Secure Physical Auctions for the Non-Expert

Secure Physical Auctions for the Non-Expert

Jannik Dreier
Institute of Information Security, ETH Zurich
jannik.dreier@inf.ethz.ch

Hugo Jonker
University of Luxembourg
hugo.jonker@uni.lu

Pascal Lafourcade
Clermont Université, Université d'Auvergne, LIMOS
pascal.lafourcade@udamail.fr

January 14, 2014

Abstract

An auction is a simple way of selling and buying goods. Modern auction protocols often rely on complex cryptographic operations to ensure manifold security properties, for instance bidder-anonymity or bid-privacy, non-repudiation, fairness or public verifiability of the result. This makes them difficult to understand for users who are not experts in cryptography. We propose two physical auction protocols inspired by Sako's cryptographic auction protocol. In contrast to Sako's protocol, they do not rely on cryptographic operations, but on physical properties of the manipulated mechanical objects to ensure the desired security properties. The first protocol only uses standard office material, whereas the second uses a special wooden box. We validate the security of our solutions using ProVerif.

1 Introduction

Auctions provide sellers and buyers with a way to exchange goods for a mutually acceptable price. Unlike a marketplace, where the *sellers* compete with each other, auctions are a seller's market where *buyers* bid against each other over the goods for sale. Because of the competitive nature of the process, often an *auctioneer* serves as a trusted third party to mediate the process. However, in many cases (for example on eBay) the auctioneer charges a percentage of the selling price as his fee. Hence he has a financial interest in the auction, which may compromise his neutrality.

Invariably, common auction protocols rely on certain cryptographic primitives and/or trusted parties to combine seemingly contrary security goals like privacy and verifiability. Examples include hash chains [30], signatures of knowledge and zero-knowledge proofs [24], coin-extractability, range proofs and proofs of knowledge [19], and proxy-oblivious transfers and secure evaluation functions [23]. Sako's protocol [27], explained in detail in Section 2, applies public-key encryption in a clever way to implement a verifiable sealed-bid auction. Although it is fully verifiable, the bidders need to trust the auctioneers for privacy of the losing bids. Brandt's protocol [8] goes even further: with the help of an ad hoc cryptographic primitive, Brandt claims to achieve full privacy for *all* bidders, i.e. only the winner and the seller learn who the winner is.

However, both protocols rely on complex cryptography difficult to understand for a non-expert. This is particularly intriguing when it comes to verifiability – anyone lacking cryptographic expertise cannot ascertain for themselves that the verification procedure is indeed correct, and is thus forced to trust the judgment of cryptographic experts. This view underlies the German Constitutional Court's decision on electronic voting machines: "the use of electronic voting machines requires that the essential steps of the voting and of the determination of the result can be examined by the citizen reliably and *without any specialist knowledge of the subject*" [9]. Chaum [10] argued along the same line in 2004 that all the ingeniously designed verifiable voting protocols that had been put forward in literature did little to empower actual voters to verify elections. To address this issue, he proposed a voting protocol using visual cryptography: the ballot was distributed over two layers, that on top of each other showed the voter's choice. One layer was destroyed, leaving the voter with a layer full of random dots from which no choice can be inferred. However, anyone can verify that the system accurately recorded this layer – *without* any cryptographic expertise. In the same spirit, we propose in this paper two auction protocols that only rely on physical manipulations to enable non-experts to understand the protocol and its verification procedure.

Apart from Chaum's "true voter-verifiable" voting protocol [10], the power of (partly) physical protocols have also been studied for other applications. Stajano and Anderson [29] proposed a partly physical auction protocol using anonymous broadcast (e.g. small radio transmitters), which however still uses some cryptography (e.g. one-way functions and a Diffie-Hellman key exchange). More generally, Moran and Naor showed that many cryptographic protocols can be implemented using tamper-evident seals [22]. They also analyzed a polling protocol based on physical envelopes [21]. Moreover, Fagin, Naor and Winkler [17] described various physical methods of comparing two secret values. Finally, Schneier [28] proposed a cypher based on a pack of solitaire cards.

Although formal verification is now a common approach to evaluate security properties, few formal analyses exist in the auction domain. Subramanian proposed a logic to analyze protocols [31], inspired by the BAN logic. He applied this logic to analyze a simple auction protocol by his own design. Later, Dong et al. [12] analyzed privacy of the protocol due to Abe and Suzuki [3] using the Applied π-Calculus. That auction protocol claims a strong form of privacy (receipt-freeness), a claim which Dong et al. were able to prove independently. Küsters et al. [18] proposed generic definitions of accountability and verifiability. They make a distinction between verifiability (the ability to see that something went wrong) and accountability (the ability to blame someone for what went wrong). Küsters et al. used their definitions to identify some problems in the auction protocol due to Parkes et al. [25]. Dreier et al. [15] formalized fairness, authentication, and privacy notions in the Applied π-Calculus. This was followed by a first-order logic definition of auction verifiability in [13].

All this work is however only concerned with the verification of cryptographic protocols, not physical protocols. Blaze [7] was among the first to argue that physical security should be taken into account in security modeling. Other work in this area focused on modeling physical security in various ways. The Portunes framework by Dimkov et al. [11] allows modeling of attacks that cross physical, digital and social domains. Basin et al. [4] proposed a model taking physical dimensions such as time and distance into account. Recently Meadows et al. [20] describe a way to formalize security procedures (accounting for physical objects) in logic. By using the frameworks of [15] and [13] we show that we can actually apply the same definitions used for the cryptographic protocols on protocols based on physical properties, hence giving us a powerful way to evaluate and compare both approaches.

Contributions. We start by recalling Sako's auction protocol [27]. Inspired by this protocol, we propose a first physical implementation called *Cardako*[1]. This variant does not rely on cryptography nor on trusted parties, yet retains the verifiability, privacy, authentication and fairness properties of Sako's protocol. Based on the definitions by Dreier et al. [13, 15] we also provide a formal analysis of these security properties in ProVerif [5], modeling their physical properties using a special equational theory. Although ensuring privacy for the losing bidders, both the Sako protocol and the Cardako variant publicly reveal the winner. Our final contribution is *Woodako*[2]: a physical auction protocol that offers stronger privacy *i.e.*, the winner is not publicly revealed, yet the result remains verifiable for losing bidders (similar to the protocol by Brandt [8]). In this protocol, physical properties take the place of cryptography and the trusted auctioneer. We build a concrete prototype, and formally verify the security properties with the help of ProVerif.

Outline. In the next section we describe Sako's protocol, its security properties and trust assumptions. In §3, we present the first protocol called "Cardako", and discuss the security properties it achieves. In §4, we describe the "Woodako" protocol, which ensures a higher level of privacy than Cardako, before concluding in §5.

2 Protocol by Sako

Sako [27] proposed a protocol for sealed-bid first-price auctions which hides the bids of losing bidders and ensures verifiability. The paper provides a high-level description using a generic cryptographic primitive that ensure certain properties. Sako also proposes two instantiations using specific cryptographic primitives: the first one uses Elgamal [16] encryption, and the second one employs a probabilistic version of RSA [26]. Note that in this protocol dishonest authorities can break privacy, but because of verifiability a manipulation of the auction outcome can be detected.

