Hawk and Aucitas: e-auction schemes from the Helios and Civitas e-voting schemes

Adam McCarthy[1], Ben Smyth[1], and Elizabeth A. Quaglia[2]

[1] INRIA Paris-Rocquencourt, France
[2] ENS, Paris, France

Abstract. The cryptographic foundations of e-auction and e-voting schemes are similar, for instance, seminal works in both domains have applied mixnets, homomorphic encryption, and trapdoor bit-commitments. However, these developments have appeared independently – for example, the adoption of mixnets in e-voting preceded a similar adoption in e-auctions by over two decades – and the two research communities are disjoint. In this paper, we demonstrate a relation between e-auction and e-voting: we present Hawk and Aucitas, two e-auction schemes derived from the Helios and Civitas e-voting schemes. Our results make progress towards the unification of the e-auction and e-voting domains, thereby paving the way for developments in e-voting to be capitalised upon in the development of e-auctions.

Keywords. Aucitas, auction, bid secrecy, Civitas, collusion resistance, Hawk, Helios, price flexibility, privacy, sealed-bid, verifiability, voting.

1 Introduction

An *e-auction* is a process for the trade of goods and services from *sellers* to *bidders* (or *buyers*), with the aid of an *auctioneer*. We study *sealed-bid auctions* (see Brandt [Bra10] for discussion of other types of auctions), which are defined as follows. First, each bidder submits a *bid* which encapsulates the *price* that the bidder is willing to pay. Secondly, the bids are *opened* to derive the *winning price*. Finally, the *winner* is *revealed*. The winning price and winner are derived in accordance with the auction's policy, for example, in *first-price sealed-bid auctions* the winning price is the highest price bid and the winner is the bidder who bid at the winning price. We shall focus on *Mth price sealed-bid auctions*, which generalise first-price sealed-bid auctions to sell M identical items at the highest price that M bidders are mutually willing to pay. For instance, in the case $M = 6$, six identical items will be sold at the sixth highest price that is bid, because six bidders are mutually willing to pay this price. Such auctions are used to auction telecoms spectrum [Gar13], vehicles [US 13b] and land [US 13a], for example.

An *election* is a decision-making process by which *voters* choose a *representative* from some *candidates*. We study *secret ballot elections* (see Saalfeld [Saa95, §2] for discussion of other types of election), which are defined as follows. First, each voter submits a *ballot* which encapsulates the voter's chosen candidate (i.e., the voter's *vote*). Secondly, all ballots are *tallied* to derive the *distribution of votes*. Finally, the representative is derived in accordance with the election's policy, e.g., in *first-past-the-post elections* (see Lijphart & Grofman [LG84] for discussion of other types of policy) the representative is the candidate with the most votes. In this paper, we shall demonstrate that it is possible to derive e-auction schemes from e-voting schemes.

Constructing e-auction schemes from e-voting schemes. Our translation from an e-voting scheme to an e-auction scheme assumes that prices can be represented as candidates, for example, an e-auction with a *starting price* of 10, *price increments* of 5 and a *price ceiling*[1] of 30 can be represented by the following five candidates: 10, 15, 20, 25 and 30 (we refer to these values as

[⋆] This paper is the long version of [MSQ14].
[1] A price ceiling – that is, an upper bound on the price that may be offered by bidders – is common in e-auctions.

biddable prices). In this setting, an e-auction proceeds as follows. First, to bid for a particular price, bidders "vote" for the candidate that represents the price that the bidder is willing to pay, for example, a bid at price 20 is captured by a "vote" for the third candidate. Secondly, the bids are "tallied" to determine the distribution of "votes" and the winning price is derived from this distribution: the winning price is the largest price in $(10, 15, 20, 25, 30)$ for which at least M bidders "voted" at or above. Finally, we link the winning price to winning bidders. This final step distinguishes our e-auction scheme from the underlying e-voting scheme and we shall see that this can be achieved in the context of secret ballot elections.

1.1 Security properties

Bidders should be able to bid in auctions without fear of repercussions; this property is known as *privacy*. Formulations of privacy depend on the environment and *bid secrecy* has emerged as a *de facto* standard privacy requirement of e-auction schemes in collusion-free environments.

- **Bid secrecy**: A losing bidder cannot be linked to a price.

A stronger formalisation of bid secrecy is *bidder anonymity.*

- **Bidder anonymity**: Bidder identities are not revealed.

Bidder anonymity is useful to hide the identities of bidders and, hence, helps prevent *bid suppression* (that is, a situation in which a conspirator asks a bidder not to submit a bid, perhaps in an attempt to eliminate competing bidders). Intuitively, bidder anonymity is stronger than bid secrecy, since losing bidders cannot be linked to prices in auctions where bidder identities are not revealed. We are also interested in *receipt freeness* and *collusion resistance*[2] (to help prevent *bid rigging* [ZZ10,JL08,HP89] by conspiring bidders), which provide privacy in hostile environments.

- **Receipt freeness**: A losing bidder does not gain information which can be used to prove, to a conspirator, how they bid.
- **Collusion resistance**: A losing bidder cannot collaborate with a conspirator to gain information which can be used to prove how they bid.

Roughly speaking, the above properties correspond to the following properties of e-voting schemes: bid secrecy corresponds to *ballot secrecy*, bidder anonymity corresponds to a notion of *invisible absenteeism*, receipt freeness corresponds to a property with the same name, and collusion resistance corresponds to *coercion resistance* [SB13,Sma12,BHM08,DKR09].

Verifiability allows bidders and observers to verify that bids have been recorded and tallied correctly without trusting the system running the e-auction. The concept is intended to avoid situations whereby systems are trusted and, subsequently, discovered to be untrustworthy, thus bringing auctions into disrepute. We distinguish the following three aspects of verifiability.

- **Outcome verifiability**: A bidder can check that their bid is included in the e-auction and anyone can check that the winning price is valid.
- **Eligibility verifiability**: Anyone can check that all bids were submitted by registered bidders.
- **Non-repudiation**: Anyone can check the winners' identities.

Outcome verifiability is a *de facto* standard requirement of e-auction schemes and eligibility verifiability is important if bidding should be restricted to registered bidders. However, eligibility verifiability typically increases complexity and reduces usability – for instance, an infrastructure for bidders' credentials is assumed by Peng *et al.* [PBDV03] – and has been largely neglected by existing e-auction schemes. Non-repudiation prevents winners from claiming that they did not win. Roughly speaking, outcome verifiability corresponds to *individual* and *universal verifiability* properties of e-voting, and *eligibility verifiability* corresponds to a property with the same name [Smy11,KRS10]. There is no corresponding property for non-repudiation.

We are also interested in the following functional requirement.

[2] Dreier *et al.* [DLL13] refer to collusion resistance as *coercion resistance*, we dislike this terminology, since it is suggestive of coercion rather than collusion.

– **Price flexibility**: Bidders can submit any price.

Price flexibility avoids restricting the bidding amount (although, for practical purposes, we typically assume a starting price of 1, price increments of 1, and the price ceiling to be bounded by the security parameter). Roughly speaking, price flexibility corresponds to the *write-in* property of e-voting.

1.2 Our contribution and motivation

There is an abundance of rich e-voting research which can be capitalised upon to advance e-auctions. Indeed, this statement can be justified with hindsight: Chaum [Cha81] exploited mixnets in e-voting schemes twenty three years before Peng *et al.* [PBDV04] made similar advances in e-auctions (Jakobsson & Juels [JJ00] use mixnets in a distinct manner from Chaum and Peng *et al.*), Benaloh & Fischer [CF85] proposed the use of homomorphic encryption seventeen years before Abe & Suzuki [AS02a], and Okamoto [Oka96] demonstrated the use of trapdoor bit-commitments six years before Abe & Suzuki [AS02b]. In this paper, we demonstrate that e-voting schemes can be used to construct e-auction schemes, thereby paving the way for developments in e-voting to be followed by advances in e-auctions. More concretely, we construct the Hawk and Aucitas Mth price sealed-bid e-auction schemes from the Helios [AMPQ09] and Civitas [CCM08,JCJ05] e-voting schemes. (In addition, Appendix A adapts Hawk to second-price sealed-bid e-auctions.) The Helios and Civitas schemes are particularly significant, since they have both been implemented. Moreover, Helios has been used in real-world elections (e.g., [IAC13,Pri12,AMPQ09]) and Civitas is the only e-voting scheme to satisfy coercion resistance and individual, universal and eligibility verifiability.

Hawk. In the Hawk scheme, prices are encapsulated as bids using an additively homomorphic encryption scheme and the winning price is revealed by decrypting the homomorphic combination of bids. Since individual prices are never decrypted, we have bid secrecy. Moreover, since bidders can ensure that their bid is included in the homomorphic combination and observers can check that the winning price is revealed correctly, we also have outcome verifiability. Bidder anonymity and non-repudiation are conflicting properties: non-repudiation allows the winner's identity to be checked, whereas, bidder anonymity demands that the winner's identity cannot be revealed. Accordingly, we propose two variants of our e-auction scheme, one satisfying non-repudiation and the other satisfying bidder anonymity. Furthermore, we implement the variant satisfying bidder anonymity.

Aucitas. In the Aucitas scheme, prices are encrypted, mixed, and decrypted. Each encrypted price is authenticated by a bidder's credential, however, authenticity is only checked after mixing; this facilitates the use of dummy credentials and prevents a bidder collaborating with a conspirator to prove how they bid (in essence, this is due to the indistinguishability of an encrypted price authenticated by a bidder's credential and an encrypted price authenticated by a dummy credential, which will be rejected after mixing), thereby ensuring collusion resistance. Moreover, the credential system ensures eligibility verifiability and non-repudiation. Furthermore, outcome verifiability is derived from the mixnet. In addition, we achieve price flexibility by defining the biddable prices as $1, 2, \ldots, |\mathfrak{m}|$, where $\mathfrak{m}$ is the encryption scheme's message space.

