipt for the winning price once the auction is over and the winning price is known.

Again, we can define a weaker version where the list of prices may be public, but it has to be unlinkable to the bidders, even for the winner: (RF-WA). It is easy to see that (RF-SA) implies (RF-BPS) and (RF-WA), and that both (RF-BPS) and (RF-WA) imply (RF-U).

Finally, the following definition is a generalization of Receipt-Free Strong Anonymity (RF-SA) (analogous to Privacy (P) and Strong Anonymity (Strong Anonymity)):

Any two instance giving the same result have to be bisimilar, even if bidders are under attack.

Definition 22 (Receipt-Freeness (RF)). *A auction protocol ensures* Receipt-Freeness (RF) *if for any two auction processes* $AP_A = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,A}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,A}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *and* $AP_B = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,B}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,B}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *and any subset* $I \subset \{1, \ldots, k\}$*, there exist processes* B'_i *such that we have*

$$\forall i \in I : B_i'^{\setminus out(chc_i,\cdot)} \approx_l B\sigma_{id_i}\sigma_{b_{i,B}}$$

and $AP_A|_{res} \approx_l AP_B|_{res} \Rightarrow AP'_A \left[\mid_{i\in I} (B\sigma_{id_i}\sigma_{v_{i,A}})^{chc_i} \right] \approx_l AP'_B \left[\mid_{i\in I} B'_i \right]$.

Similarly to Privacy (P), we prove that for protocols implementing a First-Price Sealed-Bid Auction (First-Price Sealed-Bid Auction), Receipt-Free Strong Anonymity (RF-SA) and Receipt-Freeness coincide.

Coercion-Resistance. Coercion-Resistance is a stronger property than receipt-freeness: The intruder may not only ask for a receipt, but is also allowed to interact with the bidder during the bidding process and to give orders. We can generalize the previously discussed Receipt-Freeness notions to Coercion-Resistance by adding the new intruder power as follows.

Definition 23 (CR-XXX). *An auction protocol ensures* CR-XXX *if for any two auction processes* $AP_A = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,A}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,A}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *and* $AP_B = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,B}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,B}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *such that*

- *if XXX=BPS* (Bidding-Price Secrecy)*: there exists a* j *with* $b_{j,A} = b_{j,B} = \max_i b_{i,A} = \max_i b_{i,B}$ *and for any subset* $I \subset \{1, \ldots, j-1, j+1, \ldots k\}$*,*
- *if XXX=U* (Unlinkability)*: there exists a* j *with* $b_{j,A} = b_{j,B} = \max_i b_{i,A} = \max_i b_{i,B}$ *and there exists a permutation* Π *with* $\forall i : b_{i,B} = b_{\Pi(i),A}$ *and for any subset* $I \subset \{1, \ldots, j-1, j+1, \ldots k\}$*,*
- *if XXX=SA* (Strong Anonymity)*:* $\max_i b_{i,A} = \max_i b_{i,B}$ *and for any subset* $I \subset \{1, \ldots, k\}$*,*
- *if XXX=WA* (Weak Anonymity)*: there exists a permutation* Π *with* $\forall i : b_{i,B} = b_{\Pi(i),A}$ *and for any subset* $I \subset \{1, \ldots, k\}$*,*

there exist processes B'_i *such that for any contexts* $C_i, i \in I$ *with* $C_i = \nu c_1.\nu c_2.(\_|P_i)$*,* $\tilde{n} \cap fn(C) = \emptyset$ *and*

$$AP'_A \left[\mid_{i\in I} C_i \left[(B\sigma_{id_i}\sigma_{b_{i,A}})^{c_1,c_2} \right] \right] \approx_l AP'_A \left[\mid_{i\in I} (B\sigma_{id_i}\sigma_{b_{i,A}})^{chc_i} \right]$$

we have $\forall i \in I : C_i \left[B'_i\right]^{\setminus out(chc_i,\cdot)} \approx_l B\sigma_{id_i}\sigma_{v_{i,B}}$ *and*

$$AP'_A \left[\mid_{i\in I} C_i \left[(B\sigma_{id_i}\sigma_{v_{i,A}})^{c_1,c_2} \right] \right] \approx_l AP'_B \left[\mid_{i\in I} C_i \left[B'_i\right] \right]$$