[1] **Card**board version of S**ako**'s protocol.

[2] **Wood**en box based implementation of S**ako**'s protocol.

2.1 Informal Description

Informally, the protocol works as follows:

1. The authorities select a list of allowed bids $p_1, \ldots, p_m$.

2. For each allowed bid p_i, the authorities set up encryption and decryption algorithms E_{p_i} and D_{p_i} (in both implementations simply a public-private key pair). The encryption scheme must provide an indistinguishability property. The authorities publish the encryption algorithms (or public keys in the implementation) and the list of allowed bids on a bulletin board.

3. To bid for price p_i, a bidder encrypts a public constant c using E_{p_i}, signs it and publishes the bid $C_j = E_{p_i}(c)$ together with the signature on the bulletin board.

4. After the bidding phase is over, the authorities check the signatures and start decrypting all bids with the highest possible price $t = p_m$. If $D_t(C_j) = c$, then bid j was a bid for price t. If all decryptions fail, the authorities decrease t and try again. Each time a decryption is done, they publish a proof of correct decryption to enable verifiability. This can be a zero-knowledge proof, or it might be achieved by simply publishing the secret key.

5. To verify the outcome, anybody can verify the signatures, and check the proofs of correct decryption.

In the rest of the this section we consider the implementation based on public and private key pairs as a concretization of the general encryption/decryption algorithms, however we abstract away of the precise encryption scheme. Note that dishonest authorities can break privacy since they have access to all secret keys, but because of verifiability a manipulation of the auction outcome can be detected.

2.2 Security Properties

We now argue informally that the protocol ensures *Fairness*, *Non-Repudiation*, *Non-Cancellation* (as defined in [15]) and *Verifiability* (as defined in [13]). Moreover it ensures *Privacy* of the losing bidders [15] if the authorities are trusted.

Non-Cancellation and Non-Repudiation [15]. The bids are signed and published on the append-only bulletin board. Hence a bidder cannot deny that he made his bid, and the submitted bids cannot be altered or otherwise canceled.

Fairness. We consider the two aspects defined in [15]:

- *Highest-Price-Wins:* This property is to ensure that an attacker cannot win the auction at a price below the actual highest bid. In this protocol, the authorities start by decrypting using the decryption algorithm corresponding to the highest possible price (if not, this can be detected, see Verifiability), hence they will identify the highest bid. Similarly, because of the signatures on the bids and the properties of the bulletin board, the bids cannot be modified, deleted or replaced. Hence the bidder submitting the highest price will be correctly identified as the winner, even in presence of an attacker controlling the network.

- *Weak-Non-Interference:* This property is to ensure that no information about the bidders' bids is leaked before the bidding phase ends – otherwise they might employ unfair strategies based on that information. In this protocol the bids leak no other information apart from the identity of the bidders (revealed by the signature) because of the indistinguishability property of the encryption scheme.

Verifiability. Everybody can check the signatures of the bids on the bulletin board, ensuring that all bids originated from eligible bidders and were not modified. Similarly, all participants can use the proofs of correct decryption to check whether the authorities opened the bids correctly, hence ensuring the correctness of the outcome computation.

Privacy. The authorities have all private keys and can hence open all bids, breaking privacy. If the authorities are trusted, they will discard all unused keys, thereby preventing anyone from opening the losing bids and breaking the privacy of losing bidders. Given the indistinguishability property of the encryption scheme, this ensures secrecy of the losing bids.

2.3 Formal Model

To formally verify the above properties we use ProVerif [5]. ProVerif uses a process description heavily inspired by the Applied π-Calculus [1], however has syntactical extensions and is enriched by events to check reachability and correspondence properties. Due to the space limitations we do not recall the full syntax and semantics here, this is available in the original paper [5] and the ProVerif documentation [6].

In short, the behavior of honest parties is modeled as processes in ProVerif. These processes can exchange messages on public or private channels, create keys or fresh random values and perform tests and cryptographic operations, which are modeled as functions on terms with respect to an equational theory describing their properties. In ProVerif, the attacker has complete control of the network (excluding private channels).

To verify Sako's protocol we need to model public-key encryption, signatures and proofs of correct decryption. This can be done using the following equational theory:

$$\begin{aligned}
\mathtt{checksign}(\mathtt{sign}(m,k),\mathtt{pk}(k)) &= m\\
\mathtt{getmessage}(\mathtt{sign}(m,k)) &= m\\
\mathtt{dec}(\mathtt{penc}(m,\mathtt{pk}(k),s),k) &= m\\
\mathtt{checkproof}(\mathtt{decProof}(\mathtt{penc}(m,\mathtt{pk}(k),s),m,k),&\\
\mathtt{penc}(m,\mathtt{pk}(k),s),m,\mathtt{pk}(k)) &= \mathtt{true}\\
\mathtt{checkproof}(\mathtt{decProof}(\mathtt{penc}(m,\mathtt{pk}(k1),s),&\\
\mathtt{dec}(\mathtt{penc}(m,\mathtt{pk}(k1),s),k2),k2),\mathtt{penc}(m,\mathtt{pk}(k1),s),&\\
\mathtt{dec}(\mathtt{penc}(m,\mathtt{pk}(k1),s),k2),\mathtt{pk}(k2)) &= \mathtt{true}
\end{aligned}$$

```
let bidder(b:pkey,k:skey,chBB:channel) =
 new s:seed; let offer:bitstring = penc(bidval,b,s) in
 event bid(offer,pubkey(k));
 out(chBB,(offer,sign(offer,k))).
```

Listing 1: The bidder.

The first two equations model signatures: If a signature on the message m is checked using the correct public key, we obtain the message m. Similarly the third equation models probabilistic public-key encryption: A message m encrypted with a public key $pk(k)$ and a fresh random seed s can only be opened using the corresponding private key k. The last two equations model proofs of correct decryption: The verification succeeds if the proposed plaintext is the actual decryption of the ciphertext under the claimed key, even if this decryption is not meaningful as the key is not the correct one.

Consider the ProVerif code in Listing 1 describing the behavior of a bidder in Sako's protocol. The bidder process has three parameters: the key `b`, corresponding to the key representing his bid, his secret key `k` used for signing and the channel to the bulletin board `chBB`. He draws a fresh random seed `s`, encrypts the constant `bidval` using the price-key `b`, computes a signature on the ciphertext, executes the event `bid` on his (signed) offer and sends it to the bulletin board.

The authority is modeled as shown in Listing 2. Firstly the bids are received from the bulletin board. Then the signatures are checked and the bids are decrypted using the key corresponding to the highest possible price, and the decryptions are published together with a proof. Finally, if the first bidder submitted a bid for the highest price, he is declared a winner, otherwise the second bid is checked, and so on. If none of the decryptions is correct, the authority decreases the price and tries again.

Having completed the modeling of the protocol, we need to express the security properties. Here we rely on the definitions by [15] and [13].

2.4 Analysis

We test *Non-Repudiation* using the following query in ProVerif

```
                query offer:bitstring,id:pkey;
     event(won(offer,id)) ==> event(bid(offer,id)).
```

To also account for dishonest bidders, we give all data of the bidder except for his secret key to the intruder, and let him compose the message which is signed and sent to the bulletin board (see Listing 4 for the modified bidder's process). ProVerif proves that the property still holds.

