Anonymous Sealed-Bid Exchange Mechanism for Energy Market

1st Victor Languille
Télécom Paris
EDF
Paris, France
victor.languille@telecom-paris.fr

2nd Hamza Zarfaoui
Télécom Paris
EDF
Paris, France
hamza.zarfaoui@telecom-paris.fr

I. Introduction

In the future, electricity distribution networks will support a significant and increasing proportion of variable renewable energy sources and local storage assets. They will also be confronted with new load structures due to, for example, the growth of the electric vehicle market. These trends increase the demand for new distribution grid architecture and operating paradigms to maintain grid stability and quality of supply. These paradigms will also allow the provision of cutting-edge new services.

In this context, Jose Horta proposed an architecture with peer to peer exchange of energy between local producers based on a *Multi-Unit Double Auction* mechanism, and implemented using tools such as *Virtual Distribution Grids* and *Blockchain* [1].

The use of *Blockchain* technologies, allowing mutually distrustful users to share a common ledger, was chosen to provide both robustness and transparency to the underlying transactive platform. However, this transparency which permits to each user to verify that transactions amongst the different parties fulfills the commonly accepted specified rules, comes at the cost of privacy. Indeed, the transactive platform being entirely public, any observer can read it and determine to amount of energy sold and purchased by each participant, as the prices at which the exchanges are done. If this second information is already a concern from the commercial point of view, the first one could have very worrying consequences: from the energy consumption of an household, we can infer many things about the lifestyle and habits as the number of its occupants, their standard of living, the moments of the day when they are absent etc.

Fortunately, same kinds of problem have been identified and overcome in the field of cryptocurrency and more widely in the field of blockchain technologies. After the bitcoin emergence at the beginning of the 2010's, researchers took a closer look on the occasionally claimed anonymity of bitcoin. It appeared that bitcoin is *pseudonymous* but not *anonymous*, meaning that even if a user can create as many adresses as he wants, the flow of funds between all addresses is entirely traceable, allowing potential desanonymisation (e.g, by pattern ananlysis). To remedy this, several anonymous cryptocurrency have been proposed, the most notable been Monero and ZCash, both created in 2014, proposing scheme guarantying both the anonymity of the sender an receiver of a transaction, as the confidentiality of the amount of currency exchanged.

In this paper we will reuse the technics used in Zerocash with some modifications, and build upon protocols for sealed-bid exchange mechanism usable in the context of energy market with multi-units double auctions. Doing so, we build a transactive plateform respecting the privacy of its users, answering a question left open in [1]. We will propose two variants of the scheme, regarding the amount of trust we want to put in an auxiliary third party. In the first one the auctioneer is completely trusted, in the second one he is not and he will have to prove is well-behaveness.

II. Preliminaries

We will use different cryptographic building blocks to achieve our privacy goal. We give only informal definitions, referring to the classic literature for the rigorous and standard ones, altogether with the corresponding security definitions.

A. Public-Key Encryption Scheme [2]

Triple of algorithms allowing a sender to send a confidential (i.e encrypted) message m to a receiver, identified with public key, such that no one can read the message without the corresponding private key.

- $\mathsf{KeyGen}(\mathsf{ppparam}, \mathsf{r}) \rightarrow (\mathsf{KeyPub}, \mathsf{KeyPriv})$: Takes as inputs some public parameters pparam and some randomness r. Outputs a couple of public-private key.
- $\mathsf{Enc}(\mathsf{KeyPub}, \mathsf{m}) \rightarrow \mathfrak{C}$: Takes as inputs a public key and a message m. Outputs a cipher text $\mathfrak{C}$.
- $\mathsf{Dec}(\mathsf{KeyPub}, \mathfrak{C}) \rightarrow \mathsf{m}$: Takes as inputs a private key and a cipher $\mathfrak{C}$. Output the original clear text m.

We require the public-key encryption scheme to be IND-CCA1 secure.

B. Signature Scheme [2]

Triple of algorithms allowing a signer, identified with a public key, to proves the authenticity of a message m, adjoining to it a signature that can not be forged by anyone who do not know the corresponding private key.

- $\mathsf{KeyGen}(\mathsf{pparam}, \mathsf{r}) \rightarrow (\mathsf{KeyPub}, \mathsf{KeyPriv})$: Takes as inputs some public parameters pparam and some randomness r. Outputs a couple of public-private key.
- $\mathsf{Sig}(\mathsf{m}, \mathsf{KeyPriv}, \mathsf{r}) \rightarrow \sigma_{\mathsf{m}}$: Takes as inputs some message m, a private key and a nonce r. Outputs a signature σ of $\mathfrak{m}$.
- $\mathsf{Verif}(\mathsf{m}, \sigma, \mathsf{KeyPrub}) \rightarrow b$: Takes as inputs a message m and a public key

C. Pseudorandom Function Family [3]

$\mathsf{PRF} = \{\mathsf{PRF}_{\mathsf{x}} : \{0,1\}^* \rightarrow \{0,1\}^{\mathsf{O}(|\mathsf{x}|)}\}_{\mathsf{x}}$ where x denotes the seed, computationally indistinguishable from random function family.

D. Commitment Scheme [3]

Couple of algorithms allowing a prover to first commit a message m, and in a second time to reveal the message.

- $\mathsf{Com}(\mathsf{m}, \mathsf{r}) \rightarrow \mathsf{cm}$: Takes as inputs a message m and a nonce r. Outputs the commitment com.
- $\mathsf{CheckReveal}(\mathsf{com}, \mathsf{r}, \mathsf{m}) \rightarrow b$; Takes as inputs the commitment cm, the randomness r, the message m and checks that $\mathsf{cm} = \mathsf{Com}(\mathsf{m}, \mathsf{r})$.

We require the commitment scheme to be *Perfectly Hiding* and *Computationally Biding*.

E. Non-Interactive Zero-Knowledge Poof (NIZK) [3]

Couple of algorithms allowing a prover to prove to a verifier that, given an instance $\mathbb{x}$ and a statement given by an NP relation $\mathcal{R}(a, b)$, he knows a witness $\mathbb{w}$ such that $\mathcal{R}(\mathbb{x}, \mathbb{w})$.

- $\mathsf{Setup}(\lambda, \mathcal{R}) \rightarrow \mathsf{pp}_{\mathsf{NIZK}}$: Takes as input a security parameter λ and an NP relation $\mathcal{R}$, output public parameters $\mathsf{pp}_{\mathsf{NIZK}}$.
- $\mathsf{Prove}(\mathsf{pp}_{\mathsf{NIZK}}, \mathbb{x}, \mathbb{w}) \rightarrow \pi$: Takes as input public parameter, instance and witness. Outputs a proof π that $\mathcal{R}(\mathbb{x}, \mathbb{w})$.
- $\mathsf{Verify}(\mathbb{x}, \mathsf{pp}_{\mathsf{NIZK}}, \pi) \rightarrow \mathsf{b}$: Takes as input instance $\mathbb{x}$ and proof π. Outputs 1 if the proof is valid and 0 otherwise.

In addition of *Completness* and *Computanional Zero-Knowledge*, we will require that the NIZK is *Simulation-Extractable* which basically means that an adversary acting as a prover not knowing $\mathbb{w}$ for an instance $\mathbb{x}$ can not produce a valid proof π even if he saw other valid proofs $\pi_1, \pi_2, ..., \pi_n$ for the instance $\mathbb{x}$, see [4] for rigorous definition.

F. Secure Multi-Party Computation (MPC)

Protocol allowing multiple parties $p_1, p_2, ..., p_n$ to jointly compute the output $\mathcal{F}(x_1, x_2, ..., x_n)$ of a function $\mathcal{F}$, while keeping private their respective inputs $x_1, x_2, ..., x_n$.

Typically, those protocols are constituted of several rounds during which the different parties generates some randomness, do some computations, and exchange some messages.

We require the MPC protocol we use to be secure against *Malicious* participants [1].

We do not require the MPC protocol we use to work in the *Asynchronous* model. Indeed, we are interested in blockchain-based exchange protocols, so we can potentially use the underlying blockchain as universal time.

It's important to note that all these primitives have post-quantum alternatives, which can be used if required.

III. Sealed-Bid Exchange Mechanism

Due to our emphasis on privacy concerns, we exclusively concentrate on sealed-bid auctions. These auctions involve participants whose bids are not disclosed to each other.

Our scheme will support general sealed-bid exchange mechanism (SBExM). Classical auction being a particular case where a participant act as a provider of some specific good(s), and the other one provider of money.