Our results make progress towards the unification of e-auctions and e-voting.

2 Cryptographic preliminaries

We adopt standard notation for the application of probabilistic algorithms A, namely, $A(x_1, \ldots, x_n; r)$ is the result of running A on input $x_1, \ldots, x_n$ and coins r. Moreover, $A(x_1, \ldots, x_n)$ denotes $A(x_1, \ldots, x_n; r)$, where r is chosen at random. We write $x \leftarrow \alpha$ for the assignment of α to x. Vectors are denoted using boldface, for example, $\mathbf{x}$. We write $|\mathbf{x}|$ to denote the length of a vector $\mathbf{x}$ and $\mathbf{x}[i]$ for the ith component of the vector, where $\mathbf{x} = (\mathbf{x}[1], \ldots, \mathbf{x}[|\mathbf{x}|])$. We extend set membership notation to vectors: we write $x \in \mathbf{x}$ (respectively, $x \notin \mathbf{x}$) if x is an element (respectively, x is not an element) of the set $\{\mathbf{x}[i] : 1 \leq i \leq |\mathbf{x}|\}$.

Let us recall the syntax for *asymmetric encryption schemes.*

Definition 1 (Asymmetric encryption scheme). *An* asymmetric encryption scheme *consists of the following algorithms:*

- *The* key generation algorithm Gen *takes as input the security parameter* 1^k *and outputs a public key* pk*, private key* sk*, and message space* $\mathfrak{m}$*.*
- *The* encryption algorithm Enc *takes as input a public key* pk *and message* $m \in \mathfrak{m}$*. It outputs a ciphertext* c*.*
- *The* decryption algorithm Dec *takes as input a public key* pk*, a private key* sk*, and ciphertext* c*. It outputs a message* m *or the special symbol* $\perp$ *denoting failure.*

Moreover, the scheme must be correct: for all $(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(1^k)$*, messages* $m \in \mathfrak{m}$ *and ciphertexts* $c \leftarrow \mathsf{Enc}(pk, m)$ *we have* $\mathsf{Dec}(pk, sk, c) = m$ *with overwhelming probability. We say an encryption scheme is* homomorphic *if there exists binary operators* $\oplus$*,* $\otimes$ *and* $\odot$ *such that for all* $(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(1^k)$*, messages* $m_1, m_2 \in \mathfrak{m}$ *and coins* r_1 *and* r_2*, we have* $\mathsf{Enc}(pk, m_1; r_1) \otimes \mathsf{Enc}(pk, m_2; r_2) = \mathsf{Enc}(pk, m_1 \odot m_2; r_1 \oplus r_2)$*. The scheme is* additive homomorphic *if* $\odot$ *is the addition operator or* multiplicative homomorphic *if* $\odot$ *is the multiplication operator.*

We abbreviate the standard security notion of *indistinguishability under chosen plaintext attacks* as IND-CPA.

An interactive proof system is a two party protocol between a prover and a verifier on some common input, which allows a claim of membership to be evaluated. Formally, we capture such proof systems as *sigma protocols* (Definition 2) and assume sigma protocols satisfy *special soundness* and *special honest-verifier zero-knowledge* (see [BPW12] for details), in addition to the standard completeness property.

Definition 2 (Sigma protocol). *A* sigma protocol *for an* $\mathcal{NP}$ *language* $\mathcal{L}_R$*, where* $\mathcal{L}_R = \{s \mid \exists\, w \text{ such that } (s, w) \in R\}$*, is a tuple of algorithms* $(\mathsf{Comm}, \mathsf{Chal}, \mathsf{Resp}, \mathsf{Verify})$ *such that:*

- *The* commitment algorithm Comm *takes a statement* s *and witness* w *as input, and outputs a commitment* comm *and some state information* t*.*
- *The* challenge algorithm Chal *outputs a challenge* chal *selected from a fixed challenge space.*
- *The* response algorithm Resp *takes a challenge* chal *and some state information* t *as input, and outputs a response* resp*.*
- *The* verification algorithm Verify *takes a statement* s *and transcript* $(\mathsf{comm}, \mathsf{chal}, \mathsf{resp})$ *as input, and outputs* $\top$ *or* $\perp$*.*

Our e-auction schemes are dependent upon the sigma protocols given in Definition 3.

Definition 3. *Given an asymmetric encryption scheme* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *and a sigma protocol* Σ *for the language* $\mathcal{L}_R$*, we say* Σ*:*

- proves correct key construction *if* $((1^k, pk', \mathfrak{m}'), (sk', r)) \in R \Leftrightarrow (pk', sk', \mathfrak{m}') = \mathsf{Gen}(1^k; r)$
- proves plaintext knowledge in $\mathfrak{M}$ *if* $\mathfrak{M} \subseteq \mathfrak{m}$ *and* $((pk, c, \mathfrak{M}), (m, r)) \in R \Leftrightarrow c = \mathsf{Enc}(pk, m; r) \wedge m \in \mathfrak{M}$
- proves correct ciphertext construction *if* $((pk, c_1, \ldots, c_\ell), (m_1, r_1, \ldots, m_\ell, r_\ell)) \in R \Leftrightarrow \bigwedge_{1 \leq i \leq \ell} c_i = \mathsf{Enc}(pk, m_i; r_i)$
- *is a* plaintext equality test *(PET) if* $((pk, c, c', i), sk) \in R \wedge i \in \{0, 1\} \Leftrightarrow ((i = 0 \wedge \mathsf{Dec}(pk, sk, c) \neq \mathsf{Dec}(pk, sk, c')) \vee (i = 1 \wedge \mathsf{Dec}(pk, sk, c) = \mathsf{Dec}(pk, sk, c'))) \wedge \mathsf{Dec}(pk, sk, c) \neq \perp$
- proves decryption *if* $((pk, c, m), sk) \in R \Leftrightarrow m = \mathsf{Dec}(pk, sk, c)$

where $(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(1^k)$*.*

We can derive *proofs of knowledge* from sigma protocols using the *Fiat-Shamir heuristic* [FS87], which replaces the verifier's challenge with a hash of the prover's commitment, optionally concatenated with the prover's statement [BPW12] and a message.

Definition 4 (Fiat-Shamir transformation). *Given a sigma protocol* $\Sigma = (\mathsf{Comm}_\Sigma, \mathsf{Chal}_\Sigma, \mathsf{Resp}_\Sigma, \mathsf{Verify}_\Sigma)$ *and a hash function* $\mathcal{H}$*, the Fiat-Shamir transformation* $\mathsf{FS}(\Sigma, \mathcal{H}) = (\mathsf{Prove}, \mathsf{Verify})$*, where* Prove *and* Verify *are the algorithms defined as follows:*

– *The* proof algorithm Prove *takes a statement* s*, witness* w*, and (optionally) message* m *as input. The algorithm proceeds as follows. First, compute* $(\mathit{comm}, t) \leftarrow \mathsf{Comm}_\Sigma(s, w)$*. Secondly, derive* chal *as follows: if* m *is defined, then* $\mathit{chal} \leftarrow \mathcal{H}(s, \mathit{comm}, m)$*, otherwise,* $\mathit{chal} \leftarrow \mathcal{H}(s, \mathit{comm})$*. Thirdly, compute* $\mathit{resp} \leftarrow \mathsf{Resp}_\Sigma(\mathit{chal}, t)$*. Finally, output* $\sigma = (\mathit{comm}, \mathit{resp})$*.*
– *The* verification algorithm Verify *takes a statement* s*, candidate proof* $(\mathit{comm}, \mathit{resp})$ *and (optionally) message* m *as input and outputs* $\mathsf{Verify}_\Sigma(s, (\mathit{comm}, \mathit{chal}, \mathit{resp}))$*, where* chal *is derived as follows: if* m *is defined, then* $\mathit{chal} \leftarrow \mathcal{H}(s, \mathit{comm}, m)$*, otherwise,* $\mathit{chal} \leftarrow \mathcal{H}(s, \mathit{comm})$*.*

3 Syntax for e-auction schemes

Based upon Bernhard *et al.* [BCP+11,BPW12,SB13], we formalise *e-auction schemes* as a tuple of algorithms $(\mathsf{Setup}, \mathsf{BB}, \mathsf{Open}, \mathsf{Reveal})$ which are executed by an auctioneer and bidders as follows. (We consider a single auctioneer for simplicity and note that schemes can be generalised to several auctioneers to distribute trust, if necessary.) The Setup algorithm is run by the auctioneer to initialise a key pair and bulletin board. The Bid algorithm is used by bidders to generate their bids and the BB algorithm is used by the auctioneer to process bids, in particular, the algorithm adds correctly formed bids to the bulletin board. Once all of the bids have been collected, the auctioneer runs Open to find the winning price, which is announced by the auctioneer. Finally, the Reveal algorithm is used to identify winners; the Reveal algorithm uses private data s to reveal the winners, for example, s could be a private key which is used to decrypt bids. We define the inputs and outputs of our algorithms below:

$\mathsf{Setup}(1^k) \rightarrow (pk, sk, \mathfrak{bb}, \mathit{aux\text{-}pk})$.The *setup algorithm* Setup takes the security parameter 1^k as input and outputs a public key pk, private key sk, bulletin board $\mathfrak{bb}$ and auxiliary data $\mathit{aux\text{-}pk}$, where $\mathfrak{bb}$ is a set.

$\mathsf{Bid}(pk, \mathit{aux\text{-}pk}, \mathbf{P}, p) \rightarrow b$. The *bid algorithm* Bid takes as input a public key pk, auxiliary data $\mathit{aux\text{-}pk}$, vector of biddable prices $\mathbf{P}$ and price p, where $1 \leq p \leq |\mathbf{P}|$. It outputs a bid b such that $b = \perp$ upon failure.