Non-Cancellation is tested using a process that tests if the conjunction of events `recBid` and `won` for a lower bid occurs, and then executes an event `bad`. ProVerif concludes that this event is unreachable using the query `query event(bad())`.

For *Highest Price Wins*, we again use a process that executes an event `bad` if a situation violating the property (i.e. an event `won` for a bidder different from

```
let authority(p1:skey,...,pm:skey,k1:pkey,...,kn:pkey,
 chBB1:channel,...,chBBn:channel,chO11:channel,...,
 chOn1:channel,...,chO1m:channel,...,chOnm:channel,
 chW:channel) =
  in(chBB1,(m1:bitstring, s1:bitstring)); ...
  in(chBBn,(mn:bitstring, sn:bitstring));
  if checksign(s1,k1) = m1 && ... && checksign(sn,kn) = mn then
    let dec11 = dec(m1,p1) in
    out(chO11,(m1,dec11,decProof(m1,dec11,p1))); ...
    let dec1n = dec(mn,p1) in
    out(chOn1,(mn,dec1n,decProof(mn,dec1n,p1)));
    if dec11 = bidval then
      event won(m1, k1); out(chW,(m1,s1,one,one))
    else if ...
      ...
    else if dec1n = bidval then
      event won(mn, kn); out(chW,(mn,sn,n,one))
    else
      ...
      let decm1 = dec(m1,pm) in
      out(chO1m,(m1,decm1,decProof(m1,decm1,pm))); ...
      let decmn = dec(mn,pm) in
      out(chOnm,(mn,decmn,decProof(mn,decmn,pm)));
      if decm1 = bidval then
        event won(m1, k1); out(chW,(m1,s1,one,m))
      else if ...
        ...
      else if decmn = bidval then
        event won(mn, km); out(chW,(mn,sn,n,m)).
```

Listing 2: The authority.

```
let testrvs(chRVS:channel,k1:pkey,...,kn:skey,
              chBB1:channel,...,chBBn:channel) =
 in(chBB1,(m1:bitstring,s1:bitstring)); ...
 in(chBBn,(mn:bitstring,sn:bitstring));
 if checksign(s1,k1) = m1 && ...
     && checksign(sn,kn) = mn then
   out(chRVS,OK)
 else
   out(chRVS,KO).
```

Listing 3: The test rv_s.

```
let bidder(b:pkey,k:skey,chBB:channel,chAd:channel)=
  new s:seed; out(chAd,(s,b)); in(chAd,offer:bitstring);
  event bid(offer,pubkey(k));
  out(chBB,(offer,sign(offer,k))).
```

Listing 4: The dishonest bidder.

the one submitting the highest bid) is detected. ProVerif can then conclude that such an event is unreachable.

To verify *Weak Non-Interference* we use the `choice` operator. In ProVerif `let x=choice[a,b] in P` means that ProVerif will compare the processes `P` where `x` is either `a` or `b`, and try to prove their observational equivalence. We use a small python script to generate all the cases for two bidders and two prices, which can then successfully be checked using ProVerif.

Privacy is also tested using the `choice` operator. As the protocol reveals the winner and the winning price, the highest achievable privacy notion is Strong Bidding-Price Secrecy (cf. [15]). Hence we test two situations, where in both situations bidder one bids the same highest price `one` and wins. In situation one bidder two bids price `two`, in situation two he bids `three`. Since he loses in both situations, no information about his bid should be leaked, and both situations should be observationally equivalent. If we suppose an honest authority, ProVerif is able to prove this result, even if we add a corrupted bidder. Note that the protocol is neither receipt-free nor coercion-resistant as the random values used to encrypt the constant can be used as a receipt.

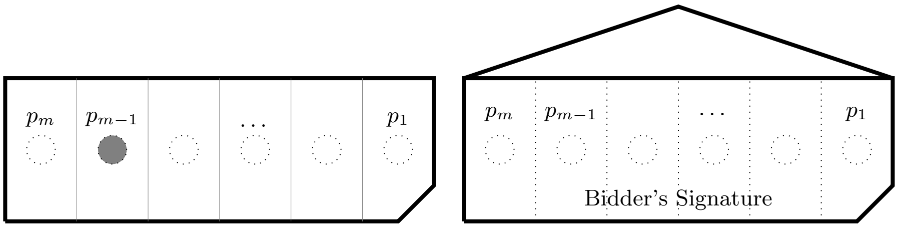
Proving *Verifiability* consists in proving that the verification tests are sound and complete. Here we have three verification tests following the model of [13]:

- A test rv_s that checks if all bids were submitted by registered bidders, which consists of checking the signatures
- A test rv_w that checks if the winning bid is one of the submitted bids
- A test ov that verifies that the announced winning bid is actually the highest bid

In each case, we express the test as a process that takes as input the data from the auction and accepts or not this input (see for example Listing 3). For soundness we then need to prove that the test only accepts if the result was actually correct, which can be achieved by testing if the branch accepting the outcome is reachable for an incorrect input – again a query for the event `bad`. Similarly for completeness we test if the branch rejecting the input data is reachable when the input is generated by honest parties according to the protocol specification. ProVerif is able to conclude successfully for all six cases (completeness and soundness of each test). For all the tests described in this section ProVerif only takes a few seconds to answer; the code is available online [14].

Remark 1 *In the literature an attack on fairness was described [2]. The attack targets the implementation using the ElGamal encryption and works as follows. A dishonest bidder encrypts the constant using a key of his choice, but using zero as a random value. He obtains a ciphertext of the form* $(1, M)$*, which decrypts under any private key. A dishonest authority can then decrypt the bid at any chosen price, for example one euro higher than the highest other bid.*

This attack violates Highest Price Wins: a corrupted bidder that did not submit the highest bid wins. Note that we did not identify it as we assume an honest authority, whereas the attack requires the authority to collude with the bidder. If however we decide to consider dishonest authorities, we obviously have even simpler attacks: the authorities can simply announce a winner and winning price of their choice.



(a) Cardako bid for price p_{m-1}. (b) Cardako bidding envelope.

Figure 1: The Cardako protocol

However, this attack will be detected during the verification phase. If the authority decides to open the bogus bid at price $x_m + 1$, he has to prove that the bogus bid does not decrypt to the value bidval for any key corresponding to a price higher than $x_m + 1$. Since the bogus bid decrypts under any key, he is unable to do so. The only possible way to pass the verification test is to open the bogus bid at the highest possible price and to declare the dishonest bidder a winner at this price, since then no other keys are used in the verification. This however gives the bidder no advantage over simply submitting a bid for the highest price using a correct encryption, hence the attack does not compromise fairness if the verification is carried out correctly.

Remark 2 *As noted in the original paper [27], there is another fairness issue. In the abstract version of the protocol, a dishonest bidder can copy somebody else's bid, sign it, and submit it as his own bid. This allows him to provoke a tie, i.e. he is sure not to bid the same price as a targeted bidder. In our model ties are automatically resolved by choosing the first bidder to submit the winning offer, hence the attack does not occur in our setting. However in general the protocol supports other tie-breaking mechanisms, where this attack can become a problem. Note however that this issue is easy to address – Sako proposed three possible fixes: rejecting copies, adding proofs of ownership, or adding the bidders identity to the encrypted plaintext.*

3 The "Cardako" Protocol

It is a practical implementation of the concepts of Sako's protocol using office material.