A SBExM is specified by the following data:

- A finite set $\mathcal{P}$ of participant. In the following of the paper, we will note n the size of the set $\mathcal{P}$, and the variable i will run amongst the participants.
- A finite set T of asset's type being exchanged. Those asset can represent fungible as non-fungible goods[2]. Typically, one of them will represents monetary unit.
- A set $X \subseteq \mathbb{R}+^{\mathsf{T}}$ of potential outcomes. Here X is simply the set of all possible assignment of quantities $q \in \mathbb{R}$ to all type of asset $t \in \mathsf{T}$. E.g, if t is a fungible good as monetary unit it will be associated to each real value $r \in \mathbb{R}+$, if t is an unique non fungible good it will be associated only to 1. Each element $x \in X$ can be represented as positive real valued vector of size $|\mathsf{T}|$ $x = [\mathbb{q}_{\mathsf{t}}]_{\mathsf{t} \in \mathsf{T}}$. In particular, the element $x \in X$ corresponding a quantity of one unit of the asset t' and no unit of other assets will be noted $x = [\delta_{\mathsf{t}'}(\mathsf{t})]_{\mathsf{t} \in \mathsf{T}}$ where $\delta_{t'}$ is the Dirac function defined by $\delta_{t'}(t) = 1$ if $t = t'$ and $\delta_{t'}(t) = 0$ if $t \neq t'$.
- For each $\mathsf{i} \in \mathcal{P}$, a real-valued function $v_{\mathsf{i}} : X \rightarrow \mathbb{R}$ representing the utility of each participant regarding the different potential outcomes. In most cases, v_{i} will be positive valued function, but it is sometimes usefull to allow negative valuation on some outcome, representing cases were i would prefer get rid of the assets x.
- For each $\mathsf{i} \in \mathcal{P}$, an element $\Gamma_{\mathsf{i}}^{in} \in X$ corresponding to the assets engaged in the exchange by the participant i. We suppose that those provided funds Γ_i^{in} are unknown by all other participant $\mathsf{j} \neq \mathsf{i}$.
- For each $\mathsf{i} \in \mathcal{P}$, a real-valued function $v_{\mathsf{i}} : X \rightarrow \mathbb{R}$ representing the utility of each participant regarding the different potential outcomes. In most cases, v_{i} will be positive valued function, but it is sometimes usefull to allow negative valuation on some outcome, representing cases were i would prefer get rid of the assets x.
- For each $\mathsf{i} \in \mathcal{P}$, a real-valued function $b_{\mathsf{i}} : X \rightarrow \mathbb{R}$, unknown by all other participant $\mathsf{j} \neq \mathsf{i}$, representing the

[1] Also called active adversary

[2] We will use the terms "goods" and "assets" interchangeably for the rest of the paper.

reported bid of each participant, which could be equal or not to v_{i}. Notice that even if T is finite, the set X is an huge infinite set. For practical reason, we have to do the additional assumption that this function b_{i} is presentable in a finite manner. Typically, often this function will be entirely specified by its values on the $[\delta_{\mathsf{t}'}(\mathsf{t})]_{\mathsf{t}\in\mathsf{T}}$ for all $t' \in \mathsf{T}$ and extended by linearity via $v_{\mathsf{i}}([\mathbb{q}_{\mathsf{t}}]_{\mathsf{t}\in\mathsf{T}}) = \sum_{t\in T} \mathbb{q}_t \cdot v_{\mathsf{i}}([\delta_{\mathsf{t}}(\mathsf{t}')]_{\mathsf{t}'\in\mathsf{T}})$, i.e if i assigns the value v to one unit of t, then he assigns the value $\lambda\cdot v$ to λ units of t[3].

- An auctioneer $\mathcal{T}$ conducing the exchange receiving the funds and bids $[\Gamma_i^{in}, b_i]_{i=1}^n$. He also provides the assets $\Gamma_{\mathcal{T}} \in X$ and takes $\Gamma_{\mathcal{T}}^{out} \in X$ from the market.
- A function $\mathcal{F} : \left| \begin{array}{ccc} X \times (X \times \mathbb{R}+^{X})^{\mathcal{P}} & \longrightarrow & X \times X^{\mathcal{P}} \\ (\Gamma_{\mathcal{T}}, [\Gamma_{\mathsf{i}}, b_{\mathsf{i}}]_{\mathsf{i}=1}^n) & \longmapsto & (\Gamma_{\mathcal{T}}^{out}, [\Gamma_i^{out}]_{i=1}^n) \end{array} \right.$, known by all participants, computable by the auctioneer $\mathcal{T}$ taking as input the provided assets $\Gamma_{\mathsf{i}}^{in} = [\mathbb{q}_{t,i}^{in}]_{\mathsf{t}\in\mathsf{T}}$ and the bids b_{i} of each participant together with the assets provided by the auctioneer $\Gamma_{\mathcal{T}}^{in} = [\mathbb{q}_{t,\mathcal{T}}^{in}]_{\mathsf{t}\in\mathsf{T}}$, and returning as output a list of received assets $\Gamma_{\mathsf{i}}^{out} = [\mathbb{q}_{t,i}^{out}]_{\mathsf{t}\in\mathsf{T}}$ of each participant together with the assets taken by the auctioneer $\Gamma_{\mathcal{T}}^{out} = [\mathbb{q}_{t,\mathcal{T}}^{out}]_{\mathsf{t}\in\mathsf{T}}$, and verifying $[\mathbb{q}_{t,\mathcal{T}}^{in} + \sum_{i=1}^n \mathbb{q}_{\mathsf{t},\mathsf{i}}^{in}]_{\mathsf{t}\in\mathsf{T}} = [\mathbb{q}_{t,\mathcal{T}}^{out} + \sum_{i=1}^n \mathbb{q}_{\mathsf{t},\mathsf{i}}^{out}]_{\mathsf{t}\in\mathsf{T}}$, that is it preserves balance in the sense that no goods are created nor destroyed during the exchange.

Because we are interested only in anonymous auction, we can make the hypothesis that $\mathcal{F}$ is symmetric under bids permutation, i.e $\mathcal{F}$ doesn't distinguishes the different participants. Another typical reasonable assumption we can make on $\mathcal{F}$ is that it satisfies $v_{\mathsf{i}}(\Gamma_{\mathsf{i}}) \leq v_{\mathsf{i}}(\Gamma_{\mathsf{i}}^{out}) \forall \mathsf{i} \in \mathcal{P}$, which in words means that no participant can be less satisfied than before the exchange.

A. Multi-Units Double Auctions

The market mechanism proposed in [1] is a form a Multi-Units Double Auctions.

Double auction is a type of auction where both buyers and sellers submit their bids and offers simultaneously, unlike traditional auctions where only one party sets the price. In a multi-units double auction (MDA) of a fungible good, buyers indicate the maximum price per unit they are willing to pay and the total quantity they are willing to purchase, while sellers specify the minimum price per unit they are willing to accept and the total quantity they are willing to sold. The market then matches buyers and sellers based on their respective bids and offers. If a buyer's bid matches or exceeds a seller's offer, a transaction occurs, and the goods or services are exchanged at the agreed-upon price. This mechanism is commonly used in financial markets, avatar commodity markets, and online platforms.