$\mathsf{BB}(pk, \mathbf{P}, \mathfrak{bb}, b) \rightarrow \mathfrak{bb}'$. The *bulletin board algorithm* BB takes as input a public key pk, vector of biddable prices $\mathbf{P}$, bulletin board $\mathfrak{bb}$ and bid b, where $\mathfrak{bb}$ is a set. It outputs $\mathfrak{bb} \cup \{b\}$ if successful or $\mathfrak{bb}$ to denote failure.

$\mathsf{Open}(pk, sk, \mathbf{P}, \mathfrak{bb}, M) \rightarrow (p, \mathit{aux\text{-}open})$. The *opening algorithm* Open takes as input a public key pk, private key sk, vector of biddable prices $\mathbf{P}$, bulletin board $\mathfrak{bb}$ and parameter M denoting the number of items to be sold, where $\mathfrak{bb}$ is a set and $M > 0$. It outputs the winning price p and auxiliary data $\mathit{aux\text{-}open}$ such that $p = 0$ if no winning price is found and $p = \perp$ upon failure.

$\mathsf{Reveal}(pk, \mathsf{s}, \mathit{aux\text{-}pk}, \mathbf{P}, \mathfrak{bb}, M, p, \mathit{aux\text{-}open}) \rightarrow (w, \mathit{aux\text{-}reveal})$. The *reveal algorithm* Reveal takes as input a public key pk, private data s, auxiliary data $\mathit{aux\text{-}pk}$, a vector of biddable prices $\mathbf{P}$, bulletin board $\mathfrak{bb}$, parameter M denoting the number of items to be sold, winning price p and auxiliary data $\mathit{aux\text{-}open}$, where $M > 0$ and $1 \leq p \leq |\mathbf{P}|$. It outputs a vector of winners w and auxiliary data $\mathit{aux\text{-}reveal}$ such that $w = \perp$ upon failure.

Our definition assumes that a vector of biddable prices $\mathbf{P}$ has been published and a bid for price $\mathbf{P}[p]$ is identified by price index p, where $\mathbf{P}[1] < \cdots < \mathbf{P}[|\mathbf{P}|]$ and $1 \leq p \leq |\mathbf{P}|$. For ease of understanding, we sometimes refer to p as a price.

4 Hawk: An e-auction scheme based on Helios

Hawk is an e-auction scheme derived from the Helios e-voting scheme [AMPQ09] (see Cortier & Smyth [CS13, §2] for a cryptographic description of Helios).

4.1 Informal description

An auction is created by naming an auctioneer. The auctioneer generates a key pair and a proof of correct construction. The auctioneer publishes the public key, proof, biddable prices, and number of items to be sold. The bidding phase proceeds as follows.

Bidding. The bidder creates a bid by encrypting her price with the auctioneer's public key and proving that the ciphertext contains a biddable price. The bidder sends her bid to the auctioneer. The auctioneer authenticates the bidder, checks that she is eligible to bid, and verifies the bidder's proof; if these checks succeed, then the auctioneer publishes the bid on the bulletin board.

After some predefined deadline, the opening and revealing phases commence.

Opening. The auctioneer homomorphically combines the bids, decrypts the homomorphic combination, proves that decryption was performed correctly, and announces the winning price.

Revealing. The auctioneer identifies bids for prices greater than or equal to the winning price, decrypts these bids, and proves that decryption was performed correctly.

Intuitively, every phase of the auction is verifiable. Bidders can check that their bid appears on the bulletin board and, by verifying bidders' proofs, observers are assured that bids represent valid prices. Moreover, anyone can check that the homomorphic combination of bids and decryption were correctly computed. Furthermore, anyone can verify that the decrypted bids contain prices greater than or equal to the winning price. It follows that outcome verifiability is satisfied. In addition, our scheme satisfies bid secrecy, since bids for prices less than the winning price are not decrypted, and also provides non-repudiation, assuming that the auctioneer authenticates the relation between bidders and bids.

4.2 Cryptographic construction

We derive Hawk (Auction Scheme 1) from our informal description using an additively homomorphic encryption scheme satisfying IND-CPA, proofs of correct key construction, proofs of plaintext knowledge, and proofs of decryption. The Setup algorithm generates the auctioneer's key pair, proves correct key construction, and initialises the bulletin board. The Bid algorithm outputs ciphertexts $c_1, \dots, c_{|\mathbf{P}|}$, such that ciphertext c_p contains plaintext 1 and the remaining ciphertexts contain plaintext 0, where $\mathbf{P}[p]$ is the price that the bidder is willing to pay. The algorithm also outputs proofs $\sigma_1, ..., \sigma_{|\mathbf{P}|}$ so that this can be verified. Moreover, it outputs a proof $\sigma_{|\mathbf{P}|+1}$ that the bidder bid for at most one price. The BB algorithm adds correctly formed ballots to the bulletin board. The Open algorithm homomorphically combines ciphertexts representing bids at the highest price and decrypts the homomorphic combination, the algorithm repeats this process for ciphertexts at lower prices, until the sum of the decrypted ciphertexts is equal to or greater than the number of items to be sold, i.e., M. (The sum of the decrypted ciphertexts may be greater than M, since bidders may bid at the same price and, hence, the number of bids eligible to win may be greater than M. Nonetheless, the Reveal algorithm will identify exactly M winning bids.)The Reveal algorithm homomorphically combines a bidder's ciphertexts at or above the winning price, and decrypts the homomorphic combination. The bidder is a winner if the decryption reveals plaintext 1. (We remark that Hawk's open and reveal algorithms are both executed by the auctioneer and could be combined into a single function, however, the separation will be useful in Section 4.3.) We demonstrate an execution of Hawk in Figure 1.

Auction Scheme 1 Hawk

Suppose $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is an additively homomorphic asymmetric encryption scheme satisfying IND-CPA, Σ_1 proves correct key construction, Σ_2 proves plaintext knowledge in $\{0,1\}$ and Σ_3 proves decryption, where Π's message space is $\{0,1\}^*$. Further suppose $\mathcal{H}$ is a hash function and let $\mathsf{FS}(\Sigma_1, \mathcal{H}) = (\mathsf{ProveKey}, \mathsf{VerKey})$, $\mathsf{FS}(\Sigma_2, \mathcal{H}) = (\mathsf{ProveCiph}, \mathsf{VerCiph})$, and $\mathsf{FS}(\Sigma_3, \mathcal{H}) = (\mathsf{ProveDec}, \mathsf{VerDec})$. We define *Hawk* as $\Gamma(\Pi, \Sigma_1, \Sigma_2, \Sigma_3, \mathcal{H}) = (\mathsf{Setup}, \mathsf{Bid}, \mathsf{BB}, \mathsf{Open}, \mathsf{Reveal})$.

$\mathsf{Setup}(1^k)$. Select coins r, compute $(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(1^k; r)$; $\rho \leftarrow \mathsf{ProveKey}((1^k, pk, \mathfrak{m}), (sk, r))$; $\textbf{aux-pk} \leftarrow (1^k, \mathfrak{m}, \rho)$; $\mathfrak{bb} \leftarrow \emptyset$ and output $(pk, sk, \mathfrak{bb}, \textbf{aux-pk})$. .

$\mathsf{Bid}(pk, \textbf{aux-pk}, \mathbf{P}, p)$. Parse $\textbf{aux-pk}$ as $(1^k, \mathfrak{m}, \rho)$, outputting $\perp$ if parsing fails or $\mathsf{VerKey}((1^k, pk, \mathfrak{m}), \rho) \neq \top$. Select coins $r_1, \ldots, r_{|\mathbf{P}|}$ and compute:

```
for $1 \leq i \leq |\mathbf{P}|$ do
    if $i = p$ then $m_i \leftarrow 1$ else $m_i \leftarrow 0$
    $c_i \leftarrow \mathsf{Enc}(pk, m_i; r_i)$; $\sigma_i \leftarrow \mathsf{ProveCiph}((pk, c_i, \{0,1\}), (m_i, r_i), i)$
$c \leftarrow c_1 \otimes \cdots \otimes c_{|\mathbf{P}|}$; $m \leftarrow m_1 \odot \cdots \odot m_{|\mathbf{P}|}$; $r \leftarrow r_1 \oplus \cdots \oplus r_{|\mathbf{P}|}$;
$\sigma_{|\mathbf{P}|+1} \leftarrow \mathsf{ProveCiph}((pk, c, \{0,1\}), (m, r), |\mathbf{P}| + 1)$
```

Output the bid $b = (c_1, \ldots, c_{|\mathbf{P}|}, \sigma_1, \ldots, \sigma_{|\mathbf{P}|+1})$.

$\mathsf{BB}(pk, \mathbf{P}, \mathfrak{bb}, b)$. Parse b as a vector $(c_1, \ldots, c_{|\mathbf{P}|}, \sigma_1, \ldots, \sigma_{|\mathbf{P}|+1})$. If parsing succeeds and $\bigwedge_{i=1}^{|\mathbf{P}|+1} \mathsf{VerCiph}((pk, c_i, \{0,1\}), \sigma_i, i) = \top$, where $c_{|\mathbf{P}|+1} \leftarrow c_1 \otimes \cdots \otimes c_{|\mathbf{P}|}$, then output $\mathfrak{bb} \cup \{b\}$, otherwise, output $\mathfrak{bb}$.

$\mathsf{Open}(pk, sk, \mathbf{P}, \mathfrak{bb}, M)$. Parse $\mathfrak{bb} = \{b_1, \ldots, b_n\}$ as a set of vectors of length $2 \cdot |\mathbf{P}| + 1$, outputting $(\perp, \perp)$ if parsing fails. Initialise index $p \leftarrow |\mathbf{P}| + 1$ and vector $\textbf{aux-open} \leftarrow (\perp, \ldots, \perp)$ of length $|\mathbf{P}|$, and compute:

```
do
    $p \leftarrow p - 1$;
    $c \leftarrow b_1[p] \otimes \cdots \otimes b_n[p]$;
    $m \leftarrow \mathsf{Dec}(pk, sk, c)$; $\textbf{aux-open}[p] \leftarrow \mathsf{ProveDec}((pk, c, m), sk)$;
    $M \leftarrow M - m$
while $M > 0 \wedge p > 0$;
if $M > 0$ then $p \leftarrow 0$
```

Output p and auxiliary data $\textbf{aux-open}$.