3.1 Description

In the Cardako protocol each bidder has a piece of cardboard marked with a number of positions corresponding to possible prices in descending order from left to right (see Figure 1a). A bidder chooses his price (in Figure 1a, the second-highest price p_{m-1}), and punches a hole in the position corresponding to his bid. Next, he inserts the card into a special envelope of the same size with marks corresponding to the different prices on the outside (see Figure 1b), and signs on the outside of the envelope. The envelope is sealed and shown to all other bidders so that they can check the signature.

Once all bidders have finished creating bids and shown their signatures, the bidders mix envelopes and jointly try to punch a needle through the marked areas of each envelope, starting with the highest possible price. If this succeeds, they find a hole and hence a bid for this price. The signature on the outside then allows the identification of the winner. If this fails for all bids, the bidders repeat the procedure for the second price, and so on. Since the order of the prices is important, the cards and envelopes are designed such that they can only fit together in one way. This ensures that the card cannot accidentally be turned around.

To fully ensure verifiability, the protocol must also ensure that only eligible bidders can bid. This is achieved through the verification of the signatures on the envelopes by the seller and bidders when bids are posted.

3.2 Security Properties

The Cardako protocol relies on the physical properties of the cardboard and the envelopes: Nobody can see from the outside the contents of a bid, but by trying to punch with a needle the bidders can test if it was an offer for a certain price. It offers verifiability similar to Sako's protocol as well as non-cancellation and non-repudiation due to the signatures and the mixing of the envelopes. It ensures fairness since no premature information is leaked (*Weak Non-Interference*) and due to the joint bid opening no cheating is possible (*Highest Price Wins*).

Obviously a malicious bidder can open an envelope and read its contents – but this is actually similar to Sako's protocol, where dishonest authorities can break privacy. The difference is that in Cardako such a behavior will be detected by the other bidders, since the envelope is damaged and the other parties are in the same room. An extension to improve privacy could be to put the signed envelopes into slightly bigger and indistinguishable envelopes after the signature has been verified by the other parties. These envelopes can be posted into a ballot box to break the link between a bidder and his bid. Hence a malicious bidder will only be able to break the privacy of a random bid, but not necessary of the one he is interested in.

A positive side effect of using indistinguishable envelopes is that ties can be detected and resolved fairly without revealing the identities of the tied bidders (unlike Sako's protocol). This happens as follows: first, the protocol determines which envelopes contain winning bids. As the envelopes are indistinguishable from one another, the identity of the tied bidders is not revealed yet. The tie is then broken by selecting a random envelope (e.g. by rolling a die, or drawing an envelope from a hat).

3.3 Formal Analysis

We model the bidders as processes exchanging messages (envelopes or real communication messages), however we also need to model the physical properties of the objects used. Our approach consists in modeling the properties using an equational theory, following the usual technique used in symbolic models: the primitives are "perfectly secure" and we simply describe their abstract properties. We describe the envelope using a function `envelope` that is created using a random seed to hide its contents and can only be opened using that seed, similar to a cryptographic commitment. However we also have functions `testone`, `testtwo`, ...,`testm`, that can test for a certain value without opening the envelope (i.e., the needle tests):

$$\begin{aligned}
\texttt{open}(\texttt{envelope}(\mathit{content}, k), k) &= \mathit{content}\\
\texttt{testone}(\texttt{sign}(\texttt{envelope}(x, k), sk)) &= \begin{cases} \texttt{true} & \text{if } x = \texttt{one},\\ \texttt{false} & \text{otherwise} \end{cases}\\
&\vdots\\
\texttt{testm}(\texttt{sign}(\texttt{envelope}(x, k), sk)) &= \begin{cases} \texttt{true} & \text{if } x = \texttt{m},\\ \texttt{false} & \text{otherwise} \end{cases}\\
\texttt{checksign}(\texttt{sign}(m, k), \texttt{pk}(k)) &= m\\
\texttt{getmessage}(\texttt{sign}(m, k)) &= m\\
\texttt{getpubkey}(\texttt{sign}(m, k)) &= \texttt{pk}(k)
\end{aligned}$$

The last three equations model signatures. The first equation allows to verify a signature, the second equation to obtain the signed message, and the last one to identify the person who signed the message.

The honest participants obviously only test for the values that they are supposed to, but dishonest participants can also apply tests they are not supposed to execute. To model the fact that any party in possession of an envelope can open it, the bidders give away the random seed they used to create it when giving away the envelope.

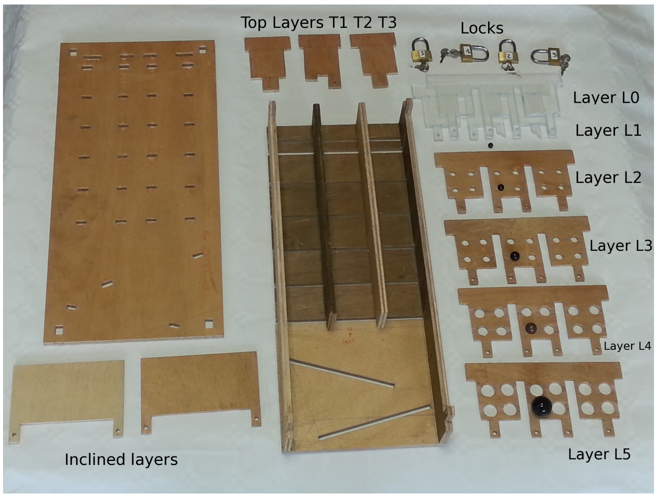
This allows us to repeat the same verification steps as for Sako's protocol and to conclude that the protocol ensures *Non-Repudiation*, *Non-Cancellation*, *Weak Non-Interference*, *Highest Price Wins* and *Verifiability.*

When verifying *Privacy*, ProVerif finds the obvious attacks of opening the envelopes or testing for all values. If however we keep the seeds private and remove all tests except for the required ones (in an attempt to model the fact that bidders do not want to risk being caught when breaking the rules) ProVerif is able to prove secrecy of the losing bids. All the above verifications succeed within a few seconds on a standard office PC; the full code is available online [14].