The difference to the previous receipt-freeness definitions is that the attacked bidders do not only reveal their data on channel c_1, but also take orders on channel c_2. The context C_i models the attacker that tries to force them to bid the price $b_{i,A}$ (this is expressed by the condition on C_i). The protocol is hence coercion-resistant if there exists a counter-strategy B' which allow the bidders to bid $b_{i,B}$ instead without the attacker noticing. For non sealed-bid first-price auction, we obtain the following definition.

Definition 24 (Coercion-Resistance (CR)). *An auction protocol ensures* Coercion-Resistance (CR) *if for any two auction processes* $AP_A = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,A}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,A}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *and* $AP_B = \nu\tilde{n}'.(B\sigma_{id_1}\sigma_{b_{1,B}} \mid \ldots \mid B\sigma_{id_k}\sigma_{b_{k,B}} \mid S \mid A_1 \mid \ldots \mid A_l)$ *and any subset* $I \subset \{1,\ldots,k\}$*, there exists processes* B'_i *such that for any contexts* $C_i, i \in I$ *with* $C_i = \nu c_1.\nu c_2.(\_ \mid P_i)$*,* $\tilde{n} \cap fn(C) = \emptyset$ *and*

$$AP'_A\left[\mid_{i\in I} C_i\left[(B\sigma_{id_i}\sigma_{b_{i,A}})^{c_1,c_2}\right]\right] \approx_l AP'_A\left[\mid_{i\in I} (B\sigma_{id_i}\sigma_{b_{i,A}})^{chc_i}\right]$$

we have $\forall i \in I : C_i\left[B'_i\right]^{\setminus out(chc_i,\cdot)} \approx_l B\sigma_{id_i}\sigma_{v_{i,B}}$ *and*

$$AP_A|_{res} \approx_l AP_B|_{res} \Rightarrow AP'_A\left[\mid_{i\in I} C_i\left[(B\sigma_{id_i}\sigma_{v_{i,A}})^{c_1,c_2}\right]\right] \approx_l AP'_B\left[\mid_{i\in I} C_i\left[B'_i\right]\right]$$

Again we can prove that for protocols implementing a First-Price Sealed-Bid Auction (FPSBA), Coercion-Resistant Strong Anonymity (CR-SA) and Coercion-Resistance (CR) coincide.

4 Case Studies

We applied the previously explained definitions on two case studies using ProVerif [**?**,**?**,**?**]: the protocol by Curtis et al. [**?**], and the protocol by Protocol by Brandt [**?**].

4.1 Protocol by Curtis, Pierprzyk and Seruga [?]

The protocol by Curtis et al. [**?**] was designed to support sealed-bid first- and second price auctions while guaranteeing fairness, privacy, verifiability and non-repudiation.

Informal Description. The main idea of the protocol is the following: The bidders register with a trusted Registration Authority (RA) using a Public-Key Infrastructure (PKI), which issues pseudonyms that will then be used for submitting bids to the Seller (S). It is split into three phases: Registration, Bidding, and Winner determination.

- *Registration:* Each bidder sends his identity, a hash of his bidding price b_i and a signature of $h(b_i)$ to the RA. The RA checks the identity and the signature using the PKI, and replies with an encrypted and signed message containing a newly generated pseudonym p and the hashed bid $h(b_i)$.

- *Bidding:* The RA generates a new symmetric key k. Each bidder will send $c = Enc_{pk_S}(b_i)$, his bid b_i encrypted with the seller's public key, and a signature of c, together with his pseudonym to the RA. The RA will reply with a signature on c, and encrypts the bidders message, together with the hashed bid $h(b_i)$ from phase one, using the symmetric key k. This encrypted message is then send to the seller.
- *Winner determination:* After all bids have been submitted, the RA will reveal the symmetric key k to the seller. The seller can then decrypt the bids, verify the correctness of the hash and determine the winner. To identify the winner using the pseudonym he can ask the RA to reveal the true identity.