More precisely, multi-unit double auction is a particular case of sealed-bid exchange mechanism with the following specifications:

- The list $\mathcal{P} = [i]_{i=1}^n$ of participants with n elements. It is divided in two disjoint subsets $\mathcal{S} = [\mathsf{j}_{\mathcal{S}}]_{\mathsf{j}_{\mathcal{S}}=1}^{\mathsf{m}_{\mathcal{S}}}$, $\mathcal{B} = [\mathsf{j}_{\mathcal{B}}]_{\mathsf{j}_{\mathcal{B}}=1}^{\mathsf{m}_{\mathcal{B}}}$
- The set $\mathsf{T} = \{\$, \odot\}$ of types is reduced to two elements: the monetary unit \$ and the fungible good's unit $\odot$.
- The set of potential outcome X is the set of couple of reals $(\mathbb{q}_{\$}, \mathbb{q}_{\odot})$.
- For each $\mathsf{i} \in \mathcal{P}$, the set of engaged assets $\Gamma_i \in X$ corresponds to a quantity $(\mathbb{q}_{\$}, 0)_{\mathsf{i}}$ if i is a buyer, and to a quantity $(0, \mathbb{q}_{\odot})_{\mathsf{i}}$ if i is a seller.
- For each $\mathsf{i} \in \mathcal{P}$, the private utility function v_{i} can be arbitrary.
- For each $\mathsf{i} \in \mathcal{P}$, the bid b_{i} is normalised regarding monetary unit such that $b_{\mathsf{i}}((\mathbb{q}_{\$}, \mathbb{q}_{\odot})) = \mathbb{q}_{\$} + b_{\mathsf{i}}((0, \mathbb{q}_{\odot}))$. Moreover we do the additional assumption that $b_{\mathsf{i}}((\mathbb{q}_{\$}, \lambda\mathbb{q}_{\odot})) = \lambda b_{\mathsf{i}}((\mathbb{q}_{\$}, \mathbb{q}_{\odot}))$. So finally b_{i} is entirely specified by its value $\mathbb{p}_{\mathsf{i}} = b_{\mathsf{i}}((0, 1_{\odot}))$ corresponding to the value i attributes to one unit of the fungible good. If i is a seller, this corresponds to the least price at which he would accepts to sell one unit of $\odot$, and if i a buyer, this corresponds to maximum price he would accept to pay for one unit of $\odot$.
- The function $\mathcal{F}$ is completely specified by the algorithm 1 below. In particular it determines the prices $\mathbb{p}_{j_{\mathcal{S}^*}}$ and $\mathbb{p}_{j_{\mathcal{B}^*}}$ at which all the sellers with asked price $\mathbb{p}_i \leq \mathbb{p}_{j_{\mathcal{S}^*}}$ will sell, and all the buyers with proposed price $\mathbb{p}_i \geq \mathbb{p}_{j_{\mathcal{B}^*}}$ will buy.

IV. Privacy

Originally defined as the right to "to be let alone" by Warren and Brandeis [5], privacy refers nowadays to the ability of one person to control (and particularly, restrict) the information known by the other about herself. We will define several privacy guarantees that one could expect from privacy preserving exchange mechanism, and in particular auction protocol, on blockchain.

- *Bidder's Anonimity*: The identity of the bidder is secret.
- *Bidder's Confidentiality*: The bid of the bidder is secret.
- *Winner's Anonimity*[4]: The identity of the winner is secret.
- *Winner's Confidentiality*: The receiving of the winner is secret.

Here, we call "winner" any participant whose assets distribution has been changed during the exchange protocol. The winner anonymity is often a lacking property of blockchain-based anonymous auctions because it can not be realised if the underlying payment scheme is not anonymous itself, and the most used supporting smart-contract blockchain which is Ethereum is not private by design.

Each of those privacy guarantee will come in three flavor, against the other bidders, against the auctioneer and against the Ledger, regarding the kind of entity is considered as Adversary. Note that because the Ledger is public, all security

[3] More general preferences specification are still allowed. E.g one industrial needing at least 100KWatt to keep the machines running can value zero any amount of KWatt inferior to 100, and value positively any amount of KWatt superior or equal to 100.

[4] Notice that in our general setting, an auction can have several winners.

Algorithm 1: Computation of the function $\mathcal{F}$

```
Input: The list $[(\Gamma_i, \mathbb{p}_{\mathsf{i}})]_{i=1}^n$ of participant's provision
       and bids. The provision $\Gamma_{\mathcal{T}} = (0,0)$ of the
       auctioneer.
Output: The list $[\Gamma_i^{out}]_{i=1}^n$ of participant's received
        assets. The taking $\Gamma_{\mathcal{T}}^{out}$ of the auctioneer.
initialisation ;
Divide the list $[(\Gamma_i, \mathbb{p}_i)]_{i=1}^n$ in two lists:
$[(\Gamma_{j_{\mathcal{S}}}, \mathbb{p}_{\mathsf{j}_{\mathcal{S}}})]_{j_{\mathcal{S}}=1}^{n_{\mathcal{S}}} = [((0, \mathbb{q}_{j_{\mathcal{S}},}^{\odot}), \mathbb{p}_{j_{\mathcal{S}}})]_{j_{\mathcal{S}}=1}^{n_{\mathcal{S}}} = \mathcal{S}_{bids}$
$[(\Gamma_{\mathsf{j}_{\mathcal{B}}}, \mathbb{p}_{j_{\mathcal{B}}})]_{j_{\mathcal{B}}=1}^{n_{\mathcal{B}}} = [((\mathbb{q}_{j_{\mathcal{B}},}^{\$}, 0), \mathbb{p}_{j_{\mathcal{B}}})]_{j_{\mathcal{B}}=1}^{n_{\mathcal{B}}} = \mathcal{B}_{bids}$
Order $\mathcal{S}_{\mathsf{bids}}$ in the ascendant price order
  $\mathbb{p}1_{\mathcal{S}} < \mathbb{p}2_{\mathcal{S}} < ... < \mathbb{p}_{n_{\mathcal{S}}}$
Order $\mathcal{B}_{\mathsf{bids}}$ in the descendent price order
  $\mathbb{p}1_{\mathcal{B}} > \mathbb{p}2_{\mathcal{B}} > ... > \mathbb{p}_{n_{\mathcal{B}}}$

Find the critical point $\mathbb{q}^*$ where aggregate offer and
  demand meet.
Let $\mathcal{S}^*$ (resp $\mathcal{B}^*$) be the corresponding seller (resp
  buyer).
Let $Q_{\mathcal{S}} = \sum_{j_{\mathcal{S}}=1}^{n_{\mathcal{S}}} \mathbb{q}_{j_{\mathcal{S}}}^{\odot}$ be the aggregate energy supply.
Let $Q_{\mathcal{B}} = \sum_{j_{\mathcal{B}}=1}^{n_{\mathcal{B}}} \frac{\mathbb{q}_{j_{\mathcal{B}}}^{\$}}{\mathbb{p}_{j_{\mathcal{B}}}}$ be the aggregate energy
  demand.

if $Q_{\mathcal{S}} \leq Q_{\mathcal{B}}$ then
    if $j_{\mathcal{S}} > j_{\mathcal{S}^*}$ then
        $\Gamma_{\mathsf{j}_{\mathcal{S}}}^{out} = \Gamma_{\mathsf{j}_{\mathcal{S}}}$
    end
    else
        $\Gamma_{\mathsf{j}_{\mathcal{S}}}^{out} = (\mathbb{q}_{j_{\mathcal{S}}}^{\odot} \mathbb{p}_{j_{\mathcal{S}^*}}, 0)$
    end
    if $j_{\mathcal{B}} > j_{\mathcal{B}^*}$ then
        $\Gamma^{out}_{\mathsf{j}_{\mathcal{B}} = \Gamma_{\mathsf{j}_{\mathcal{B}}}}$
    end
    else
        $\Gamma_{\mathsf{j}_{\mathcal{B}}}^{out} = (\mathbb{q}_{j_{\mathcal{B}}}^{\$} - \frac{\mathbb{q}_{j_{\mathcal{B}}}^{\$}}{\mathbb{p}_{j_{\mathcal{B}}}} \frac{Q_{\mathcal{S}}}{Q_{\mathcal{B}}} \mathbb{p}_{j_{\mathcal{B}^*}}, \frac{\mathbb{q}_{j_{\mathcal{B}}}^{\$}}{\mathbb{p}_{j_{\mathcal{B}}}} \frac{Q_{\mathcal{S}}}{Q_{\mathcal{B}}})$
    end
end
if $Q_{\mathcal{B}} \leq Q_{\mathcal{S}}$ then
    if $j_{\mathcal{S}} > j_{\mathcal{S}^*}$ then
        $\Gamma_{j_{\mathcal{S}}}^{out} = \Gamma_{j_{\mathcal{S}}}$
    end
    else
        $\Gamma_{\mathsf{j}_{\mathcal{S}}}^{out} = (\mathbb{q}_{j_{\mathcal{S}}}^{\odot} \frac{Q_{\mathcal{B}}}{Q_{\mathcal{S}}} \mathbb{p}_{j_{\mathcal{S}^*}}, \mathbb{q}_{j_{\mathcal{S}}}^{\odot} - \mathbb{q}_{j_{\mathcal{S}}}^{\odot} \frac{Q_{\mathcal{B}}}{Q_{\mathcal{S}}})$
    end
    if $j_{\mathcal{B}} > j_{\mathcal{B}^*}$ then
        $\Gamma_{j_{\mathcal{B}}}^{out} = \Gamma_{j_{\mathcal{B}}}$
    end
    else
        $\Gamma_{j_{\mathcal{B}}}^{out} = (0, \frac{\mathbb{q}_{j_{\mathcal{B}}}^{\$}}{\mathbb{p}_{j_{\mathcal{B}^*}}})$
    end
end
$\Gamma_{\mathcal{T}}^{out} = (\mathbb{p}_{j_{\mathcal{B}^*}} - \mathbb{p}_{j_{\mathcal{S}^*}}) min(Q_{\mathcal{S}}, Q_{\mathcal{B}})$
return $(\Gamma_{\mathcal{T}}^{out}, [\Gamma_i]_{i=1}^n)$
```

properties that do not hold for against the Ledger cannot hold against any other entity.