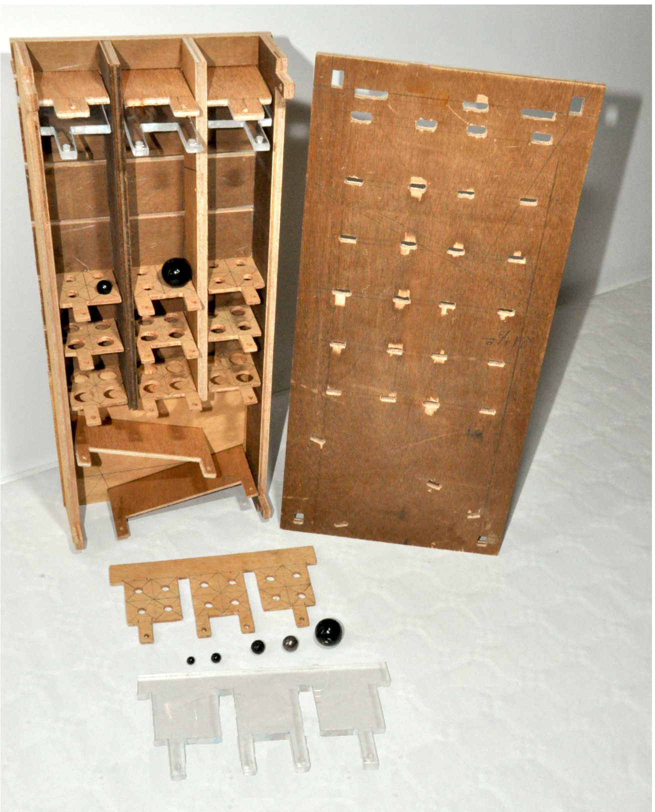
4 The "Woodako" Protocol

To improve the privacy of the Cardako protocol, we developed *Woodako*, which relies on a special wooden box. Our prototype is designed for 3 bidders and 5 possible prices, but such a box can be built for any number n of bidders and any number k of prices. Figure 2a shows all components of the box. The Woodako auction system uses:

- Five black marbles per bidder, each size represents one price.
- Six layers ($L0 - L5$): Layers $L0$ and $L1$ are made of transparent plexiglass and have no holes. The other layers are made of wood and contain four holes per column, which correspond to the size of the marbles: the holes in $L2$ are only big enough for the smallest marbles, the holes in $L3$ for the second-smallest etc.
- Three top layers $T1$, $T2$, $T3$. Each layer is associated with a bidder.
- Two inclined layers: these are placed below the layers, near the bottom of the box.
- Locks and keys: each bidder and the seller has a set of locks and keys.



(a) The Woodako prototype.



(b) Inside our Woodako prototype, where layers $L0$ and $L1$ are removed.

Figure 2: Our prototype

- One front side made of wood that closes the box and contains holes to insert the extremities of the layers. These extremities will stick out and so constitute a place where the parties can put locks. These locks are used to ensure security properties regarding that layer, for example that it cannot be removed unless everybody agrees.

4.1 Description

The wooden box carries out the important steps of the auction in a secure way through its physical properties. The box (see Figure 2b) is composed of three columns and seven horizontal plus two inclined layers. Each column (the left, middle and right part of the box) corresponds to one bidder. The top layers $T1$, $T2$ and $T3$ are used to achieve confidentiality of the bid of each bidder, as the marbles (corresponding to the bids) are inserted underneath. The transparent layer $L0$ is used to lock the bids, once they are made, to achieve non-repudiation and non-cancellation. The five lower horizontal layers $L1 - L5$ are used to determine the winning price in a private way. Finally, the two inclined layers are intended to make it impossible to know from which column a marble fell by guiding all of them to the same spot in the bottom left part of the box.

The general idea is as follows: Each bidder will place his bid, modeled by a marble of a certain size, in the top part of the box. We use five different sizes, the smallest one representing the highest possible price, and the biggest one representing the lowest possible price. In the bidding phase, all marbles are inserted into the box onto solid layer $L0$. In the opening phase, layer $L0$ is removed. Below there is layer $L1$ with holes big enough to only let the smallest marbles pass through. Below layer $L1$, there is layer $L2$ with bigger holes (the size of the next biggest marble), and so on. If a bidder inputs the highest possible price, i.e. the smallest marble, this marble will fall through all layers once the solid layer is removed, hence revealing the winning price – but not the winning bidder, thanks to the inclined layers. If nobody inserted the smallest marble, no marble will fall through and the participants can the remove the next layer to check for the second highest price, and so on.

All layers $L0 - L5$ are equipped with four locks, one for each of the three bidders, plus one for the seller. This ensures that a layer can only be removed if all parties agree to do so. Similarly, the removable front side of the box is attached using four locks in the four corners (cf. Figure 3a), one for each bidder plus one for the seller. This allows the parties to inspect the interior of the box before starting the protocol.

The topmost layer consists of three independent parts $T1$, $T2$ and $T3$ that each bidder can use to secure his bid (i.e. his marble inside the box, cf. Figure 3a) from the other participants. Once all bids are inserted, the transparent layer $L0$ is inserted just below and locked by all for parties to ensure non-cancellation (cf. Figure 2b). Once the winning price is determined, the bidders can open their column by removing their lock on Ti and check through the transparent layer if their part of the box is empty or not, i.e. if they won or not (cf. Figure 3c).

Similarly the seller can remove the two inclined layers at the bottom to check if a marble is present inside a column or not (cf. Figure 3d). The first solid layer $L1$ of the price determination part is transparent to allow the participants to check at the start of the protocol if each bidder inserted exactly one marble.

Note also that all participants are always in presence of the box to be able to detect misbehavior of somebody.

The protocol is then broken down into 4 phases:

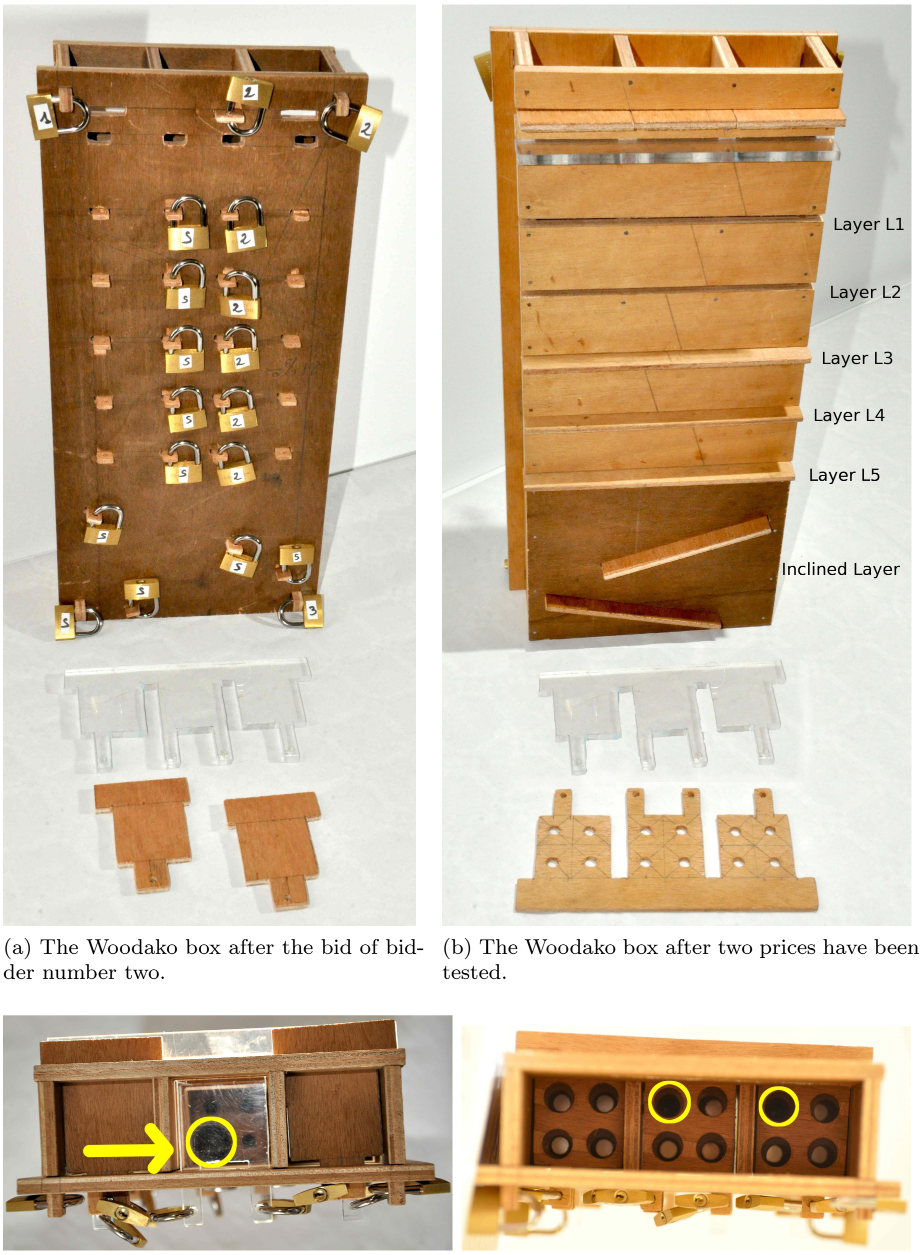
1) Initialization: Each participant can check all the material and see the inside of the box as in Figure 2b to convince himself of the correct design of the machine. The seller gives black marbles of different sizes to each bidder. The smallest marble corresponds to the biggest price and the biggest marble represents the lowest price. Moreover the seller and each bidder have a set of padlocks and keys (as in Figure 3a). Once all bidders have checked the box and received their material (in the case of our prototype five marbles and eight padlocks with keys), the seller closes the box with the front side. The seller and each bidder put a padlock on the box (on each corner of Figure 3b, marked with 1, 2, 3, and S). The seller places the layers $L1 - L5$ in the box, but neither the individual top layers $T1$, $T2$ and $T3$ nor the transparent layer $L0$. The seller also places the two inclined layers in the bottom of the box. Finally, he puts one lock on each layer on the middle column, and all four locks on the inclined layers. He also assigns a column to each bidder.