Formal Model. We modeled the protocol in ProVerif using a standard equational theory for symmetric encryption (functions `senc` and `sdec`), asymmetric encryption (functions `enc`, `dec` and `pubkey` – which generates the public key corresponding to a secret key) and signatures (functions `sign`, `checksign` and `getmessage`):

$$\begin{aligned} \texttt{sdec}(\texttt{senc}(m, key), key) &= m \\ \texttt{dec}(\texttt{enc}(m, \texttt{pubkey}(sk)), sk) &= m \\ \texttt{checksign}(\texttt{sign}(m, sk), \texttt{pubkey}(sk)) &= m \\ \texttt{getmessage}(\texttt{sign}(m, sk)) &= m \end{aligned}$$

Due to space limitations we cannot include the full model here, the ProVerif code is available on our website [**?**].

Analysis. We discuss the following properties:

Non-Repudiation (NR): To prove (NR), we have to show that on each possible trace the event `won(p,id)` is preceded by the event `bid(p,id)`. ProVerif can verify such properties using queries, in this case using the query

```
query p:price,id:identity;
event(won(p,id)) ==> event(bid(p,id)).
```

This query means that for any value `p` of type `price` and any `id` of type `identity`, if the event `won(p,id)` is recorded, it is preceded by the event `bid(p,id)`. ProVerif finds the following attack: Since the channel between the Registration Authority and the Seller is not protected, anybody can pretend to be the RA and submit false bids, encrypted with a self-chosen symmetric key. After all false bids are submitted, the attacker reveals the symmetric key and the seller will decrypt the bogus bids. Hence the event `won(p,id)` can be emitted on a trace without any event `bid(p,id)`. We propose a solution to address this problem: If the messages from the RA to the seller are signed, the attacker cannot impersonate RA any more and ProVerif is able to prove Non-Repudiation for the accordingly modified protocol.

Non-Cancellation (NC): Here we have to show that even if a bidder reveals his secret data to the intruder, the intruder cannot cancel a submitted bid, i.e. there is no trace with the events `recBid(p_1,id_a)` and `won(p_2,id_b)` where `p_1 > p_2`. To

verify this we need to model at least two distinct prices, which can be implemented using constants, i.e. by setting `p_1 = max_price` and `p_2 = smaller_price`, where `max_price` and `smaller_price` are two constants such that `max_price > smaller_price`[2]. Then we want to test the conjunction (not the precedence as above) of two events, which is not possible directly in ProVerif. A well-known solution is to replace the underlying events with messages over a private channel to a newly added processes which will call a conjunction event `recBid_and_won` once he received all the messages. Then we can use the following query:

```
query event(recBid_and_won(max_price, id_a,
            smaller_price, id_b)).
```

where the first two parameters are from the event `recBid(p_1,id_a)` and the second from the event `won(p_2,id_b)`, here instantiated with price constants as explained above and two constants for two different bidders. For the original protocol, ProVerif finds a similar attack to the one described above: An attacker can delete the messages sent by the the RA to the seller, and choose a symmetric key and send bogus messages containing prices of his choice instead. When he reveals the symmetric key, a bidder of his choice will win, hence there will be an event `won(smaller_price,id_b)` for a smaller price than the one recorded by `recBid(max_price,id_a)`. Even if we add signatures as proposed above, ProVerif still comes up with an attack: A dishonest bidder might submit a first bid triggering the event `recBid` for this bid, delete the forwarded message to the seller, and then submit a second bid at a different price. A first attempt to fix this issue would be – as proposed in the original paper – by including the number of bids in the message where the RA reveals the symmetric key. This would allow the seller to verify if he received the correct number of bids. However the attack still works if two auctions take place in parallel: Since the RA uses the same PKI in both cases, he will use the same keys. The malicious bidder could register in the second auction, obtain the signed bid and replace his original bid with this message. The new message will include a different pseudonym, but the seller has no means of verifying if a pseudonym corresponds to the current auction. A solution would be to use different keys for different auctions (which need to be set up in a secure way), but we were unable to verify the resulting protocol because of some limitations of ProVerif: For example the counting of messages requires to maintain state information for the RA.