Here are the privacy guarantees claimed by the protocols we propose:

Privacy Guarantees of the *Protocol 1* used with MPC:

	Other Bidders	Auctioneer	Ledger
Bidder Anonimity	✓	∅	✓
Bidder Confidentiality	✓	∅	✓
Winner Anonimity	✓	∅	✓
Winner Confidentiality	∼	∅	∼

Privacy Guarantees of the *Protocol 2*:

	Other Bidders	Auctioneer	Ledger
Bidder Anonimity	✓	✓	✓
Bidder Confidentiality	✓	×	✓
Winner Anonimity	✓	✓	✓
Winner Confidentiality	∼	×	∼

V. Zerocash-like infrastructure

To obtain a anonymous exchange mechanism, we need to rely on an payment infrastructure anonymous itselfs.

Ordinary cryptocurrencies as Bitcoin or Ethereum are pseudonymous but not anonymous. This means that even if one user can create any number of accounts apriori not directly linked to his personal identity he wants, the list of transactions from the different users' accounts to other ones is entirely public. This is a problem because in practice it is not hard to retrieve the personal identity of a user's account from its historic of transactions [6].

Zerocash is a protocol invented in 2014 to remedy this situation [7], proposing an anonymous and confidential cryptocurrency, where transactions reveals neither their senders and receivers, nor the amount of coin exchanged. Zerocash have been concretely implemented, constituting the core of the cryptocurrency ZCash.

We will reuse the protocol Zerocash, with slight adjustments and modifications, as our base for the SBExM protocols we will present in the next section.

For conciseness reasons, we made the choice to present directly this modified Zerocash protocol with slight adjustments, instead of presenting first the original Zerocash protocol and then our modifications. We will highlight the few places were we modified the original Zerocash protocol. Also, the version we present is quite simplified regarding the actual one, because we prefere to omit some details than can obscure the general comprehension in first lecture. We refer to the original paper for a more in depht and pedagogical presentation [7].

A. Functioning

As a first approximation, it could be said that unlike Bitcoin, which operates on the model of bank accounts being credited and debited, Zerocash's operation is similar to that of fiat money, where banknotes are exchanged from hand to hand; a user's wealth corresponding to the sum of the values of the banknotes they hold. However, the analogy is not perfect: in the transfer of a note from a user A to a user B corresponds to the destruction of one or several old notes owned by A and the creation of one several new notes owned by B.

The underlying infrastructure which will be unchanged between the different versions of the protocol is the following slight modification of Zerocash.

Setup: -Participants share a common *append-only public ledger* $\mathfrak{L}$. This ledger can typically be implemented as distributed ledger such as a blockchain.

-Participants share a commitment scheme $\mathsf{Com}(\mathsf{m}, \mathsf{r})$ and a Non-Interactive Zero-Knowledge argument(NIZK) protocol.

-Each user u creates a *private/public address* pair $(\mathsf{addr}_{\mathsf{pk,u}}, \mathsf{addr}_{\mathsf{sk,u}}) = (\mathsf{r}, \mathsf{H}(\mathsf{r}))$ by generating a random number r and calculating its hash. It keeps its *private address* $\mathsf{addr}_{\mathsf{sk,u}} = \mathsf{r}$ secret and sends its *public address* $\mathsf{H}(\mathsf{r})$ to other users.

Notes: Each participant owns a wallet containing *notes*. Together, those *notes* constitute his fortune. A *note* $\mathbf{n}$ is a set of data $\mathbf{n} = (\Gamma, \mathsf{addr}_{\mathsf{pk,u}}, \mathsf{r}, \rho, \mathsf{cm})$ where:

- Γ is an tuple $[\mathbb{q}_t]_{\mathsf{t} \in \mathsf{T}}$ with $\forall \mathsf{t} \in \mathsf{T} \mathbb{q}_t \in \mathbb{R}+$, corresponding to the values of the different asset embodied by the *note*[5].
- $\mathsf{addr}_{\mathsf{pk,u}}$ is the *public address* of the owner u of the *note*.
- ρ is a secret nonce used to generate the *note*.
- r a random number used to generate the commitment.
- cm is the *commitment* $\mathsf{cm} = \mathsf{Com}(\mathsf{v}||\mathsf{addr}_{\mathsf{pk,u}}||\rho, \mathsf{r})$ of the *note*[6], computed and added to the ledger at the creation of the *note*.

Amongst those data, only the *commitment* cm will appears to the ledger. The data $(\Gamma, \mathsf{addr}_{\mathsf{pk,u}}, \mathsf{r}, \rho)$ are kept secret by the owner u, so we refer to them as the private data of the note. Remark that a *public address* suffice to compute cm, this will permit to a user A to generate a note for another user B as we will detail in the next part.

To a *note* $\mathbf{n}$ is also attached a serial number sn. This serial number is computed from the private data of the note and the secret address of the user u as $\mathsf{sn} = \mathsf{PRF}_{\mathsf{addr}_{\mathsf{sk,u}}}(\rho)$. This serial number is computed and added the ledger at the spending/destruction of the *note*. Remark that a *private address* is necessary to compute sn, this will prevent any user different from B to consume a note owner by B. Because this serial number is derived from a random nonce ρ by a PRF, it is unique to a note. It prevents double-spending as we will see in the next part.

The ledger $\mathcal{L}$ maintains two lists SnList and CmList respectively listing the serial number sn and the commitment cm of all notes $\mathbf{n}$ respectively spent and created to date.

Transactions
Transfer of assets from A to B is done thought *transactions*.

A *transaction* from a sender A to a receiver B corresponds to a tuple of data $\mathsf{tx} = ([\mathsf{sn}_{\mathsf{i}}^{\mathsf{old}}]_{\mathsf{i}=1}^{\mathsf{n}}], [\mathsf{cm}_{\mathsf{j}}^{\mathsf{new}}]_{\mathsf{j}=1}^{\mathsf{m}}, \pi)$ computed and added by A to the ledger, consuming old notes $[\mathbf{n}_{\mathsf{i}}^{\mathsf{old}}]_{\mathsf{i}=1}^{\mathsf{n}}$ owned by A and creating new notes $[\mathbf{n}_{\mathsf{j}}^{\mathsf{new}}]_{\mathsf{j}=1}^{\mathsf{m}}$ owned by B. More precisely, the computation of a transaction proceeds as follows:

-The owner A of the *notes* $[\mathbf{n}_{\mathsf{i}}^{\mathsf{old}}]_{i=1}^{n} = [(\Gamma_i^{\mathsf{old}}, \mathsf{addr}_{\mathsf{pk_A}}^{\mathsf{old}}, \rho_{\mathsf{i}}^{\mathsf{old}}, \mathsf{r}_{\mathsf{i}}^{\mathsf{old}})]_{i=1}^{n}$, whose *commitments* $[\mathsf{cm}_{\mathsf{i}}^{\mathsf{old}}]_{i=1}^{n}$ was previously added to the list CmList, creates new *notes* $[\mathbf{n}_{\mathsf{j}}^{\mathsf{new}}]_{j=1}^{m} = [(\Gamma_{\mathsf{j}}^{\mathsf{new}}, \mathsf{addr}_{\mathsf{pk,B}}^{\mathsf{new}}, \rho_{\mathsf{j}}^{\mathsf{new}}, \mathsf{r}_{\mathsf{j}}^{\mathsf{new}})]_{j=1}^{m}$ and calculates their *commitments* $[\mathsf{cm}^{\mathsf{new}}]_{\mathsf{j}=1}^{\mathsf{m}} = [\mathsf{Com}(\Gamma_{\mathsf{j}}^{\mathsf{new}}||\mathsf{addr}_{\mathsf{pk,B}}^{\mathsf{new}}||\rho_{\mathsf{j}}^{\mathsf{new}}, \mathsf{r}_{\mathsf{j}}^{\mathsf{new}})]_{j=1}^{m}$.