$\mathsf{Reveal}(pk, sk, \textbf{aux-pk}, \mathbf{P}, \mathfrak{bb}, M, p, \textbf{aux-open})$. Parse $\mathfrak{bb} = \{b_1, \ldots, b_n\}$ as a set of vectors of length $2 \cdot |\mathbf{P}| + 1$, outputting $(\perp, \perp)$ if parsing fails. Initialise a set $w \leftarrow \emptyset$, vector $\textbf{aux-reveal} \leftarrow (\perp, \ldots, \perp)$ of length n and integer $j \leftarrow 1$, and compute:

```
do
    $c \leftarrow b_j[p] \otimes \cdots \otimes b_j[|\mathbf{P}|]$;
    $m \leftarrow \mathsf{Dec}(pk, sk, c)$; $\textbf{aux-reveal}[j] \leftarrow \mathsf{ProveDec}((pk, c, m), sk)$;
    if $m = 1$ then $w \leftarrow w \cup \{b_j\}$
    $j \leftarrow j + 1$
while $M > |w| \wedge j \leq n$;
```

Output $(w, \textbf{aux-reveal})$.

A comparison of Helios and Hawk. In terms of functionality, the new contribution of Hawk is the introduction of its reveal algorithm, which can be used to link a price to a bidder, given the auctioneer's private key. In addition, we improve efficiency: Hawk's opening algorithm modifies Helios's tallying algorithm, in particular, Hawk only decrypts homomorphic combinations of ciphertexts until the sum of the decrypted ciphertexts is equal to or greater than the number of items to be sold, whereas Helios decrypts all homomorphic combinations of ciphertexts.

4.3 Hawk*: A variant of Hawk with bidder anonymity

Hawk does not satisfy bidder anonymity, since this property conflicts with non-repudiation. However, we can sacrifice non-repudiation in favour of bidder anony-mity, as shown by Hawk* (Auction Scheme 2). In this variant, a winning bidder identifies her ballot b on the bulletin board $\mathfrak{bb}$, computes r such that $b[p] \otimes \cdots \otimes b[|\mathbf{P}|] = \mathsf{Enc}(pk, 1; r)$ – that is, r is a partial homomorphic combination

Fig. 1 An execution of the Hawk auction scheme

Suppose a seller wants to sell two items ($M = 2$) using biddable prices $\mathbf{P} = (10, 15, 20, 25, 30)$. Further suppose that Alice, Bob, Charlie and Daniel bid at prices 25, 20, 10 and 15. An execution of the Hawk auction scheme proceeds as follows.

Bidding. At the end of the bidding phase, the bulletin board is defined as follows.

$$\mathfrak{bb} = \left\{ \begin{array}{l} (\mathsf{Enc}(pk, 0; r_{A,1}), \mathsf{Enc}(pk, 0; r_{A,2}), \mathsf{Enc}(pk, 0; r_{A,3}), \mathsf{Enc}(pk, 1; r_{A,4}), \mathsf{Enc}(pk, 0; r_{A,5})), \\ (\mathsf{Enc}(pk, 0; r_{B,1}), \mathsf{Enc}(pk, 0; r_{B,2}), \mathsf{Enc}(pk, 1; r_{B,3}), \mathsf{Enc}(pk, 0; r_{B,4}), \mathsf{Enc}(pk, 0; r_{B,5})), \\ (\mathsf{Enc}(pk, 1; r_{C,1}), \mathsf{Enc}(pk, 0; r_{C,2}), \mathsf{Enc}(pk, 0; r_{C,3}), \mathsf{Enc}(pk, 0; r_{C,4}), \mathsf{Enc}(pk, 0; r_{C,5})), \\ (\mathsf{Enc}(pk, 0; r_{D,1}), \mathsf{Enc}(pk, 1; r_{D,2}), \mathsf{Enc}(pk, 0; r_{D,3}), \mathsf{Enc}(pk, 0; r_{D,4}), \mathsf{Enc}(pk, 0; r_{D,5})) \end{array} \right\}$$

We omit proofs for brevity and assume that the auctioneer authenticates the relation between bidders and bids to derive non-repudiation. This assumption could be dropped, in the presence of a public key infrastructure, by signing bids.

Opening. The opening phase performs the following decryptions:

$$\begin{array}{l} \mathsf{Dec}(pk, sk, \mathsf{Enc}(pk, 0 \odot 0 \odot 0 \odot 0; r_{A,5} \oplus r_{B,5} \oplus r_{C,5} \oplus r_{D,5})) = 0 \\ \mathsf{Dec}(pk, sk, \mathsf{Enc}(pk, 1 \odot 0 \odot 0 \odot 0; r_{A,4} \oplus r_{B,4} \oplus r_{C,4} \oplus r_{D,4})) = 1 \\ \mathsf{Dec}(pk, sk, \mathsf{Enc}(pk, 0 \odot 1 \odot 0 \odot 0; r_{A,3} \oplus r_{B,3} \oplus r_{C,3} \oplus r_{D,3})) = 1 \end{array}$$

Thereby revealing two bids at the winning price 20. The ciphertexts representing bids at prices 15 and 10 are not decrypted, because the sum of the decrypted ciphertexts (i.e., $0 + 2$) are equal to M.

Revealing. The reveal phase performs the following decryptions:

$$\begin{array}{l} \mathsf{Dec}(pk, sk, \mathsf{Enc}(pk, 0 \odot 1 \odot 0; r_{A,3} \oplus r_{A,4} \oplus r_{A,5})) = 1 \\ \mathsf{Dec}(pk, sk, \mathsf{Enc}(pk, 1 \odot 0 \odot 0; r_{B,3} \oplus r_{B,4} \oplus r_{B,5})) = 1 \\ \mathsf{Dec}(pk, sk, \mathsf{Enc}(pk, 0 \odot 0 \odot 0; r_{C,3} \oplus r_{C,4} \oplus r_{C,5})) = 0 \\ \mathsf{Dec}(pk, sk, \mathsf{Enc}(pk, 0 \odot 0 \odot 0; r_{D,3} \oplus r_{D,4} \oplus r_{D,5})) = 0 \end{array}$$

Alice and Bob are declared winners.

of the coins used by the bidder to construct her bid – and provides b and r as input to the reveal algorithm. Unlike Hawk's reveal algorithm, Hawk*'s reveal algorithm does not take the auctioneer's private key as input and it follows that the algorithm can be executed by the seller, rather than the auctioneer. This introduces the possibility of an e-auction scheme in which the seller learns the winning bidders' identities, but not the auctioneer. We demonstrate an execution of Hawk* in Figure 2.

Intuitively, Hawk*'s reveal algorithm can be optimised by omitting ciphertexts, from the homomorphic combination, for which no bid was made and the corresponding coins can be omitted from the bidders' input (a similar optimisation can be made to Hawk's reveal algorithm). For instance, in our example (Figure 2), the last ciphertext in each bid can be omitted from the homomorphic combination, because the opening phase has shown that no bids were cast at price 30, and the bidders will omit $r_{A,5}$ and $r_{B,5}$ from their partial homomorphic combinations of coins.

Implementation. We implement Hawk*. Our implementation[3] exploits the additive homomorphic [CDS94,CGS97] and distributed decryption [Ped91,CP93] properties of ElGamal [ElG85], and proofs of knowledge demonstrating disjunctive proofs of equality between discrete logarithms [BPW12,AMPQ09,CDS94]. (Note that our implementation distributes trust amongst several auctioneers.) The implementation builds upon the Helios code base, adding approximately 900 lines of Javascript and Python code to ensure that: 1) the coins used to construct the bid are revealed to bidders (the coins are not revealed by Helios to provide some practical resistance against vote selling), enabling bidders to disclose the coins during the reveal phase; 2) auctioneers

[3] Our implementation is available from the following URL: `http://bensmyth.com/publications/2014-Hawk-and-Aucitas-auction-schemes/`.

Auction Scheme 2 Hawk*

Suppose $\Gamma(\Pi, \Sigma_1, \Sigma_2, \Sigma_3, \mathcal{H}) = (\mathsf{Setup}, \mathsf{Bid}, \mathsf{BB}, \mathsf{Open}, \mathsf{Reveal})$, where Π, Σ_1, Σ_2, Σ_3 and $\mathcal{H}$ satisfy the preconditions of Auction Scheme 1. We define *Hawk** as $\Gamma^*(\Pi, \Sigma_1, \Sigma_2, \Sigma_3, \mathcal{H}) = (\mathsf{Setup}, \mathsf{Bid}, \mathsf{BB}, \mathsf{Open}, \mathsf{Reveal}^*)$, where $\mathsf{Reveal}^*(pk, \mathbf{s}, \rho, \mathbf{P}, \mathfrak{bb}, M, p, \textbf{aux-open})$ is defined as follows. If $\mathsf{VerKey}(pk, \rho) = \perp$, then output $(\perp, \perp)$. Otherwise, parse $\mathfrak{bb}$ as a set of vectors of length $2 \cdot |\mathbf{P}| + 1$ and $\mathbf{s}$ as a vector $(b_1, r_1, \ldots, b_M, r_M)$, outputting $(\perp, \perp)$ if parsing fails. Compute:

```
$w \leftarrow \emptyset$;
for $1 \leq i \leq M$ do
    $c \leftarrow \mathsf{Enc}(pk, 1; r_i)$;
    if $b_i \in \mathfrak{bb} \wedge c = b_i[p] \otimes \cdots \otimes b_i[|\mathbf{P}|]$ then $w \leftarrow w \cup \{b_i\}$
```

Output $(w, \perp)$.