2) Bidding Phase: Each bidder selects a marble corresponding to the price he wants to bid and puts it in his column without showing the marble to the other parties. He then closes his column using his top layer Ti and secures it using one of his locks. He also puts locks on the five layers $L1 - L5$ below. In Figure 3a you can see the box after bidder number 2 assigned to the middle column has made his bid. Once all bids are made and all locks in place, the seller introduces the transparent plexiglass layer $L0$, i.e. in the hole between the individual top layers and the first full layer $L1$. Finally each participant puts a lock on plexiglass layer $L0$.

3) Opening Phase: The seller and all bidders verify that each bidder inserted exactly one marble by removing the inclined layers (to which the seller has the keys) and looking through the holes of layers $L2 - L5$ and the plexiglass layer $L1$ from below. After the inclined layers have been reinstalled and locked by the seller, all participants remove their lock on the layer $L1$, and the seller removes it. If somebody chose to bid the highest possible price, i.e. insert the smallest possible marble, it will now fall down through all the holes (since all lower layers have bigger holes) and all participants know the winning price, yet not the winner. If no marble falls down, they repeat this process with the next layer below corresponding to the next price. In Figure 3b, we see the back of the box once the two first prices have been tested. The inclined layers are there to hide from which column the marble fell, as all marbles will end up in the bottom left part independently of where they came from (cf. Figure 2b).

4) Verification Phase: Once a marble has fallen down, each bidder can open his lock on his top layer Ti and check if his marble is still inside. In Figure 3c, bidder number two notes that his marble is still inside the box, so he did not win. Similarly the seller can remove the two inclined layers and check for each column, whether there is still a marble inside, hence determining the winner – the column with no marble. An example is given in Figure 3d: the left bidder won since his column is empty, and the two others lost, as their marbles are still there (highlighted by the yellow circles).

Resolving ties: Note that in the case of a tie two or more marbles fall down at the same time. Thus everybody knows that there is a tie, the seller can also identify the tied parties, and the bidders know if they are tied or not.



(a) The Woodako box after the bid of bidder number two.

(b) The Woodako box after two prices have been tested.

(c) Bidder verifiability (i.e. view from top).

(d) Seller verifiability (i.e. view from bottom).

Figure 3: The Woodako box: bidding, determining the winner, and verification

Moreover a tied party can prove to anybody that he is tied by opening his top and showing that his compartment is empty. To resolve the situation either an external tie-breaking mechanism can be used (e.g. rolling a die), or the auction can simply be restarted. Using an external mechanism implies revealing the identity of the tied parties or trusting the seller, since he is the only one who knows who is tied. If privacy is the main concern and the seller is not to be trusted, the auction can simply be restarted and giving the bidders the chance to modify their bids. Sako's protocol (our inspiration) also reveals the identity of the tied parties.

4.2 Security Properties

We now argue how the properties defined in [15, 13] are achieved by our protocol, as long as there is at least one honest party following the protocol (i.e. one bidder or the seller).

Non-cancellation and non-repudiation. Everybody can see in which column a bidder inserted his marble. Due to the fact that the layer $L0$ is locked by all the participants, nobody can change his price during the execution of the protocol. Hence nobody can cancel his bid. Similarly nobody can deny that it was his marble that fell down as the seller and the concerned bidder can verify in which column a marble is still present. Moreover the check at the beginning of the opening phase ensures due to the transparent layer $L1$ that there is exactly one marble per bidder.

Fairness. We consider the two aspects defined in [15]: *1) Highest-Price-Wins:* By the design of the box and the holes of different size in layers $L2 - L5$, the highest price offered by a bidder which is represented by the smallest marble is the first marble to fall down. No bidder can make a larger marble drop before a smaller one. *2) Weak-Non-Interference:* For a given set of bidders no information about the bids is leaked until the end of the bidding phase, since each bidder can choose his marble privately and drop it into the box in such a way that nobody can identify its size.

Privacy. The winner is only known to the seller and himself, but everybody knows the winning price. The inclined layers prevent anybody else from determining the winner by observing from which column the marble fell[3]. Once a marble has dropped, the winner can check if his column is empty by unlocking his top layer Ti and looking inside. The seller can also determine the winner by removing the inclined layers and checking which column is empty as shown in Figure 3d. Since the remaining marbles are too big to fall through the holes, the seller can only see if there is a marble, but will be unable to determine its size, as all marbles have the same color. This preserves the secrecy of the losing bids. The losing bidders can also open their top layers Ti and verify if their marbles are still inside as shown in Figure 3c. This leaks no information about

[3] Note that with two layers as shown in Figure 2b there is a side-channel attack: If the marble falls down in the rightmost column, one can hear the sound of a falling marble only once, whereas in the case of the other two columns the marble falls down twice. However there are some simple solutions: one can extend both layers further to the right so that the marbles fall down twice independently of their original column, or one can use something similar to a "bean machine", i.e. several rows of pins, arranged so that the falling marble hits a pin in each row. The idea is that the marble has a 50% chance of falling down on either side of the pin, hence arrives at a random location on the bottom.

the winner, yet they know the price from the moment when the marble falls as each layer corresponds to a price.