Noninterference: It is clear that the protocol does not ensure Strong Noninterference (SN) since an attacker can simply count the number of messages to determine the number of participants. However we can check Weak Noninterference (WN), i.e. that any two instances containing the same bidders and only differing in the bids are bisimilar up to the end of the bidding phase, using the following query in ProVerif:

```
noninterf b_1, ..., b_n.
```

This query will ask ProVerif to verify strong secrecy of the variables `b_1, ..., b_n.`, i.e. to check that any two instances of the protocol that only differ in these variables are bisimilar. For the original protocol ProVerif finds an attack which is based on

[2] Note that most auction protocols assume a finite number of possible prices anyway, which we can model using a list of constants.

the first message, which includes the hashed bidding price. An attacker simply hashes the possible values and compares the result. If we encrypt this message using the RA's public key, ProVerif is able to prove Weak Noninterference (WN). This modification was proposed in the original paper to achieve anonymity of bidders, but turns out to be also necessary to ensure fairness.

No Default Winning Strategy (NDWS): Here we have to show that a malicious bidder cannot win the auction at a chosen price, even if another bidder submitted a higher bid. Again, we will assume that we have a finite number of possible prices. Then we can check the property using ProVerif by modeling two bidders, the first one bidding `max_price`, and the second one is corrupted by the adversary (according to Def. 7). To prevent the adversary from just winning using the highest possible price (which would not necessarily correspond to an attack), we declare the constant `max_price` private[3]. We also have to be sure that the protocol does not leak `max_price` before the end of the bidding phase (which would contradict the intention of declaring it private). As we already showed Weak Noninterference (WN), we can be sure that this is not the case. Hence we can check if the event `won` is reachable for the corrupted bidder `id_B` using the following query

```
query p:price; event(won(p, id_B)).
```

Since bidder `id_A` submitted the highest possible price and the attacker cannot access and submit this value, he should be unable to make `id_B` win the auction. For the original protocol – only corrected with added encryption of the first bid to ensure Weak Noninterference –, ProVerif finds an attack again using the fact that the messages from the RA to the Seller are not authenticated, hence an attacker can pretend to be RA and submit bids of his choice to win the auction at a price of his choice. If we add signatures again, ProVerif still comes up with an attack: A dishonest bidder might register twice and then replace the message from the RA to the seller containing the correct bid with his own, bogus bid obtained using the second registration. As above, this could probably be circumvented by counting the messages and using different keys for different auction, but we hit again the limitations of ProVerif when trying to model and verify the resulting protocol.

Privacy: The authors claim in the original paper that if the first message is encrypted, their protocol ensures anonymity of the bidders. Yet we can see that it does not ensure Strong Anonymity (SA) since after the symmetric key has been published, an attacker can obtain a list with hashes of all bids, which allows to distinguish $h(b_A)$, $h(b_C)$ from $h(b_B)$, $h(b_C)$. Hence we checked Weak Anonymity (WA) using the `choice[]` operator in ProVerif, which verifies if the processes obtained by instantiating a variable with two different values are bisimilar. More precisely, we can check if for two swapping bidders (the first bidder bids `b_A = choice[b_1,b_2]`, the second `b_B`

[3] In the definition we did not require A to submit the highest possible bid, but only a higher bid than anybody else. We could model the existence of higher prices by defining additional private constants, but this would not change the verification task since they are never used by any honest participants and are not accessible to the attacker.

= choice[b_2,b_1]) the resulting processes are bisimilar. This query leads to another possible attack: The intruder might delay the messages from the second bidder until the first bidder sent his final message and this was relayed to the seller by the RA. This allows the attacker to link this message to the first bidder and distinguish both cases based on the hash after decrypting the message using the published symmetric key. This type of attack is well-known in electronic voting [**?**]. As a solution, we have to ensure that both messages to the seller are sent at exactly the same time using synchronization. Inspired by some techniques used in ProSwapper [**?**], we prove that the accordingly modified protocol ensures Weak Anonymity (WA).