Fig. 2 An execution of the Hawk* auction scheme

Suppose the bidding and opening phases of a Hawk* auction are given in Figure 1. The reveal phase proceeds as follows.

Revealing. Alice and Bob reveal $r_{A,3} \odot r_{A,4}$ and $r_{B,3} \odot r_{B,4}$, and the reveal phase checks that $\mathsf{Enc}(pk, 1; r_{A,3}) \otimes \mathsf{Enc}(pk, 0; r_{A,4}) = \mathsf{Enc}(pk, 1; r_{A,3} \odot r_{A,4})$ and $\mathsf{Enc}(pk1; r_{B,3}) \otimes \mathsf{Enc}(pk, 0; r_{B,4}) = \mathsf{Enc}(pk, 1; r_{B,3} \odot r_{B,4})$.

We assume that the bidders' coins are not revealed for bidder anonymity. This assumption could be dropped using designated verifier proofs that demonstrate knowledge of coins.

only provide partial decryptions if the sum of the decrypted ciphertexts is equal to or greater than M.

4.4 Security analysis

Since cryptographic security definitions have not been proposed in the literature (Dreier *et al.* [DLL13,DJL13] propose definitions in the symbolic model), we present an informal security analysis and leave formal analysis as a direction for future work.

Bid secrecy. Hawk and Hawk* derive bid secrecy from the ballot secrecy (i.e., *voters' votes cannot be revealed*) property of Helios. Informally, this can be witnessed up until the reveal phase as follows: by contradiction, suppose a losing bidder can be linked to a price before the winners are revealed; since Hawk and Hawk* are constructed from Helios as per our description in Section 1, it follows immediately that actions in an auction can be mapped to actions in an election and, hence, the link between a losing bidder and a price can be mapped to a relation between a voter and a vote, that is, a voter's vote is revealed, thereby deriving a contradiction. Moreover, it follows from our definition of the Reveal algorithm that the reveal phase only leaks the winners, hence, we conclude that our schemes satisfies bid secrecy: a losing bid cannot be linked to a bidder. We stress that the existence of the reveal algorithm is not prohibited by Helios's ballot secrecy property, because ballot secrecy asserts that an adversary cannot derive the private key[4] and, hence, the reveal algorithm cannot be used by the adversary to obtain an advantage.

Outcome verifiability. Hawk and Hawk* derive outcome verifiability from the individual verifiability (i.e., *a voter can check that her own ballot is published on the election's bulletin board*) and universal verifiability (i.e., *anyone can check that all the votes in the election outcome correspond to ballots published on the election's bulletin board and at most one vote is tallied per ballot*) properties of Helios, in particular, we have "a bidder can check that their bid is included in the

[4] Formally, we can witness that ballot secrecy asserts that the adversary cannot derive the private key from [SB13, Definitions 3 & 5], since the tally algorithm can be used to reveal votes encapsulated inside individual ballots, given the private key.

e-auction" from the individual verifiability property and "anyone can check that the winning price is valid" from the universal verifiability property.

Bidder anonymity/non-repudiation. Hawk satisfies non-repudiation and Hawk* satisfies bidder anonymity. We derive non-repudiation in Hawk under the assumption that the auctioneer authenticates the relation between bidders and bids, hence, a winner's identity can be checked by verifying if the corresponding bid contains plaintext 1 in the homomorphic combination of ciphertexts at or above the winning price. We derive bidder anonymity in Hawk* under the assumption that coins supplied by the winners are not leaked and the auctioneer does not reveal bidder identities, since, in this case, bidder anonymity follows immediately from bid secrecy. Our assumptions can be relaxed using digital signatures for non-repudiation and designated verifier proofs that demonstrate knowledge of coins.

5 Aucitas: An e-auction scheme based on Civitas

Aucitas is an e-auction scheme derived from the Civitas e-voting scheme [CCM08,CCM07], which extends the e-voting scheme by Juels, Catalano & Jakobsson [JCJ02,JCJ05,JCJ10].

5.1 Informal description

An auction is created by naming an auctioneer and registrar. The auctioneer generates a key pair and a proof of correct key construction. The auctioneer publishes the public key, proof, biddable prices, and number of items to be sold. The registration phase proceeds as follows.

Registration. For each eligible bidder, the registrar constructs a (private) credential, sends the credential to the bidder, and derives the public credential by encrypting the credential with the auctioneer's public key.

The registrar authentically publishes the public credentials $\mathbf{L}$ and the bidding phase proceeds as follows.

Bidding. The bidder produces two ciphertexts under the auctioneer's public key: the first contains her price and the second contains her credential. In addition, the bidder proves plaintext knowledge of both ciphertexts. The bidder sends the bid – namely, the ciphertexts and proof – to the auctioneer. The auctioneer verifies the bidder's proof and if verification succeeds, then the auctioneer publishes the bid on the bulletin board.

After some predefined deadline, the opening and revealing phases commence.

Opening. The auctioneer proceeds as follows.

- *Eliminating duplicates:* The auctioneer performs pairwise plaintext equality tests on the ciphertexts containing credentials and discards any bids for which a test holds, that is, bids using the same credential are discarded.
- *Mixing:* The auctioneer mixes the ciphertexts in the bids (i.e., the ciphertexts containing prices and the ciphertexts containing credentials), using the same secret permutation for both mixes, hence, the mix preserves the relation between encrypted prices and credentials. Let $\mathbf{C_1}$ and $\mathbf{C_2}$ be the outputs of these mixes. The auctioneer also mixes the public credentials published by the registrar and assigns the output to $\mathbf{C_3}$.
- *Checking credentials:* The auctioneer discards ciphertexts $\mathbf{C_1}[i]$ from $\mathbf{C_1}$ if there is no ciphertext c in $\mathbf{C_3}$ such that a PET holds for c and $\mathbf{C_2}[i]$, that is, bids cast using ineligible credentials are discarded.
- *Decrypting:* The auctioneer decrypts the remaining encrypted prices in $\mathbf{C_1}$ and proves that decryption was performed correctly.

The auctioneer identifies the winning price from the decrypted prices.

Revealing. The auctioneer identifies ciphertexts $\mathbf{C_1}[i]$ containing prices greater than or equal to the winning price, and performs PETs between $\mathbf{C_2}[i]$ and $\mathbf{L}$ to reveal the identities of winning bidders.

Intuitively, every phase of the auction is verifiable and, hence, outcome and eligibility verifiability, and non-repudiation are derived from the individual, universal and eligibility verifiability properties of Civitas. Moreover, we shall define biddable prices from a starting price of 1 using price increments of 1 and a price ceiling equal to the size of the encryption scheme's message space, hence we have price flexibility. Furthermore, we derive collusion resistance from the coercion resistance property of Civitas.

5.2 Cryptographic construction

For our cryptographic construction of Aucitas, we extend the syntax for e-auctions schemes to include a registration algorithm, hence, an e-auction scheme is a tuple of algorithms (Setup, $\mathsf{Register}$, Bid, BB, Open, Reveal) such that $\mathsf{Register}(pk, \mathit{aux\text{-}pk}) \rightarrow (d, pd)$, where pk is the auctioneer's public key, $\mathit{aux\text{-}pk}$ is auxiliary data, d is a (private) credential, and pd is a public credential. Moreover, we modify the input parameters of Bid, Open and Reveal, namely, $\mathsf{Bid}(d, pk, \mathit{aux\text{-}pk}, \mathbf{P}, p) \rightarrow b$, $\mathsf{Open}(pk, sk, \mathit{aux\text{-}pk}, \mathbf{P}, \mathfrak{bb}, M, \mathbf{L}) \rightarrow (p, \mathit{aux\text{-}open})$ and $\mathsf{Reveal}(pk, sk, \mathit{aux\text{-}pk}, \mathbf{P}, \mathfrak{bb}, p, \mathit{aux\text{-}open}, \mathbf{L})$ $\rightarrow (\mathbf{L}', \mathit{aux\text{-}reveal})$, where d is a bidder's credential, $\mathbf{L}$ and $\mathbf{L}'$ are vectors of public credentials, and the remaining inputs and outputs are as per Section 3. We define a mixnet as $\mathsf{Mix}(\mathbf{c}) \rightarrow (\mathbf{c}', \rho)$ such that $\mathbf{c}'$ contains a permutation of the ciphertexts in $\mathbf{c}$ after re-encryption and ρ is a proof that the mix has been performed correctly. For brevity, we omit a formal definition and refer the reader to Jakobsson, Juels & Rivest [JJR02].

We present Aucitas in Auction Scheme 3. The Setup algorithm generates the auctioneer's key pair using an asymmetric encryption scheme, proves that the key has been correctly constructed, and initialises the bulletin board. The scheme is price flexible using biddable prices $\mathbf{P} = (1, 2, \ldots, |\mathfrak{m}|)$, where $\mathfrak{m}$ is the encryption scheme's message space. The $\mathsf{Register}$ algorithm generates bidders' credentials and we assume that the auctioneer provides the bidder with a credential d corresponding to a public credential $\mathsf{Enc}(pk, d)$; this assumption can be dropped using designated verifier proofs, for example. The specification of the Bid, BB, Open and Reveal algorithms follow from our informal description in Section 5.1. We demonstrate an execution of Aucitas in Figure 3.

Intuitively, collusion resistance is satisfied if a bidder can convince a conspirator that they behaved as instructed, when they actually behaved differently. In Aucitas, this condition is satisfied as follows: given an instruction, a bidder generates a fake credential and follows the instruction using the fake credential. For instance, if the bidder is instructed to bid for a particular price, then the bidder constructs a bid for the price using the fake credential. It follows from the description of Aucitas that this bid will be removed during credential checking, however, the adversary will be unable to detect this, assuming at least one bidder bids at the adversary's price. We acknowledge that price flexibility and collusion resistance are conflicting properties – allowing bidders to submit any price decreases the probability that at least one bidder bids the price instructed by an adversary – and we can balance the degree of price flexibility and collusion resistance by restricting the prices.