-With knowledge of the data of the $[\mathbf{n}_{\mathsf{i}}^{\mathsf{old}}]_{i=1}^{n}$ and $\mathsf{addr}_{\mathsf{sk,A}}$, A calculates the serial numbers $[\mathsf{sn}_{\mathsf{i}}^{\mathsf{old}}]_{\mathsf{i}=1}^{\mathsf{n}} = [\mathsf{PRF}_{\mathsf{addr}_{\mathsf{sk,A}}}(\rho_{\mathsf{i}}^{\mathsf{old}})]_{i=1}^{n}$.

-A then sends to the ledger the transaction tx, consisting of a triplet $\mathsf{tx} = ([\mathsf{sn}_{\mathsf{i}}^{\mathsf{old}}]_{\mathsf{i}=1}^{\mathsf{n}}], [\mathsf{cm}_{\mathsf{j}}^{\mathsf{new}}]_{\mathsf{j}=1}^{\mathsf{m}}, \pi)$, where π is an NIZK defined by the following statement, instance and witness:

Instance:	**Witness**:
CmList	$[\mathsf{a}_{\mathsf{sk,i}}^{\mathsf{old}}]_{i=1}^{n}$
$[\mathsf{sn}_{\mathsf{i}}^{\mathsf{old}}]_{\mathsf{i}=1}^{\mathsf{n}}$	$[\mathbf{n}_{\mathsf{i}}^{\mathsf{old}}]_{\mathsf{i}=1}^{\mathsf{n}} = (\Gamma_{\mathsf{i}}^{\mathsf{old}}, \mathsf{a}_{\mathsf{pk,i}}^{\mathsf{old}}, \rho_{\mathsf{i}}^{\mathsf{old}}, \mathsf{r}_{\mathsf{i}}^{\mathsf{old}}, \mathsf{cm}_{\mathsf{i}}^{\mathsf{old}})$
$[\mathsf{cm}_{\mathsf{j}}^{\mathsf{new}}]_{\mathsf{j}=1}^{\mathsf{m}}$	$[\mathbf{n}_{\mathsf{j}}^{\mathsf{new}}]_{\mathsf{j}=1}^{\mathsf{m}} = (\Gamma_{\mathsf{j}}^{\mathsf{new}}, \mathsf{a}_{\mathsf{pk,j}}^{\mathsf{new}}, \rho_{\mathsf{j}}^{\mathsf{new}}, \mathsf{r}_{\mathsf{j}}^{\mathsf{old}}, \mathsf{cm}_{\mathsf{j}}^{\mathsf{ne}}$

Statement:

$[\mathsf{cm}_{\mathsf{i}}^{\mathsf{old}}]_{\mathsf{i}=1}^{\mathsf{n}} \subseteq \mathsf{CmList}$: The consumed notes have previously been added to the ledger.

$[\mathsf{addr}_{\mathsf{pk,i}}^{\mathsf{old}}]_{\mathsf{i}=1}^{\mathsf{n}} = [\mathsf{H}(\mathsf{addr}_{\mathsf{sk,i}}^{\mathsf{old}})]_{\mathsf{i}=1}^{\mathsf{n}}$: The public address is computed from the private address.

$[\mathsf{sn}_{\mathsf{i}}^{\mathsf{old}}]_{\mathsf{i}=1}^{\mathsf{n}} = [\mathsf{PRF}_{\mathsf{addr}_{\mathsf{sk,A}}}(\rho_{\mathsf{i}}^{\mathsf{old}})]_{i=1}^{n}$: The serial number are correctly computed.

$[\mathsf{cm}_{\mathsf{i}}^{\mathsf{old}}]_{\mathsf{i}=1}^{\mathsf{n}} = [\mathsf{Com}(\Gamma_{\mathsf{i}}^{\mathsf{old}}||\mathsf{addr}_{\mathsf{pk,A}}^{\mathsf{old}}||\rho_{\mathsf{i}}^{\mathsf{old}}||, \mathsf{r}_{\mathsf{i}}^{\mathsf{old}})]_{\mathsf{i}=1}^{\mathsf{n}}$: The commitments of consumed notes are well-formed.

$[\mathsf{cm}_{\mathsf{j}}^{\mathsf{new}}]_{\mathsf{j}=1}^{\mathsf{m}} = [\mathsf{Com}(\Gamma_{\mathsf{j}}^{\mathsf{new}}||\mathsf{a}_{\mathsf{pk,B}}^{\mathsf{new}}||\rho_{\mathsf{j}}^{\mathsf{new}}, \mathsf{r}_{\mathsf{j}}^{\mathsf{new}})]_{\mathsf{j}=1}^{\mathsf{m}}$: The commitments of created notes are well-formed.

$\sum_{i=1}^{n} \Gamma_i^{old} = \sum_{j=1}^{m} \Gamma_j^{new}$: The balance is preserved (i.e no values are created nor destroyed)[7].

Receiving a transaction $\mathsf{tx} = ([\mathsf{sn}_{\mathsf{i}}^{\mathsf{old}}]_{\mathsf{i}=1}^{\mathsf{n}}, [\mathsf{cm}_{\mathsf{j}}^{\mathsf{new}}]_{\mathsf{j}=1}^{\mathsf{m}}, \pi)$, the ledger checks that:

- The proof π is valid.

[5] Here there is a first difference with Zerocash which only support coins value. We simply generalised the unique real $\mathbb{v}$ of Zerocash by a general tuple of real.

[6] Here there is a second slight difference with the protocol zerocash. The commitment is directly computed from the other private data as in ZEXE. The symbol || corresponding to the concatenation

[7] Here another little difference with the original Zerocash which checks only that the balance of value of coins (the only asset circulating) is preserved.

- None of the serial numbers $[\mathsf{sn}_\mathsf{i}^\mathsf{old}]_{\mathsf{i}=1}^\mathsf{n}$ appears in the list SnList (thus avoiding double-spending).

If those two conditions are verified, the ledger validates the transaction, adding the *serial numbers* $[\mathsf{sn}_\mathsf{i}^\mathsf{old}]_{\mathsf{i}=1}^\mathsf{n}$ to SnList, the *commitments* $[\mathsf{cm}^\mathsf{new}]_{\mathsf{j}=1}^\mathsf{m}$ to CmList and the *transaction* tx to TxList.

VI. Protocols

We will propose two different protocols regarding the amount of trust we want to put in the trusted third party/auctioneer $\mathcal{T}$ we include in the protocol. Namely, we will consider the following cases:

- **Absolutely Trusted Third Party**: The users are a priori convinced that $\mathcal{T}$ follows strictly the specified protocol, executing correctly each mandatory action and avoiding any action which are not specified as possible. In particular, $\mathcal{T}$ do not communicate with users if it is not specified, avoiding potential collusion. It is indifferent that such party learns private data of participants.
- **Almost not Trusted Third Party**: The users are not a priori convinced that $\mathcal{T}$ follows strictly the specified protocol. Users considers a priori that $\mathcal{T}$ could deviates either by not executing or misexecuting some specified actions, or by executing additional actions. In particular, users could suspect collusion via extra-communication between $\mathcal{T}$ and other participants. It is desirable that such party learns the less private data of participants as possible.

In the following protocols, we will consider the list $[\mathcal{S}_\mathsf{i}]_{i=1}^{n}$ of the n sellers, and $[\mathcal{B}_\mathsf{j}]_{j=1}^{m}$ the list of the m buyers. Let $[(\mathcal{S}_\mathsf{i}, \mathsf{p}_\mathsf{i}, \mathsf{q}_\mathsf{i})]_{i=1}^{n}$ be the list of the triples of requested prices and quantities by the sellers, and let $[(\mathcal{B}_\mathsf{j}, \mathsf{b}_\mathsf{j}, \mathsf{q}_\mathsf{j})]_{j=1}^{m}$ be the list of the triples of proposed bids and quantities by the buyers.

Each protocol starts with an registration phase where the participants i have a time delay Δ to manifest their will to participate the auction iteration, communicating their bid/ask to the third party following the modality we will specify.

A. Protocol 1 (Absolutely Trusted Third Party)

Registration Phase:

The ledger opens auction at time t_1, letting a delay Δ_1 specified by the protocol to the participants to register.