It is also clear that the protocol is neither Receipt-Free nor Coercion-Resistant for any of the proposed notions since the hashed bidding price in the first message can be used as a receipt. Even if this message is encrypted, the data used to encrypt (keys, random values) can be used as a receipt. Note that for all properties ProVerif responds in less than a second on a standard PC.

4.2 Protocol by Brandt [?]

The protocol by Brandt [**?**] was designed to ensure full privacy in a completely distributed way. It exploits the homomorphic properties of a distributed El-Gamal Encryption scheme for a secure multi-party computation of the winner.

Informal Description. The participating bidders and the seller communicate using a bulletin board, i.e. an append-only memory accessible for everybody. The bids are encoded as bit-vectors where each entry corresponds to a price. The protocol then uses linear algebra operations on the bid vectors to compute a function f_i, which returns a vector containing one zero if the bidder i submitted the highest bid, and only random numbers otherwise. To be able to compute this function in a completely distributed way, and to guarantee that no coalition of malicious bidders can break privacy, these computations are performed on the encrypted bids using homomorphic properties of a distributed El-Gamal Encryption.

In a nutshell, the protocol realizes the following steps:

1. Firstly, the distributed key is generated: each bidder chooses his part of the secret key and publishes the corresponding part of the public key on the bulletin board.
2. Each bidder then computes the joint public key, encrypts his offer using this key and publishes it on the bulletin board.
3. Then the auction function f is calculated for every bidder using some operations exploiting the homomorphic property of the encryption scheme.
4. The outcome of this computation (n encrypted values) are published on the bulletin board, and each bidder partly decrypts each value using his secret key.
5. These shares are posted on the bulletin board, and can be combined to obtain the result.

Formal Model. Modeling the exchanged messages is straightforward (see [**?**] for the ProVerif code). Modeling the distributed encryption scheme and the distributed computations is a more challenging task since a too abstract model might miss attacks, whereas a too fine-grained model can lead to non-termination or false attacks.

The protocol assumes a finite set of possible prices, which we will model as constants p_1, ..., p_n. Assuming q bidders, we can define the following equational theory to model steps 3 and 4 of the protocol:

$$\begin{aligned}&\mathtt{f}(\mathtt{enc}(\mathtt{b\_1},\mathtt{pkey},\mathtt{r\_1}),...,\mathtt{enc}(\mathtt{b\_q},\mathtt{pkey},\mathtt{r\_q}),\mathtt{sk\_i})\\ =\ &\mathtt{share}((\max_{\mathtt{i}}\{\mathtt{b\_i}\},\arg\max_{\mathtt{i}}\{\mathtt{b\_i}\}),(\mathtt{b\_1},...,\mathtt{b\_q}),\mathtt{pkey},\mathtt{sk\_i},\mathtt{g}(\mathtt{r\_1},..,\mathtt{r\_q}))\end{aligned}$$

This equation models the following properties of the function f: If we have bids b_1, ..., b_q encrypted using the same joint public key pkey, some random values r_1, ... ,r_q, and a part sk_i of the secret key we obtain a share of the function outcome, i.e. the tuple (winning price, id of the winner), for the same public and secret keys and a function of the used random values. Since the share will look slightly different depending on the bids even if winning bid is the same, we also include b_1, ..., b_q in the share. This is necessary to avoid false attacks in ProVerif. The next equation corresponds to step 5 of the protocol and uses the function combine(pk(k_1), ..., pk(k_q)) which models the computation of the joint public key based on the individual ones.

$$\begin{aligned}&\mathtt{dec}(\mathtt{share}(\mathtt{m},\mathtt{x\_1},\mathtt{combine}(\mathtt{pk}(\mathtt{k\_1}),...,\mathtt{pk}(\mathtt{k\_q})),\mathtt{k\_1},\mathtt{r\_1}),...,\\ &\ \mathtt{share}(\mathtt{m},\mathtt{x\_q},\mathtt{combine}(\mathtt{pk}(\mathtt{k\_1}),...,\mathtt{pk}(\mathtt{k\_q})),\mathtt{k\_q},\mathtt{r\_q}))=\mathtt{m}\end{aligned}$$

The equation models that knowing all shares of the function outcome allows to decrypt it, if

- all shares have been constructed using the same joint public key, which was computed using the function combine from the individual public keys, and
- the individual public keys were computed from the same secret keys that were used to compute the shares.