A comparison of Civitas and Aucitas. Similarly to Hawk, in terms of functionality, the new contribution of Aucitas is the introduction of its reveal algorithm, which can be used to link a price to a bidder, given the auctioneer's private key. In addition, we improve efficiency: Aucitas's bid algorithm modifies Civitas's vote algorithm by dropping the proof that demonstrates that ciphertext c_1 contains a biddable price. This proof is needed in the election setting to prevent the following attack [JCJ10, §2.1]: an adversary coerces a voter to cast a vote for a random string, this ensures that the adversary can verify if the voter followed instructions by checking that the random string is output by the tallying algorithm and ensures that this vote is not counted, since random strings do not correspond to legitimate candidates. However, this attack is not possible in the e-auction setting, because every message in the encryption scheme's message space corresponds to a price.

Auction Scheme 3 Aucitas

Suppose $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is a homomorphic asymmetric encryption scheme satisfying IND-CPA, Σ_1 proves correct key construction, Σ_2 proves correct ciphertext construction, Σ_3 proves decryption, Σ_4 is a PET, and $\mathcal{H}$ is a hash function. Let $\mathsf{FS}(\Sigma_1, \mathcal{H}) = (\mathsf{ProveKey}, \mathsf{VerKey})$, $\mathsf{FS}(\Sigma_2, \mathcal{H}) = (\mathsf{ProveBind}, \mathsf{VerBind})$, $\mathsf{FS}(\Sigma_3, \mathcal{H}) = (\mathsf{ProveDec}, \mathsf{VerDec})$, and $\mathsf{FS}(\Sigma_4, \mathcal{H}) = (\mathsf{ProvePET}, \mathsf{VerPET})$. We define *Aucitas* below.

$\mathsf{Setup}(1^k)$. Select coins r, compute $(pk, sk, \mathfrak{m}) \leftarrow \mathsf{Gen}(1^k; r); \rho \leftarrow \mathsf{ProveKey}((1^k, pk, \mathfrak{m}), (sk, r)); \mathfrak{bb} \leftarrow \emptyset; \textbf{aux-pk} \leftarrow (1^k, \mathfrak{m}, \rho)$ and output $(pk, sk, \mathfrak{bb}, \textbf{aux-pk})$.

$\mathsf{Register}(pk, \textbf{aux-pk})$. Parse **aux-pk** as $(1^k, \mathfrak{m}, \rho)$, outputting $(\perp, \perp)$ if parsing fails. Assign a random element from $\mathfrak{m}$ to d and compute $pd \leftarrow \mathsf{Enc}(pk, d)$ and output (d, pd).

$\mathsf{Bid}(d, pk, \textbf{aux-pk}, \mathbf{P}, p)$. Parse **aux-pk** as $(1^k, \mathfrak{m}, \rho)$, outputting $\perp$ if parsing fails or $\mathsf{VerKey}((1^k, \mathfrak{m}, \rho), \rho) \neq \top$. Suppose $\mathfrak{m} = \{m_1, \dots, m_{|\mathfrak{m}|}\}$ such that $m_1 < \dots < m_{|\mathfrak{m}|}$. Select coins r_1 and r_2, compute $c_1 \leftarrow \mathsf{Enc}(pk, m_p; r_1); c_2 \leftarrow \mathsf{Enc}(pk, d; r_2); \sigma \leftarrow \mathsf{ProveBind}((pk, c_1, c_2), (m_p, r_1, d, r_2)); b \leftarrow (c_1, c_2, \sigma)$ and output bid b.

$\mathsf{BB}(pk, \mathbf{P}, \mathfrak{bb}, b)$. Parse b as (c_1, c_2, σ). If parsing succeeds and $\mathsf{VerBind}((pk, c_1, c_2), \sigma) = \top$, then output $\mathfrak{bb} \cup \{b\}$, otherwise, output $\mathfrak{bb}$.

$\mathsf{Open}(pk, sk, \textbf{aux-pk}, \mathbf{P}, \mathfrak{bb}, M, \mathbf{L})$. Parse **aux-pk** as $(1^k, \mathfrak{m}, \rho)$ and $\mathfrak{bb} = \{b_1, \dots, b_n\}$ as a set of vectors of length 3, outputting $(\perp, \perp)$ if parsing fails. Proceed as follows.

- $\mathsf{Eliminating\ duplicates}$: Let **aux-dupl** be a vector of length n and **BB** be the empty vector. For each $1 \leq i \leq n$, if there exists σ and $j \in \{1, \dots, i-1, i+1, \dots, n\}$ such that $\sigma \leftarrow \mathsf{ProvePET}((pk, b_i[2], b_j[2], 1), sk)$ and $\mathsf{VerPET}((pk, b_i[2], b_j[2], 1), \sigma) = \top$, then assign $\textbf{aux-dupl}[i] \leftarrow \sigma$, otherwise, compute $\sigma_j \leftarrow \mathsf{ProvePET}((pk, b_i[2], b_j[2], 0), sk)$ for each $j \in \{1, \dots, i-1, i+1, \dots, n\}$ and assign $\textbf{aux-dupl}[i] \leftarrow (\sigma_1, \dots, \sigma_{i-1}, \sigma_{i+1}, \dots, \sigma_n); \mathbf{BB} \leftarrow \mathbf{BB} \parallel (b_i)$, where $\mathbf{BB} \parallel (b_i)$ denotes the concatenation of vectors $\mathbf{BB}$ and (b_i), i.e., $\mathbf{BB} \parallel (b_i) = (\mathbf{BB}[1], \dots, \mathbf{BB}[|\mathbf{BB}|], b_i)$.
- Mixing: Suppose $\mathbf{BB} = (b'_1, \dots, b'_\ell)$, select coins r, and compute $(\mathbf{C_1}, \textbf{aux-mix}_\mathbf{1}) \leftarrow \mathsf{Mix}((b'_1[1], \dots, b'_\ell[1]); r); (\mathbf{C_2}, \textbf{aux-mix}_\mathbf{2}) \leftarrow \mathsf{Mix}((b'_1[2], \dots, b'_\ell[2]); r); (\mathbf{C_3}, \textbf{aux-mix}_\mathbf{3}) \leftarrow \mathsf{Mix}(\mathbf{L})$.
- $\mathsf{Checking\ credentials}$: Let **aux-cred** be a vector of length $|\mathbf{C_2}|$. For each $1 \leq i \leq |\mathbf{C_2}|$, if there exists σ and $c \in \mathbf{C_3}$ such that $\sigma \leftarrow \mathsf{ProvePET}((pk, \mathbf{C_2}[i], c, 1), sk)$ and $\mathsf{VerPET}((pk, \mathbf{C_2}[i], c, 1), \sigma) = \top$, then assign $\textbf{aux-cred}[i] \leftarrow \sigma$, otherwise, compute $\sigma_j \leftarrow \mathsf{ProvePET}((pk, \mathbf{C_2}[i], \mathbf{C_3}[j], 0), sk)$ for each $j \in \{1, \dots, |\mathbf{C_3}|\}$ and assign $\textbf{aux-cred}[i] \leftarrow (\sigma_1, \dots, \sigma_{|\mathbf{C_3}|})$.
- $\mathsf{Decrypting}$: Let *aux-dec* be the empty set. For each $1 \leq i \leq |\mathbf{C_1}|$ such that $|\textbf{aux-cred}[i]| = 1$ assign $\textit{aux-dec} \leftarrow \textit{aux-dec} \cup \{((\mathbf{C_1}[i], \mathbf{C_2}[i]), \sigma, m)\}$, where $m \leftarrow \mathsf{Dec}(pk, sk, \mathbf{C_1}[i])$ and $\sigma \leftarrow \mathsf{ProveDec}((pk, \mathbf{C_1}[i], m), sk)$.

If $|\textit{aux-dec}| < M$, then output $(0, \perp)$. Otherwise, output $(p, \textbf{aux-open})$, where $p \in \{1, \dots, |\mathfrak{m}|\}$ is the largest integer such that M integers in the set $\{m \mid (b, \sigma, m) \in \textit{aux-dec}\}$ are greater than or equal to m_p, and $\textbf{aux-open} \leftarrow (\textbf{aux-dupl}, \textbf{aux-mix}_\mathbf{1}, \textbf{aux-mix}_\mathbf{2}, \textbf{aux-mix}_\mathbf{3}, \textbf{aux-cred}, \textit{aux-dec})$.

$\mathsf{Reveal}(pk, sk, \textbf{aux-pk}, \mathbf{P}, \mathfrak{bb}, M, p, \textbf{aux-open}, \mathbf{L})$. Let $\textit{aux-dec} \leftarrow \textbf{aux-open}[6]$. Parse **aux-pk** as $(1^k, \mathfrak{m}, \rho)$ and *aux-dec* as a set of vectors of length 3, outputting $(\perp, \perp)$ if parsing fails. Suppose $\mathfrak{m} = \{m_1, \dots, m_{|\mathfrak{m}|}\}$ such that $m_1 < \dots < m_{|\mathfrak{m}|}$. If there exist M distinct triples $(b_1, \sigma_1, m'_1), \dots, (b_M, \sigma_M, m'_M) \in \textit{aux-dec}$ and ciphertexts $c_1, \dots, c_M \in \mathbf{L}$ such that for each $1 \leq i \leq M$ we have $\mathsf{VerPET}((pk, b_i[2], c_i, 1), \tau_i) = \top \wedge m'_i \geq m_p$, where $\tau_i \leftarrow \mathsf{ProvePET}((pk, b_i[2], c_i, 1), sk)$, then output $((c_1, \dots, c_M), (\tau_1, \dots, \tau_M))$, otherwise, output $(\perp, \perp)$.