Each participant i sends to the third party $\mathcal{T}$ his bid/ask $(\perp_\mathsf{i}, \mathbb{p}_\mathsf{i}, \mathbb{q}_\mathsf{i})$ together with m notes $[\mathsf{n}_{\mathsf{i,j}}^\mathsf{old}]_{\mathsf{j}=1}^{\mathsf{m}_\mathsf{i}}$ such that the sum of the values $\mathbb{v}_\mathsf{i,j}$ contained in the note is superior or equal to his bid/ask. That is $\sum_{\mathsf{j}=1}^{\mathsf{m}} \mathbb{v}_\mathsf{i,j} \geq \mathbb{p}_\mathsf{i} \times \mathbb{q}_\mathsf{i}$ if i is a buyer and $\perp_\mathsf{i} = \$$, and $\sum_{j=1}^{m} \mathbb{v}_{i,j} \geq \mathbb{q}_i$ if i is a seller and $\perp_\mathsf{i}, \odot$.

The ledger announces the closing of the registration phase at time $\mathsf{t}_2 = \mathsf{t}_1 + \Delta_1$

Auction Phase:

The ledger announces the start of the auction phase at time t_2 letting a delay Δ_2 specified by the protocol to the trusted third party $\mathcal{T}$ to compute the auction.

$\mathcal{T}$ considers the the notes $[[\mathbf{n}_\mathsf{i,j}]_{\mathsf{j}=1}^{\mathsf{m}_\mathsf{i}}]_{\mathsf{i}=1}^\mathsf{n}$ altogether with the bids/asks $[(\perp_\mathsf{i}, \mathbb{p}_\mathsf{i}, \mathbb{q}_\mathsf{i})]_{\mathsf{i}=1}^\mathsf{n}$ received between t_1 and t_2.

$\mathcal{T}$ checks that each participant i provides sufficient funds regarding to his bid/ask. If not, the participant i is considered as faulty and $\mathcal{T}$ sends him a message indicating that his bid/ask will not be taken into account for this iteration.

$\mathcal{T}$ computes the auction results $[\tilde{\mathbb{q}}_{\underline{\mathsf{i}}}]_{\underline{\mathsf{i}}=1}^\mathsf{n} = \mathcal{F}([(\perp_{\underline{\mathsf{i}}}, \mathbb{q}_{\underline{\mathsf{i}}}, \mathbb{p}_{\underline{\mathsf{i}}}]_{\underline{\mathsf{i}}=1}^\mathsf{n})$ where $\underline{\mathsf{i}}$ now runs amongst the not faulty participants.

$\mathcal{T}$ creates two new notes $[\mathbf{n}_{\$,\mathsf{i}}^\mathsf{new}]_{\mathsf{i}=1}^\mathsf{n}$ and $[\mathbf{n}_{\odot,\mathsf{i}}^\mathsf{new}]_{\mathsf{i}=1}^\mathsf{n}$ for each non faulty participant, containing coins and token energy corresponding to the auction results.

$\mathcal{T}$ creates a transaction $\mathsf{tx} = ([\mathsf{sn}_\mathsf{i}^\mathsf{old}]_{\mathsf{i}=1}^\mathsf{n}, [\mathsf{cm}_{\$,\mathsf{i}}^\mathsf{new}, \mathsf{cm}_{\odot,\mathsf{i}}^\mathsf{new}]_{\mathsf{i}=1}^\mathsf{n}, \pi)$ computing the corresponding proof π and sends it to the ledger.

$\mathcal{T}$ sends the two notes $\mathbf{n}_{\$,\mathsf{i}}^\mathsf{new}, \mathbf{n}_{\odot,\mathsf{i}}^\mathsf{new}$ to each corresponding participants i.

The ledger announces the closing of the auction phase at time $\mathsf{t}_3 = \mathsf{t}_2 + \Delta_2$

Receiving Phase:

At time t_3, each participant i has received the notes $\mathbf{n}_{\$,\mathsf{i}}^\mathsf{new}, \mathbf{n}_{\odot,\mathsf{i}}^\mathsf{new}$ and adds it to his wallet.

B. Protocol 2 (Almost Untrusted Third Party)

Setup:

In addition to the permanent lists CmList and SnList basically maintained on the ledger, two temporary lists CmList' and SnList' are created. Those lists will be used only during the time of the auction (each iteration of the auction process will create new lists). To those temporary lists will corresponds special kinds of transactions: the transactions "in" that consume coins of the ordinary list CmList to create new coins in the temporary list CmList' (in which case spending numbers are added to SnList), and the transactions "out" that consume coins of the temporary list CmList' to create new coins in the ordinary list CmList (in which case spending numbers are added to SnList').

Registration Phase:

The ledger opens auction at time t_1, allocating special memory emplacement to the auction iteration and letting a delay Δ_1 specified by the protocol to the participants to register.

Each participant i creates *two* temporary couples of *public-private addresses*$(\mathsf{addr}_\mathsf{pk,i}^\mathsf{in}, \mathsf{addr}_\mathsf{sk,i}^\mathsf{in})$ and $(\mathsf{addr}_\mathsf{pk,i}^\mathsf{out}, \mathsf{addr}_\mathsf{sk,i}^\mathsf{out})$. The first one will serve to anonymously provide funds to the auctioneer and the second to anonymously receive funds from the auctioneer.

Each participant provides his address $\mathsf{addr}_\mathsf{pk,i}^\mathsf{in}$ with funds from his permanent account(s) corresponding to his requested bid/ask-quantity $(\perp_i, \mathbb{p}_i, \mathbb{q}_i)$. To do so, he simply creates a new note $\mathbf{n}_i^\mathsf{in} = ((\perp_i, \mathbb{v}_i), \mathsf{addr}_\mathsf{pk,i}^\mathsf{in}, \rho_i^\mathsf{in}, \mathsf{r}_\mathsf{i}^\mathsf{in}, \mathsf{cm}_\mathsf{i}^\mathsf{in})$, with $\perp_\mathsf{i} = \odot$ and $\mathbb{v} = \mathbb{q}_i$ (i.e, the quantity of energy token proposed) if he is a seller, and $\perp_\mathsf{i} = \$$ and $\mathbb{v} = \mathbb{p}_i \times \mathbb{q}_i$ (i.e, the total price he is willing to pay) if he is a buyer, via an

$in-transaction$ $\mathsf{tx}_i^{\mathsf{in}} = ([\mathsf{sn}_{i,j}^{\mathsf{old}}]_{j=1}^{m}, \mathsf{cm}_i^{\mathsf{in}}, \pi_{\mathsf{tx},i}^{\mathsf{in}})$ consuming old notes $[\mathbf{n}_{i,j}^{\mathsf{old}}]_{j=1}^{m}$ from his permanent account(s).

Each participant encrypts the tuple $(\mathbf{n}_i^{\mathsf{in}}, \mathsf{addr}_{\mathsf{sk},i}^{\mathsf{in}}, (\bot_i, \mathbb{p}_i, \mathbb{q}_i), \mathsf{addr}_{\mathsf{pk},i}^{\mathsf{out}})$ constituted by the note and its corresponding private address, his bid/ask-quantity couple and his desired reception address to the third party $\mathcal{T}$ thanks to her public key $\mathsf{Key}_{\mathcal{T}}^{\mathsf{pub}}$, obtaining the cipher $\mathfrak{C}_i = \mathsf{Enc}_{\mathsf{r}_{\mathsf{Enc}}}(\mathsf{Key}_{\mathcal{T}}^{\mathsf{pub}}, (\mathbf{n}_i^{\mathsf{in}}, \mathsf{addr}_{\mathsf{sk},i}^{\mathsf{in}}, (\bot_i, \mathbb{p}_i, \mathbb{q}_i)), \mathsf{addr}_{\mathsf{pk},i}^{\mathsf{out}})$. He keeps in memory the randomness $\mathsf{r}_{\mathsf{Enc}}$ he used in the encryption algorithm until the end of the auction iteration.