Since the number of different prices n and the number of participants q are finite, we can enumerate all possible equations. In particular we can list all possible parameters of the function f, which allows us to enumerate all instances and replace the max and arg max functions which their actual values. This yields a convergent equational theory, which allows ProVerif to verify all the tested properties in less than one second.

Analysis. We use the same ProVerif techniques we discussed in the previous section. Essentially the protocol ensures none of the defined properties, manly due the lack of authentication. The attacker can simulate a completely different protocol execution towards the seller (i.e. setting up keys, encrypting bids of his choice, doing the calculation, and publishing the shares), which allows attacks on Non-Repudiation (a trace with event won, but without event bid), Non-Cancellation (a trace with event recBid and event won with a different, lower bid from the same bidder) and No Default Winning Strategy (the event won with a lower bid from a corrupted bidder is reachable).

Although the protocol claims to be fully private, ProVerif finds an attack that allows to completely uncover a bidder's bid: Since there is no authentication, an intruder can simulate all other parties with respect to a participant. He will generate secret keys, publish the according public keys and on reception of the attacked bidder's bid, simply

copy it and claim that it is his own bid. Then the joint computation and decryption will take place, and the announced winning price will be attacked bidder's offer, which is hence public. Note that this is not an attack on the security of the computation, but on the structure of the protocol.

It is also clear that the protocol does not ensure Strong Noninterference since the number of participants is public, which allows to distinguish instances with different number of participants. However we prove Weak Noninterference using `choice[]` (the use of `noninterf` leads to false attacks). The ProVerif-code is available in [**?**].

5 Conclusion and Future Work

We provided a framework to formally verify security properties in e-Auction protocols. In particular we discussed how protocols can be modeled in the Applied π-Calculus and how security properties such as different notions of Privacy, Fairness and Authentication can be expressed. We analyzed the relationship between the different notions and provided a hierarchy of privacy notions (Fig. 2).

Using two case studies [**?**,**?**], we showed how our definitions can be applied on existing protocols and are suitable for automated analysis using ProVerif. The results were surprising: One of the two protocols provided none of our security notions without modifications, the other protocol only one. It was particularly interesting to see that even the protocol by Brandt did not ensure privacy, although it was especially designed with privacy in mind. The discovered flaw is however not an attack on the cryptographic primitive used, but on the protocol architecture. This underlines again the complexity of designing secure protocols: A combination of secure building blocks can be insecure. In case of the protocol by Curtis et al. we also subsequently discussed several modifications to improve security.

As future work, we would like to verify Non-Cancellation and No Default Winning Strategy on the modified protocol by Curtis et al., which was not possible directly in ProVerif. There exist extensions which allow to model states, e.g. StatVerif [**?**] which might be used in this case. Additionally, we would like to formally define and check verifiability for auction protocols. This is necessary if a bidder does not want to trust the authorities. It requires to model tests that will allow a bidder to verify the outcome in a sound way, where it is not clear what "verify" actually means: If the winning bid is smaller than a bidder's own bid he knows that something went wrong, but this is probably not the only "bad" situation. Additionally, to be able to contest, the bidder would need some evidence to prove that he submitted a higher price.