6 Related work

Magkos, Alexandris & Chrissikopoulos [MAC02] and Her, Imamot & Sakurai [HIS05] also study the relation between e-auction and e-voting schemes. Magkos, Alexandris & Chrissikopoulos remark that e-voting and e-auction schemes have a similar structure and share similar security properties. Her, Imamot & Sakurai contrast privacy properties of e-voting and e-auctions, and compare the use of homomorphic encryption and mixnets between domains. Our work is distinguished from these earlier works, since we *demonstrate* a relation between e-auction and e-voting schemes.

Lipmaa, Asokan & Niemi [LAN02] propose an e-auction scheme, based upon homomorphic encryption, which is similar to the e-voting scheme proposed by Damgård, Jurik & Nielsen [DJN10,DJ01] (although the similarities are not explicitly discussed) and Hawk. In essence, their scheme is defined as follows: 1) encrypted bids are sent to the seller during the bidding phase, 2)

Fig. 3 An execution of the Aucitas auction scheme

Suppose a seller wants to sell one item ($M = 1$) using biddable prices $\mathbf{P} = (1, 2, \ldots, |\mathfrak{m}|)$, where $\mathfrak{m} = \{1, \ldots, |\mathfrak{m}|\}$ is the encryption scheme's message space. Further suppose that Alice, Bob and Charlie bid at prices 25, 20 and 15. In addition, let us suppose that an adversary instructs Charlie to bid 20. An execution of the Aucitas auction scheme proceeds as follows.

Registration. The registration phase results in the publication of public credentials $\mathsf{L} = (\mathsf{Enc}(pk, d_A; r_1), \mathsf{Enc}(pk, d_B; r_2), \mathsf{Enc}(pk, d_C; r_3))$.

Bidding. At the end of the bidding phase, the bulletin board is defined as follows.

$$\mathfrak{bb} = \left\{ \begin{array}{l} (\mathsf{Enc}(pk, 25; r_A), \mathsf{Enc}(pk, d_A; \hat{r}_A)), \\ (\mathsf{Enc}(pk, 20; r_B), \mathsf{Enc}(pk, d_B; \hat{r}_B)), \\ (\mathsf{Enc}(pk, 20; r), \mathsf{Enc}(pk, d; \hat{r})), \\ (\mathsf{Enc}(pk, 15; r_C), \mathsf{Enc}(pk, d_C; \hat{r}_C)) \end{array} \right\}$$

We omit proofs for brevity. The ballot $(\mathsf{Enc}(pk, 20; r), \mathsf{Enc}(pk, d; \hat{r}))$ is constructed by Charlie to avoid coercion.

Opening. The opening phase proceeds as follows.

- *Eliminating duplicates.* Since credentials are only used once, no bids are discarded from $\mathfrak{bb}$.
- *Mixing.* The auctioneer mixes the ciphertexts encapsulated in bids, resulting in two vectors: the first vector contains a permutation of $\mathsf{Enc}(pk, 25; r'_A)$, $\mathsf{Enc}(pk, 20; r'_B)$, $\mathsf{Enc}(pk, 20; r')$ and $\mathsf{Enc}(pk, 15; r'_C)$, and the second contains a permutation of $\mathsf{Enc}(pk, d_A; \hat{r}'_A)$, $\mathsf{Enc}(pk, d_B; \hat{r}'_B)$, $\mathsf{Enc}(pk, d; \hat{r}')$ and $\mathsf{Enc}(pk, d_C; \hat{r}'_C)$, using the same permutation in both vectors. In addition, the auctioneer mixes the public credentials, resulting in a vector containing a permutation of $\mathsf{Enc}(pk, d_A; r'_1)$, $\mathsf{Enc}(pk, d_A; r'_2)$ and $\mathsf{Enc}(pk, d_B; r'_3)$.
- *Checking credentials.* The auctioneer discards $\mathsf{Enc}(pk, 20; r')$.
- *Decrypting:* The auctioneer decrypts the remaining encrypted prices to reveal 25, 20 and 15.

The auctioneer identifies the winning price as 25.

Revealing. The auctioneer performs pairwise PETs between $\mathsf{Enc}(pk, d_A; \hat{r}'_A)$ and L to reveal $\mathsf{Enc}(pk, d_A; r_1)$ as the winning bidder's identity.

these encrypted bids are homomorphically combined by the seller in the opening phase and the homomorphic combination is decrypted by the auctioneer, and 3) bidders demonstrate to sellers that they are winning bidders during the reveal phase. Their scheme satisfies bid secrecy under the assumption that either the seller or auctioneer is trusted; by comparision, Hawk assumes that the auctioneer is trusted. This suggests that Hawk requires a stronger trust assumption, however, as we have discussed (Section 3), we can mitigate against the possibility that the auctioneer is dishonest by distributing trust amongst several auctioneers and, hence, the trust assumptions of Hawk and the scheme by Lipmaa, Asokan & Niemi are similar in the case that the seller is also an auctioneer. In addition, Lipmaa, Asokan & Niemi claim that their e-auction scheme could be used to construct an e-voting scheme [LAN02, §9]; by comparision, we focus on the inverse, i.e., the construction of e-auction schemes from e-voting schemes.

Abe & Suzuki [AS02a] propose an e-auction scheme based upon homomorphic encryption. Their scheme satisfies bid secrecy and a complimentary privacy property: with the exception of the winning price, prices are not revealed (this property helps protect bidding strategies, for example). The scheme is similar to Hawk until the opening phase, but differs thereafter, using Jakobsson & Juels's *mix and match* technique [JJ00] to find the winning price, for instance. By contrast, Hawk and Hawk* are conceptually simpler, and Hawk* has the additional property that allows the seller, rather than the auctioneer, to learn the winning bidders' identities.

Peng *et al.* [PBDV04] propose an e-auction schemes based upon mixnets, however, unlike Aucitas, they focus on bid secrecy rather than collusion resistance. Abe & Suzuki [AS02b] introduce an e-auction scheme using trapdoor bit-commitments and Chen, Lee & Kim [CLK03] introduce a scheme using mixnets; these two schemes satisfy collusion resistance. However, Abe & Suzuki assume the existence of a *bidding booth*, where the bidder must bid and cannot communicate with a conspirator, and Chen, Lee & Kim assume the seller is trusted. By comparision, Aucitas achieves collusion resistance without such assumptions.

Acknowledgements. We are particularly grateful to Florian Kerschbaum and the anonymous reviewers who read earlier versions of this paper and provided useful guidance. This work has been partly supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC project *CRYSP* (259639), the ANR-09-VERS-016 BEST project, and Campus France.

A Second-price sealed-bid e-auctions

This paper focuses on first-price sealed-bid e-auctions in which the winning price is the highest price bid and the winner is the bidder who bid at the winning price. Vickrey [Vic61] has shown that first-price auctions motivate bidders to adopt strategies such as *bid shading* (i.e., bidding below their valuation of the item) to maximise their expected utility. *Second-price sealed-bid e-auctions* (also known as *Vickrey auctions*) overcome this problem by defining the winning price as the second-highest price bid, which incentivises bidders to bid their valuation of the item. In this appendix, we show how Hawk can be adapted to second-price e-auctions.

A variant of Hawk for second-price e-auctions We can derive a variant of Hawk to construct second-price auctions by announcing the second-highest price bid, rather than the highest, which can be achieved by re-running the open algorithm with $M = 2$. Since the open algorithm reveals the number of bidders that bid at a particular price, rather than the identities of bidders that bid at a particular price, bid secrecy is preserved. We demonstrate an execution of our variant as follows. Suppose the bidding phase of an auction is given in Figure 1. The initial execution of the open algorithm (parametrised with $M = 1$) performs the first two decryptions given in the opening phase of Figure 1 and the re-run performs the remaining decryption to identify 20 as the winning price (i.e., the second-highest price bid). Finally, the reveal algorithm computes the decryptions $\mathsf{Dec}(pk, sk, \mathsf{Enc}(pk, 1 \odot 0; r_{A,4} \oplus r_{A,5}) = 1$, $\mathsf{Dec}(pk, sk, \mathsf{Enc}(pk, 0 \odot 0; r_{B,4} \oplus r_{B,5}) = 0$, $\mathsf{Dec}(pk, sk, \mathsf{Enc}(pk, 0 \odot 0; r_{C,4} \oplus r_{C,5}) = 0$ and $\mathsf{Dec}(pk, sk, \mathsf{Enc}(pk, 0 \odot 0; r_{D,4} \oplus r_{D,5}) = 0$, and announces Alice as the winner.

References

[AMPQ09] Ben Adida, Olivier de Marneffe, Olivier Pereira, and Jean-Jacques Quisquater. Electing a University President Using Open-Audit Voting: Analysis of Real-World Use of Helios. In *EVT/WOTE'09: Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*. USENIX Association, 2009.

[AS02a] Masayuki Abe and Koutarou Suzuki. M + 1-st price auction using homomorphic encryption. In *PKC'02: 5th International Workshop on Practice and Theory in Public Key Cryptography*, volume 2274 of *LNCS*, pages 115–124. Springer, 2002.

[AS02b] Masayuki Abe and Koutarou Suzuki. Receipt-free sealed-bid auction. In *Information Security*, volume 2433 of *LNCS*, pages 191–199. Springer, 2002.

[BCP$^+$11] David Bernhard, Véronique Cortier, Olivier Pereira, Ben Smyth, and Bogdan Warinschi. Adapting Helios for provable ballot privacy. In *ESORICS'11: 16th European Symposium on Research in Computer Security*, volume 6879 of *LNCS*, pages 335–354. Springer, 2011.

[BHM08] Michael Backes, Cătălin Hriţcu, and Matteo Maffei. Automated Verification of Remote Electronic Voting Protocols in the Applied Pi-calculus. In *CSF'08: 21st Computer Security Foundations Symposium*, pages 195–209. IEEE Computer Society, 2008.