Verifiability. The registration is done at the beginning of the protocol by the seller, and all participants can check if only the registered bidders participate by inserting a marble into the box. Hence the protocol ensures registration verifiability. Outcome verifiability is achieved by the fact that each participant can check the box and the mechanism at the beginning of the protocol, and that each bidder can check at the end whether he lost or won by opening his top layer Ti. The seller can also verify the outcome by opening the bottom of the box.

4.3 Formal Analysis

To formally verify the security properties of the protocol we need to model the properties of the box. We represent its current state by an object denoted `machine(·)`. We use an equational theory to model possible changes to it. A `machine(·)` has the following parameters, where the index i represents a bidder among the n bidders, j a price among the m prices, and s the seller:

- b_{ji} representing the different compartments (for each bidder and price, i.e. above $L1$ to $L5$ for bidder one, two and three in our prototype) of the box, which can be empty or contain a marble of a certain size.

- l_{ji} and l_{js} represent the locks (or rather: the keys necessary to open the locks) that need to be opened to remove a layer Lj (where $j \in \{1, \ldots, n\}$) from the "sieve" part of the machine, one for each bidder and one for the seller.

- t_i are the locks used by the bidders to close the top layer Ti after they inserted their bid.

- p_i and p_s are the locks on the plexiglass layer $L0$.

- b_s represents the locks by the seller on the inclined layers at the bottom (for simplicity we model only one instead of four).

- w_k ($k \in \{1, \ldots, n * m\}$) represent the lower left part of the box where the "winning" marbles that have fallen down end up. We need multiple variables since all marbles fall down if all layers are removed. To simplify the equational theory we have different variables for each price, as this allows us to have independent equations for removing each layer – otherwise we need to take the current state of the w_ks into account, which further increases the number of equations.

We also define the following functions:

- `check_window`: takes as input a machine and returns the w_k to check if a marble has fallen down. This function dose not require any key to be applied.

- `price_j`: takes as input a machine, the keys l_{ji} and l_{js}, and returns a machine where the layer j was removed and potentially marbles have fallen down.

- open_top_i: takes as input a machine and the key t_i, and returns the contents of all b_{ji} for bidder j. This corresponds to the bidder verification check by looking through the plexiglass layer $L0$.

- open_bottom: takes as input a machine and the key b_s, and returns a vector indicating if the columns contain marbles or not. This corresponds to the seller verification check.

- change_top_i: takes as input a machine, all the keys p_k (k among all the bidders), p_s and the single t_i and a new marble to place into bidder i's top compartment.

Consider an example of two bidders and two prices. Suppose the first bidder bids the highest possible price (constant one), and the second bidder bids the lower price two. Then the initial state of the machine m is:

$$m = \texttt{machine}(\texttt{one}, \texttt{two}, \texttt{empty}, \texttt{empty}, l_{11}, l_{12}, l_{1s}, l_{21}, l_{22}, l_{2s}, t_1, t_2, p_1, p_2, p_s, b_s, \texttt{empty}, \texttt{empty}, \texttt{empty}, \texttt{empty})$$

If we compute $m_1 = \texttt{open\_top\_one}(m, l_{11}, l_{12}, l_{1s})$ we obtain

$$m_1 = \texttt{machine}(\texttt{empty}, \texttt{empty}, \texttt{empty}, \texttt{two}, l_{11}, l_{12}, l_{1s}, l_{21}, l_{22}, l_{2s}, t_1, t_2, p_1, p_2, p_s, b_s, \texttt{one}, \texttt{empty}, \texttt{empty}, \texttt{empty})$$

Any party can apply check_window on m_1 to obtain $(\texttt{one}, \texttt{empty}, \texttt{empty}, \texttt{empty})$ and hence observe that a bidder won at price one. The seller can determine that bidder one won by computing $\texttt{open\_bottom}(m_1, b_s) = (\texttt{empty}, \texttt{something})$. Similarly bidder one can check he is the winner by applying $\texttt{open\_top\_1}(m_1, t_1)$ $= (\texttt{empty}, \texttt{empty})$, and bidder two can verify his marble is still in the box using $\texttt{open\_top\_2}(m_1, t_2) = (\texttt{empty}, \texttt{two})$.

Since the number of parameters of the machine depends on the number of bidders and prices, we are unable to define them in a general way in ProVerif. However we have developed a python script that generates the necessary equations for a given number of bidders and prices. For the functions price_j and open_bottom this also consists in enumerating all possible cases based on the possible bid values.

The script also generates a process procMachine that receives the marbles and all the keys from the bidders and the seller, and sends the resulting machine to all participants. They can then execute the "computation" on their copy of the machine, and only need to exchange the keys necessary. Note also that this is an over-approximation since we create many copies of the same machine, which can even evolve differently, although this is not possible in the real world. The ProVerif code for the bidder and seller is straightforward, it is available online [14].

Note that in our model a bidder may insert at most one marble into the box, whereas in the real world he could try to insert several. This may lead to attacks on e.g. non-repudiation or non-cancellation: a bidder could insert two marbles. The seller will observe one marble falling down when the first layer is removed. However, when he opens the bottom of the box to check for the winner, each column still contains a marble. To prevent this, the bidders and the seller check at the beginning if there is exactly one marble per column. Hence we argue that the approximation in our model is correct.

The model allows us to prove in ProVerif that the Woodako protocol ensures *Non-Repudiation* (even if all bidders are corrupted), *Non-Cancellation*, *Weak Non-Interference*, *Highest Price Wins* and *Verifiability*.

When verifying *Privacy* we consider two cases: If the seller is dishonest, the protocol only ensures secrecy of the losing bids, but the winner and winning price are revealed. If the seller is honest, the winner stays anonymous, and only the winning price is revealed. We can prove both results in ProVerif, but the verification takes approximately 24 hours for the dishonest seller and 36 hours for the honest seller case. This is due to the complexity of the equational theory and the equivalence proof. All other properties can be proved within a few seconds. Note that – as above – we only consider the base case (i.e. two bidders) due to the complexity of the equational theory for higher number of bidders and possible prices.

5 Conclusion

We argued that verifiability of an auction must not depend on cryptographic expertise – without understanding, there is no meaningful verifiability. With that in mind, we adapted a suitable cryptographic auction protocol to achieve its security properties *without* cryptography. We began by analyzing Sako's protocol for *Non-Cancellation*, *Non-Repudiation*, *Fairness*, *Verifiability* and *Privacy* using the ProVerif tool. As the protocol mostly passed our automated scrutiny (for privacy, the auctioneers have to be trusted), we took this protocol as a base for the development of our two protocols.

We then proposed the Cardako protocol, an auction protocol inspired by Sako's protocol where each bidder marks their bid by making a hole in their piece of cardboard, which is then put into one envelope. By using a needle, it is possible to detect if a hole is in a certain place (i.e., if the envelope contains a bid for this price), without opening the envelope. We modeled the physical process of testing the envelope for a certain price using equational theory in ProVerif, which allows us to apply the exact same analysis as for the cryptographic protocol. The analysis successfully proved *Non-Repudiation*, *Non-Cancellation*, *Weak Non-Interference*, *Verifiability*, and *Highest Price Wins*. For privacy, an issue was automatically found: dishonest participants may open an envelope or test an envelope for all possible bids. A mitigation is that such actions are readily detectable by all, as any handling of the envelopes occurs in public view.