References

1. Krishna, V.: Auction Theory. Academic Press (2002)
2. Brandt, F.: A verifiable, bidder-resolved auction protocol. In: Proceedings of the 5th AAMAS Workshop on Deception, Fraud and Trust in Agent Societies. (2002) 18–25
3. Brandt, F.: How to obtain full privacy in auctions. International Journal of Information Security **5** (2006) 201–216

4. Brandt, F., Sandholm, T.: On the existence of unconditionally privacy-preserving auction protocols. ACM Trans. Inf. Syst. Secur. **11** (2008) 6:1–6:21
5. Passch, C., Song, W., Kou, W., Tan, C.J.: Online auction protocols: A comparative study. In: Proceedings of the Second International Symposium on Topics in Electronic Commerce. ISEC '01, Springer-Verlag (2001) 170–186
6. Curtis, B., Pieprzyk, J., Seruga, J.: An efficient eauction protocol. In: ARES, IEEE Computer Society (2007) 417–421
7. Subramanian, S.: Design and verification of a secure electronic auction protocol. In: Proceedings of the The 17th IEEE Symposium on Reliable Distributed Systems. SRDS '98, IEEE Computer Society (1998) 204–
8. Dong, N., Jonker, H.L., Pang, J.: Analysis of a receipt-free auction protocol in the applied pi calculus. In: FAST 2010. Volume 6561 of LNCS., Springer (2010) 223–238
9. Backes, M., Hritcu, C., Maffei, M.: Automated verification of remote electronic voting protocols in the applied pi-calculus. CSF 2008 **0** (2008) 195–209
10. Delaune, S., Kremer, S., Ryan, M.: Verifying privacy-type properties of electronic voting protocols. Journal of Computer Security **17** (December 2009) 435–487
11. Dreier, J., Lafourcade, P., Lakhnech, Y.: Vote-independence: A powerful privacy notion for voting protocols. In: Proceedings of the 4th Workshop on Foundations & Practice of Security (FPS). LNCS, Springer (2011)
12. Dreier, J., Lafourcade, P., Lakhnech, Y.: Defining privacy for weighted votes, single and multi-voter coercion. In: ESORICS 2012. LNCS, Springer (2012)
13. Dreier, J., Lafourcade, P., Lakhnech, Y.: A formal taxonomy of privacy in voting protocols. In: First IEEE International Workshop on Security and Forensics in Communication Systems (ICC'12 WS - SFCS). (2012)
14. Küsters, R., Truderung, T.: An Epistemic Approach to Coercion-Resistance for Electronic Voting Protocols. In: S&P 2009, IEEE Computer Society (2009) 251–266
15. Smyth, B., Cortier, V.: Attacking and fixing helios: An analysis of ballot secrecy. In: CSF 2011, IEEE (2011) 297–311
16. Klay, F., Vigneron, L.: Formal aspects in security and trust. Springer-Verlag (2009) 192–209
17. Liu, J., Vigneron, L.: Design and verification of a non-repudiation protocol based on receiver-side smart card. Information Security, IET **4**(1) (March 2010) 15 –29
18. Abadi, M., Fournet, C.: Mobile values, new names, and secure communication. In: POPL 2001. POPL '01, New York, ACM (2001) 104–115
19. Blanchet, B.: An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In: CSFW 2001, Cape Breton, Nova Scotia, Canada, IEEE Computer Society (June 2001) 82–96
20. Blanchet, B.: From Secrecy to Authenticity in Security Protocols. In: 9th International Static Analysis Symposium (SAS'02). Volume 2477 of LNCS., Springer Verlag (2002) 342–359
21. Blanchet, B., Abadi, M., Fournet, C.: Automated verification of selected equivalences for security protocols. Journal of Logic and Algebraic Programming **75**(1) (2008) 3–51
22. Dreier, J., Lafourcade, P., Lakhnech, Y.: Formal verification of e-auction protocols. Technical Report TR-2012-17, Verimag Research Report (October 2012) Available at `http://www-verimag.imag.fr/TR/TR-2012-17.pdf`.
23. Dreier, J.: The proverif code used to automatically verify the examples is available at `http://www-verimag.imag.fr/~dreier/papers/post-code.zip` (2012)
24. Klus, P., Smyth, B., Ryan, M.D.: Proswapper: Improved equivalence verifier for proverif. `http://www.bensmyth.com/proswapper.php` (2010)
25. Arapinis, M., Ritter, E., Ryan, M.D.: Statverif: Verification of stateful processes. In: CSF 2011. CSF '11, IEEE Computer Society (2011) 33–47