[BPW12] David Bernhard, Olivier Pereira, and Bogdan Warinschi. How Not to Prove Yourself: Pitfalls of the Fiat-Shamir Heuristic and Applications to Helios. In *ASIACRYPT'12: 18th International Conference on the Theory and Application of Cryptology and Information Security*, volume 7658 of *LNCS*, pages 626–643. Springer, 2012.

[Bra10] Felix Brandt. Auctions. In Burton Rosenberg, editor, *Handbook of Financial Cryptography and Security*, pages 49–58. CRC Press, 2010.

[CCM07] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a Secure Voting System. Technical Report 2007-2081, Cornell University, May 2007. Revised March 2008.

[CCM08] Michael R. Clarkson, Stephen Chong, and Andrew C. Myers. Civitas: Toward a Secure Voting System. In *S&P'08: 29th Security and Privacy Symposium*, pages 354–368. IEEE Computer Society, 2008.

[CDS94] Ronald Cramer, Ivan Damgård, and Berry Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. In *CRYPTO'94: 14th International Cryptology Conference*, volume 839 of *LNCS*, pages 174–187. Springer, 1994.

[CF85] Josh Daniel Cohen and Michael J. Fischer. A Robust and Verifiable Cryptographically Secure Election Scheme. In *FOCS'85: 26th Symposium on Foundations of Computer Science*, pages 372–382. IEEE Computer Society, 1985.

[CGS97] Ronald Cramer, Rosario Gennaro, and Berry Schoenmakers. A Secure and Optimally Efficient Multi-Authority Election Scheme. In *EUROCRYPT'97: 16th International Conference on the Theory and Applications of Cryptographic Techniques*, volume 1233 of *LNCS*, pages 103–118. Springer, 1997.

[Cha81] David L. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24:84–90, 1981.

[CLK03] Xiaofeng Chen, Byoungcheon Lee, and Kwangjo Kim. Receipt-Free Electronic Auction Schemes Using Homomorphic Encryption. In *ICISC'03: 6th International Conference on Information Security and Cryptology*, volume 2971 of *LNCS*, pages 259–273. Springer, 2003.

[CP93] David Chaum and Torben P. Pedersen. Wallet Databases with Observers. In *CRYPTO'92: 12th International Cryptology Conference*, volume 740 of *LNCS*, pages 89–105. Springer, 1993.

[CS13] Véronique Cortier and Ben Smyth. Attacking and fixing Helios: An analysis of ballot secrecy. *Journal of Computer Security*, 21(1):89–148, 2013.

[DJ01] Ivan Damgård and Mads Jurik. A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System. In *PKC'01: 4th International Workshop on Practice and Theory in Public Key Cryptography*, volume 1992 of *LNCS*, pages 119–136. Springer, 2001.

[DJL13] Jannik Dreier, Hugo Jonker, and Pascal Lafourcade. Defining verifiability in e-auction protocols. In *ASIA CCS'13: 8th ACM SIGSAC symposium on Information, computer and communications security*, pages 547–552. ACM Press, 2013.

[DJN10] Ivan Damgård, Mads Jurik, and Jesper Buus Nielsen. A Generalization of Paillier's Public-Key System with Applications to Electronic Voting. *International Journal of Information Security*, 9(6):371–385, 2010.

[DKR09] Stéphanie Delaune, Steve Kremer, and Mark D. Ryan. Verifying privacy-type properties of electronic voting protocols. *Journal of Computer Security*, 17(4):435–487, July 2009.

[DLL13] Jannik Dreier, Pascal Lafourcade, and Yassine Lakhnech. Formal verification of e-auction protocols. In *POST'13: 2nd Conference on Principles of Security and Trust*, volume 7796 of *LNCS*, pages 247–266. Springer, 2013.

[ElG85] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[FS87] Amos Fiat and Adi Shamir. How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In *CRYPTO'86: 6th International Cryptology Conference*, volume 263 of *LNCS*, pages 186–194. Springer, 1987.

[Gar13] Juliette Garside. 4g spectrum bidders to be confirmed as auction begins. `http://www.guardian.co.uk/business/2013/jan/06/4g-spectrum-bidders-auction`, 2013. [Online; accessed 9th April 2013].

[HIS05] Yong-Sork Her, Kenji Imamoto, and Kouichi Sakurai. Analysis and comparison of cryptographic techniques in e-voting and e-auction. Technical Report 10(2), Information Science and Electrical Engineering, Kyushu University, September 2005.

[HP89] Kenneth Hendricks and Robert H Porter. Collusion in auctions. *Annales d'Economie et de Statistique*, 15/16:217–230, 1989.

[IAC13] IACR. IACR Elections. `http://www.iacr.org/elections/` (accessed 3 April 2013), 2013.

[JCJ02] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. Cryptology ePrint Archive, Report 2002/165, 2002.

[JCJ05] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. In *WPES'05: 4th Workshop on Privacy in the Electronic Society*, pages 61–70. ACM Press, 2005. See also `http://www.rsa.com/rsalabs/node.asp?id=2860`.

[JCJ10] Ari Juels, Dario Catalano, and Markus Jakobsson. Coercion-Resistant Electronic Elections. In David Chaum, Markus Jakobsson, Ronald L. Rivest, and Peter Y. A. Ryan, editors, *Towards Trustworthy Elections: New Directions in Electronic Voting*, volume 6000 of *LNCS*, pages 37–63. Springer, 2010.

[JJ00] Markus Jakobsson and Ari Juels. Mix and match: Secure function evaluation via ciphertexts. In *Advances in CryptologyASIACRYPT 2000*, pages 162–177. Springer, 2000.

[JJR02] Markus Jakobsson, Ari Juels, and Ronald L. Rivest. Making Mix Nets Robust for Electronic Voting by Randomized Partial Checking. In *11th USENIX Security Symposium*, pages 339–353, 2002.

[JL08] Zhu Ji and KJ Ray Liu. Multi-stage pricing game for collusion-resistant dynamic spectrum allocation. *IEEE Journal on Selected Areas in Communications*, 26(1):182–191, 2008.

[KRS10] Steve Kremer, Mark D. Ryan, and Ben Smyth. Election verifiability in electronic voting protocols. In *ESORICS'10: 15th European Symposium on Research in Computer Security*, volume 6345 of *LNCS*, pages 389–404. Springer, 2010.

[LAN02] Helger Lipmaa, N. Asokan, and Valtteri Niemi. Secure Vickrey Auctions without Threshold Trust. In *FC'02: 6th International Conference on Financial Cryptography and Data Security*, volume 2357 of *LNCS*, pages 87–101. Springer, 2002.

[LG84] Arend Lijphart and Bernard Grofman. *Choosing an electoral system: Issues and Alternatives.* Praeger, 1984.

[MAC02] Emmanouil Magkos, Nikos Alexandris, and Vassilis Chrissikopoulos. A Common Security Model for Conducting e-Auctions and e-Elections. CSCC'02: 6th WSEAS International Multiconference on Circuits, Systems, Communications and Computers `http://www.wseas.us/e-library/conferences/crete2002/papers/444-766.pdf`, 2002.

[MSQ14] Adam McCarthy, Ben Smyth, and Elizabeth A. Quaglia. Hawk and Aucitas: e-auction schemes from the Helios and Civitas e-voting schemes. In *FC'14: 18th International Conference on Financial Cryptography and Data Security*, LNCS. Springer, 2014. To appear.

[Oka96] Tatsuaki Okamoto. An electronic voting scheme. In *Advanced IT Tools: IFIP World Conference on IT Tools*, IFIP Advances in Information and Communication Technology, pages 21–30, 1996.

[PBDV03] Kun Peng, Colin Boyd, Ed Dawson, and Kapali Viswanathan. Five sealed-bid auction models. In *ACSW'03: 21st Australasian Information Security Workshop*, pages 77–86. Australian Computer Society, 2003.

[PBDV04] Kun Peng, Colin Boyd, Ed Dawson, and Kapalee Viswanathan. Efficient implementation of relative bid privacy in sealed-bid auction. In *Information Security Applications*, volume 2908 of *LNCS*, pages 244–256. Springer, 2004.

[Ped91] Torben P. Pedersen. A Threshold Cryptosystem without a Trusted Party. In *EUROCRYPT'91: 10th International Conference on the Theory and Applications of Cryptographic Techniques*, number 547 in LNCS, pages 522–526. Springer, 1991.

[Pri12] Princeton. Helios Princeton Elections. `https://princeton.heliosvoting.org/` (accessed 8 February 2013), 2012.

[Saa95] Thomas Saalfeld. On Dogs and Whips: Recorded Votes. In Herbert Döring, editor, *Parliaments and Majority Rule in Western Europe*, chapter 16. St. Martin's Press, 1995.

[SB13] Ben Smyth and David Bernhard. Ballot secrecy and ballot independence coincide. In *ESORICS'13: 18th European Symposium on Research in Computer Security*, volume 8134 of *LNCS*, pages 463–480. Springer, 2013.

[Sma12] Matthew James Smart. *Anonymity vs. traceability: revocable anonymity in remote electronic voting protocols.* PhD thesis, School of Computer Science, University of Birmingham, 2012.

[Smy11] Ben Smyth. *Formal verification of cryptographic protocols with automated reasoning.* PhD thesis, School of Computer Science, University of Birmingham, 2011.

[US 13a] US Government. 2013 alaska state land offering auction #472. `http://aws.state.ak.us/OnlinePublicNotices/Notices/View.aspx?id=167357`, 2013. [Online; accessed 9th April 2013].

[US 13b] US Government. Auctions; vehicles and surplus property. `http://www.nj.gov/treasury/dss/csdssauc.shtml`, 2013. [Online; accessed 9th April 2013].

[Vic61] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

[ZZ10] Xia Zhou and Haitao Zheng. Breaking bidder collusion in large-scale spectrum auctions. In *MobiHoc'10: 11th ACM international symposium on Mobile ad hoc networking and computing*, pages 121–130. ACM Press, 2010.