To improve privacy, we introduced the Woodako protocol. This protocol is again inspired by Sako's protocol, and again replaces cryptography and trusted parties by physical properties. Bids are represented by marbles, where smaller marbles denote higher bids. Bidders place the marble corresponding to their bid in their designated column in a (mechanical) contraption. Then, the first layer below all columns is removed, leaving a new layer with holes the size of the smallest marble. If at least one marble falls through, there is a winner, otherwise this layer is removed and the next layer with larger holes is now the base layer. We argued that Woodako achieves *Non-Repudiation*, *Non-Cancellation*, *Weak Non-Interference*, *Verifiability*, and *Highest Price Wins*. Moreover, this argumentation did not require any expert knowledge to understand, nor did it hinge on correct behavior by trusted parties.

Finally, we formally analyzed the Woodako protocol, again modeling physical properties in equational theory. The model of our physical implementation was proven correct with respect to the mentioned security properties using ProVerif. As the seller knows the winning bidder, a dishonest seller can reveal the winner. As such, we automatically proved privacy for all bidders including anonymity of the winner in case of an honest seller, and simple privacy for losing bidders in case of a dishonest seller.

As future work we look to improve the practicality of our protocols, as they do not scale well for higher numbers of bidders or possible prices, and both require all bidders to be in the same room at least during the winner determination phase.

Moreover, the proofs we currently have for Sako's protocol are not generic in the number of bidders and bids. As future work, we are looking to provide a fully generic proof for the Sako protocol. Similarly, the states space of the Woodako protocol scales too fast for automated proofs with higher numbers of bidders and prices. In this case, we are working towards a reduction proof for the general case.

References

[1] M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *POPL'01*, pages 104–115, 2001.

[2] M. Abdalla, M. Bellare, and G. Neven. Robust encryption. In *Proc. 7th Theory of Cryptography Conference (TCC'10)*, volume 5978 of *LNCS*, pages 480–497. Springer, 2010.

[3] M. Abe and K. Suzuki. Receipt-free sealed-bid auction. In *Proc. 5th Conference on Information Security*, volume 2433 of *LNCS*, pages 191–199. Springer, 2002.

[4] D. Basin, S. Capkun, P. Schaller, and B. Schmidt. Formal Reasoning about Physical Properties of Security Protocols. *ACM Transactions on Information and System Security*, pages 1–28, 2011.

[5] B. Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *Proc. 14th Computer Security Foundations Workshop (CSFW'14)*, pages 82–96. IEEE, June 2001.

[6] B. Blanchet, B. Smith, and V. Cheval. *Proverif Manual*, 1.87beta6 edition, march 2013. `http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf`.

[7] M. Blaze. Toward a broader view of security protocols. In *Proc. Security Protocols workshop 2004*, volume 3957 of *LNCS*, pages 106–120. Springer Verlag, 2006.

[8] F. Brandt. How to obtain full privacy in auctions. *International Journal of Information Security*, 5:201–216, 2006.

[9] Bundesverfassungsgericht (Germany's Federal Constitutional Court). Use of voting computers in 2005 bundestag election unconstitutional. Press release 19/2009 `http://www.bundesverfassungsgericht.de/en/press/bvg09-019en.html`, 2009.

[10] D. Chaum. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security & Privacy*, 2(1):38–47, 2004.

[11] T. Dimkov, W. Pieters, and P. H. Hartel. Portunes: Representing attack scenarios spanning through the physical, digital and social domain. In *ARSPA-WITS'10*, LNCS, 2011.

[12] N. Dong, H. L. Jonker, and J. Pang. Analysis of a receipt-free auction protocol in the applied pi calculus. In *FAST'10*, volume 6561 of *LNCS*, 2011.

[13] J. Dreier, H. L. Jonker, and P. Lafourcade. Defining verifiability in e-auction protocols. In *ASIACCS 2013*, pages 547–552. ACM, 2013.

[14] J. Dreier, P. Lafourcade, and H. Jonker. The proverif code used to automatically verify the examples is available at `http://people.inf.ethz.ch/jdreier/papers/physical-code.zip`, 2014.

[15] J. Dreier, P. Lafourcade, and Y. Lakhnech. Formal verification of e-auction protocols. In *Proc. 2nd Conference on Principles of Security and Trust (POST'13)*, LNCS, 2013.

[16] T. El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Proc. Advances in cryptology - CRYPTO'84*, pages 10–18. Springer, 1985.

[17] R. Fagin, M. Naor, and P. Winkler. Comparing information without leaking it. *Commun. ACM*, 39(5):77–85, May 1996.

[18] R. Küsters, T. Truderung, and A. Vogt. Accountability: definition and relationship to verifiability. In *CCS'10*, pages 526–535. ACM, 2010.

[19] H. Lipmaa, N. Asokan, and V. Niemi. Secure vickrey auctions without threshold trust. In *Proc. 6th Conference on Financial Cryptography*, volume 2357 of *LNCS*, pages 87–101, 2003.

[20] C. Meadows and D. Pavlovic. Formalizing physical security procedures. In *Proc. 8th workshop on Security and Trust Management (STM12)*, volume 7783 of *LNCS*, pages 193–208, 2013.

[21] T. Moran and M. Naor. Polling with physical envelopes: A rigorous analysis of a human-centric protocol. In *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 88–108. Springer, 2006.

[22] T. Moran and M. Naor. Basing cryptographic protocols on tamper-evident seals. *Theor. Comput. Sci.*, 411(10):1283–1310, 2010.

[23] M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. In *Proc. 1st ACM Conference on Electronic Commerce*, pages 129–139, 1999.

[24] K. Omote and A. Miyaji. A practical english auction with one-time registration. In *ACISP*, volume 2119 of *LNCS*, pages 221–234, 2001.

[25] D. C. Parkes, M. O. Rabin, S. M. Shieber, and C. Thorpe. Practical secrecy-preserving, verifiably correct and trustworthy auctions. *Electronic Commerce Research and Applications*, 7(3):294–312, 2008.

[26] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, Feb. 1978.

[27] K. Sako. An auction protocol which hides bids of losers. In *Proc. 3rd Workshop on Practice and Theory in Public Key Cryptosystems*, volume 1751 of *LNCS*, pages 422–432, 2000.

[28] B. Schneier. The solitaire encryption algorithm. `http://www.schneier.com/solitaire.html`, 1999.

[29] F. Stajano and R. J. Anderson. The cocaine auction protocol: On the power of anonymous broadcast. In *Information Hiding*, volume 1768 of *LNCS*, pages 434–447. Springer, 1999.

[30] S. G. Stubblebine and P. F. Syverson. Fair on-line auctions without special trusted parties. In *3rd Conference on Financial Cryptography*, volume 1648 of *LNCS*, pages 230–240, 1999.

[31] S. Subramanian. Design and verification of a secure electronic auction protocol. In *Proc. 17th IEEE Symposium on Reliable Distributed Systems (SRDS'98)*, 1998.