The participant computes a NIZK $\pi_{\mathsf{Enc},i}$ that the commitment $\mathsf{cm}_i^{\mathsf{in}}$ corresponds to the note encrypted in $\mathfrak{C}_i$ for $\mathsf{Key}_{\mathcal{T}}^{\mathsf{pub}}$ and has enought coins or energy token to carry out the auction regarding his bid/ask. So $\pi_{\mathsf{Enc},i}$ is specified by the following instance, witness and statement:

Instance:	**Witness**:
$\mathsf{Key}_{\mathcal{T}}^{\mathsf{pub}}$	$(\mathbf{n}_i^{\mathsf{in}}, (\bot_i, \mathbb{p}_i, \mathbb{q}_i), \mathsf{addr}_{\mathsf{pk},i}^{\mathsf{out}})$
$\mathfrak{C}_i$	$\mathsf{addr}_{\mathsf{sk},i}^{\mathsf{in}}$
$\mathsf{cm}_i^{\mathsf{in}}$	$\mathsf{r}_{\mathsf{Enc}}$

Statement:

$$\mathfrak{C}_i == \mathsf{Enc}_{\mathsf{r}_{\mathsf{Enc}}}(\mathsf{Key}_{\mathcal{T}}^{\mathsf{pub}}, (\mathbf{n}_i^{\mathsf{in}}, \mathsf{addr}_{\mathsf{sk},i}^{\mathsf{in}}, (\bot_i, \mathbb{p}_i, \mathbb{q}_i), \mathsf{addr}_{\mathsf{pk},i}^{\mathsf{out}}))$$
$$\mathbf{n}_i^{\mathsf{in}} == ((\bot_i^{\mathsf{in}}, \mathbb{v}_i^{\mathsf{in}}), \mathsf{addr}_{\mathsf{pk},i}^{\mathsf{in}}, \rho_i^{\mathsf{in}}, \mathsf{r}_i^{\mathsf{in}}, \mathsf{cm}_i^{\mathsf{in}})$$
$$\mathsf{cm}_i^{\mathsf{in}} == \mathsf{H}(\mathsf{addr}_{\mathsf{pk},i}^{\mathsf{in}} || \mathbb{v}_i^{\mathsf{in}} || \rho_i^{\mathsf{in}} || \mathsf{r}_i^{\mathsf{in}})$$
$$\mathsf{addr}_{\mathsf{pk},i}^{\mathsf{in}} == \mathsf{H}(\mathsf{addr}_{\mathsf{sk},i}^{\mathsf{in}})$$
$$(\bot_i == \$) \implies \mathbb{v}_i = \mathbb{p}_i \times \mathbb{q}_i$$
$$(\bot_i == \odot) \implies \mathbb{v}_i = \mathbb{q}_i$$

Each participant sends altogether to the Ledger the triple $(\mathsf{tx}_i^{\mathsf{in}}, \mathfrak{C}_i, \pi_{\mathsf{Enc},i})$ constituted with the $in-transaction$, the cipher and the NIZK.

The ledger receives the triple, and if the $\mathsf{tx}_i^{\mathsf{in}}$ is valid[8] and the NIZK $\pi_{\mathsf{Enc},i}$ is valid, then he accepts $\mathsf{tx}_i^{\mathsf{in}}$, appending $[\mathsf{sn}_{i,j}^{\mathsf{out}}]_{j=1}^{m}$ to the permanent SnList and $\mathsf{cm}_i^{\mathsf{in}}$ to the temporary CmList'.

The ledger announces the closing of the registration phase at time $\mathsf{t}_2 = \mathsf{t}_1 + \Delta_1$, preventing any further appends to CmList' or Aux_1.

Auction Phase:

The ledger announces the start of the auction phase at time t_2 letting a delay Δ_2 specified by the protocol to the trusted third party $\mathcal{T}$ to compute the auction. During this time, no $out-transaction$ can be added to the ledger excepts if it comes from the trusted third party $\mathcal{T}$ and has the form we will soon specify.

$\mathcal{T}$ reads the ledger and decipher each message $\mathfrak{C}_i$ thanks to her private key $(\mathsf{Key}_{\mathcal{T}}^{\mathsf{priv}})$.

$\mathcal{T}$ computes the auction result applying the function $\mathcal{F}$ to the list $[(\bot_i, \mathbb{p}_i, \mathbb{q}_i)]_{i=1}^{n}$ obtaining the price $\mathcal{P}$ and the auction's result distribution $(\mathcal{P}, [\tilde{\mathbb{q}}_i]_{i=1}^{n}) = \mathcal{F}([(\bot_i, \mathbb{p}_i, \mathbb{q}_i)]_{i=1}^{n})$.

$\mathcal{T}$ creates new notes $[\mathbf{n}_{\$,i}^{\mathsf{out}}]_{i=1}^{n}$ and $[\mathbf{n}_{\odot,i}^{\mathsf{out}}]_{i=1}^{n}$ with respective coins and energy token values corresponding to the auction result, and with owner's addresses corresponding to the receiving addresses of the participants $[\mathsf{addr}_i^{\mathsf{out}}]_{i=1}^{n}$. Remark amongst the two notes received by each participant, one will serve to redeem to him the amount of coins (in case he his a buyer) or token energy (in case he his a seller) that the auction did not consumed. As example, if a participant bought/sold all his demanded/proposed quantity, this redeeming note will have a value 0 ; conversely if he bought/sold nothing this redeeming note will have value equal to the one of $\mathbf{n}_i^{\mathsf{in}}$ and the other note will value 0[9].

$\mathcal{T}$ creates a transaction $\mathsf{tx} = ([\mathsf{sn}_i^{\mathsf{in}}]_{i=1}^{n}, [\mathsf{cm}_{\$,i}^{\mathsf{out}}, \mathsf{cm}_{\odot,i}^{\mathsf{out}}]_{i=1}^{n}, \pi)$ computing the corresponding zero-knowledge proof π[10].

$\mathcal{T}$ encrypts each note $[\mathbf{n}_{\$,i}^{\mathsf{out}}]_{i=1}^{n}$ and $[\mathbf{n}_{\odot,i}^{\mathsf{out}}]_{i=1}^{n}$ for the temporary key $[\mathsf{Key}_i^{\mathsf{Pub}}]_{i=1}^{n}$ of the corresponding participant $\mathcal{P}_i$, obtaining a list of cipher $[[[\mathsf{n}_1], [[\mathsf{n}_2]]_{i=1}^{n}$.

The third party $\mathcal{T}$ then computes a zero-knowledge proof $\pi_{\mathcal{F}}$ proving that the distribution of coin and token energy contained in the transaction matches the auction's result, the private addresses of the created notes matches the addresses requested by the participants, and the notes of each participant are indeed encrypted with his temporary public-key. This zero-knowledge proof has the following instance, witness and statement:

Instance:	**Witness**:
$[\mathsf{sn}_i^{\mathsf{in}}]_{i=1}^{n}$	$[\mathbf{n}_i^{\mathsf{in}}]_{i=1}^{n}$
$[\mathsf{cm}_{\$,i}^{\mathsf{out}}, \mathsf{cm}_{\odot,i}^{\mathsf{out}}]_{i=1}^{n}$	$[\mathbf{n}_i^{\mathsf{out}}]_{i=1}^{n}$
$\mathsf{AuxList} = [\mathfrak{C}_i]_{i=1}^{n}$	$\mathsf{Key}_{\mathcal{T}}^{\mathsf{priv}}$
$\mathsf{AuxList}_2 = [\mathfrak{C}_{\mathcal{T}}^{i}]_{i=1}^{n}$	

Statement:

$$[(\mathbf{n}_i^{\mathsf{in}}, \mathsf{addr}_i^{\mathsf{in}}, (\bot_i, \mathbb{p}_i, \mathbb{q}_i), \mathsf{addr}_{\mathsf{pk},i}^{\mathsf{out}})]_{i=1}^{n} == [Dec(\mathsf{Key}_{\mathcal{T}}^{\mathsf{priv}}, \mathfrak{C}_i)]_{i=1}^{n}$$
$$[\mathbf{n}_i^{\mathsf{in}}.\mathsf{value}()]_{i=1}^{n} == [(\bot_i^{\mathsf{in}}, \mathbb{v}_i^{\mathsf{in}})]_{i=1}^{n}$$
$$(\mathcal{P}, [\tilde{\mathbb{q}}_i]_{i=1}^{n}) == \mathcal{F}([(\bot_i, \mathbb{p}_i, \mathbb{q}_i)]_{i=1}^{n})$$
$$for\ \bot\ \in \{\$, \odot\}$$
$$[\mathbf{n}_{\bot,i}^{\mathsf{out}}]_{i=1}^{n} == [((\bot_i^{\mathsf{out}}, \mathbb{v}_{\bot,i}^{\mathsf{out}}), \mathsf{addr}_{\mathsf{pk},i}^{\mathsf{out}}, \rho_{\bot,i}^{\mathsf{out}}, \mathsf{r}_{\bot,i}^{\mathsf{out}}, \mathsf{cm}_{\bot,i}^{\mathsf{out}})]_{i=1}^{n}$$
$$\mathsf{cm}_{\bot,i}^{\mathsf{out}} == \mathsf{H}(\mathsf{addr}_{\mathsf{pk},\bot,i}^{\mathsf{out}} || \mathbb{v}_{\bot,i}^{\mathsf{out}} || \rho_{\bot,i}^{\mathsf{out}} || \mathsf{r}_{\bot,i}^{\mathsf{out}})$$
$$([\bot_i^{\mathsf{out}}]_{i=1}^{n} == \$) \wedge (\bot_i == \mathcal{S}) \implies [\mathbb{v}_i^{\mathsf{out}}]_{i=1}^{n} == [\mathcal{P} \times \tilde{\mathsf{q}}_i]_{i=1}^{n}$$
$$([\bot_i^{\mathsf{out}}]_{i=1}^{n} == \$) \wedge (\bot_i == \mathcal{B}) \implies [\mathbb{v}_i^{\mathsf{out}}]_{i=1}^{n} == [\mathbb{v}_i^{\mathsf{in}} - \mathcal{P} \times \tilde{\mathsf{q}}_i]_{i=1}^{n}$$
$$([\bot_i^{\mathsf{out}}]_{i=1}^{n} == \odot) \wedge (\bot_i == \mathcal{S}) \implies [\mathbb{v}_i^{\mathsf{out}}]_{i=1}^{n} == [\mathbb{v}_i^{\mathsf{in}} - \tilde{\mathsf{q}}_i]_{i=1}^{n}$$
$$([\bot_i^{\mathsf{out}}]_{i=1}^{n} == \odot) \wedge (\bot_i == \mathcal{B}) \implies [\mathbb{v}_i^{\mathsf{out}}]_{i=1}^{n} == [\tilde{\mathsf{q}}_i]_{i=1}^{n}$$
$$[\mathfrak{C}_{\mathcal{T}}^{i}]_{i=1}^{n} == [Enc(\mathsf{Key}_i^{\mathsf{priv}}, (\mathbf{n}_{\$,i}^{\mathsf{out}}, \mathbf{n}_{\odot,i}^{\mathsf{out}}))]_{i=1}^{n}$$

The third party $\mathcal{T}$ sends to the ledger the triple $(\mathsf{tx}, \mathfrak{C}_{\mathcal{T}}^{i}, \pi_{\mathcal{F}})$ constituted by the transaction $\mathsf{tx} = ([\mathsf{sn}_i^{\mathsf{in}}]_{i=1}^{n}, [\mathsf{cm}_{\$,i}^{\mathsf{out}}, \mathsf{cm}_{\odot,i}^{\mathsf{out}}]_{i=1}^{n}, \pi)$, the list of encrypted output notes $\mathfrak{C}_{\mathcal{T}}^{i}$ and the proof $\pi_{\mathcal{F}}$.

The ledger checks that the proof $\pi_{\mathcal{F}}$ and the proof π of the transaction tx are valid, and that the spending numbers $[\mathsf{sn}]_{i=1}^{n}$ do not appear on the ledger. If those conditions are

[8] i.e that the proof π contained in $\mathsf{tx}_i^{\mathsf{in}} = ([\mathsf{sn}_i^{\mathsf{old}}]_{i=1}^{n}, \mathsf{cm}_i^{\mathsf{in}}, \pi_{\mathsf{tx},i}^{\mathsf{in}})$ is valid and none of the $[\mathsf{sn}_{i,j}^{\mathsf{old}}]_{j=1}^{m}$ appears on $SnList$.

[9] So, in case the participant is a buyer, the redeeming note is $\mathbf{n}_{\$,i}^{\mathsf{out}}$; in case he is a seller the redeeming note is $\mathbf{n}_{\odot,i}^{\mathsf{out}}$

[10] The transaction has just $2n$ commitments, $cm_{\$,i}$ and $cm_{\odot,i}$ do not necessary seem pairwise linked.

verified, the ledger adds the commitments $[\mathsf{cm}^{\mathsf{out}}_{\odot,\mathsf{i}}, \mathsf{cm}^{\mathsf{out}}_{\$,\mathsf{i}}]^{n}_{i=1}$ to the permanent list CmList and adds the encrypted notes to Aux.

The ledger announces the closing of the auction phase at time $\mathsf{t}_3 = \mathsf{t}_2 + \Delta_2$

Receiving Phase:

At time t_3 the ledger will allows withdraw of funds in case the $\mathcal{T}$ did not followed the protocol, allowing valid $out-transaction$ from CmList' to be performed. Thus there is two different situations that can happen:

- $\mathcal{T}$ correctly performed the exchange. In this case, each participant reads Aux, decipher his encrypted notes $\mathbf{n}^{\mathsf{out}}_{\$,\mathsf{i}}$ and $\mathbf{n}^{\mathsf{out}}_{\odot,\mathsf{i}}$ with his private key $\mathsf{Key}^{\mathsf{Priv}}_{\mathsf{i}}$ and adds it to his personal wallet.
- $\mathcal{T}$ failed to performed the exchange. In this case, each participant i withdraw his engaged funds consuming the note $\mathbf{n}^{\mathsf{in}}_{\mathsf{i}}$ and creating a note $\mathbf{n}^{\mathsf{out}}_{\mathsf{i}}$ for the address $\mathsf{addr}^{\mathsf{out}}_{\mathsf{i}}$ via an $out-transaction$ $\mathsf{tx}^{\mathsf{out}} = (\mathsf{sn}^{\mathsf{in}}_{\mathsf{i}}, \mathsf{cm}^{\mathsf{out}}_{\mathsf{i}}, \pi)$. To the transaction $\mathsf{tx}^{\mathsf{out}}$, i adds a NIZK π_{withdraw} that the output address matches the initial requested output address, using in particular the randomness used in encryption scheme in registration phase. π_{withdraw} has the following instance, witness and statement:

Instance:	**Witness**:
$\mathsf{tx} = (\mathsf{sn}^{\mathsf{in}}_{\mathsf{i}}, \mathsf{cm}^{\mathsf{out}}_{\mathsf{i}}, \pi)$	$\mathbf{n}^{\mathsf{in}}_{\mathsf{i}}$
$\mathsf{Key}^{\mathsf{pub}}_{\mathcal{T}}$	$\mathbf{n}^{\mathsf{out}}_{\mathsf{i}}$
$\mathfrak{C}_i \in \mathsf{AuxList}$	$\mathsf{r}_{\mathsf{Enc}}$

Statement:

$\mathfrak{C}_{\mathsf{i}} == \mathsf{Enc}_{\mathsf{r}_{\mathsf{Enc}}}(\mathsf{Key}^{\mathsf{pub}}_{\mathcal{T}}, (\mathbf{n}^{\mathsf{in}}_{\mathsf{i}}, \mathsf{addr}^{\mathsf{in}}_{\mathsf{sk,i}}, (\perp_{\mathsf{i}}, \mathbb{p}_{\mathsf{i}}, \mathbb{q}_{\mathsf{i}})), \mathsf{addr}^{\mathsf{out}}_{\mathsf{pk,i}})$
$\mathbf{n}^{\mathsf{out}}_{\mathsf{i}} == ((\perp_{\mathsf{i}}, \mathbb{v}_{\mathsf{i}}), \mathsf{addr}^{\mathsf{out}}_{\mathsf{pk,i}}, \rho^{\mathsf{out}}_{\mathsf{i}}, \mathsf{r}^{\mathsf{out}}_{\mathsf{i}}, \mathsf{cm}^{\mathsf{out}}_{\mathsf{i}})$
$\mathbf{n}^{\mathsf{in}}_{\mathsf{i}} == ((\perp_{\mathsf{i}}, \mathbb{v}_{\mathsf{i}}), \mathsf{addr}^{\mathsf{in}}_{\mathsf{pk,i}}, \rho^{\mathsf{in}}_{\mathsf{i}}, \mathsf{r}^{\mathsf{in}}_{\mathsf{i}}, \mathsf{cm}^{\mathsf{in}}_{\mathsf{i}})$
$\mathsf{addr}^{\mathsf{in}}_{\mathsf{pk,i}} == \mathsf{H}(\mathsf{addr}^{\mathsf{in}}_{\mathsf{sk,i}})$
$\mathsf{sn}^{\mathsf{in}}_{\mathsf{i}} == \mathsf{PRF}_{\mathsf{addr}^{\mathsf{in}}_{\mathsf{sk,i}}}(\rho^{\mathsf{in}}_{\mathsf{i}})$
$\mathsf{cm}^{\mathsf{out}}_{\mathsf{i}} == \mathsf{Com}((\perp^{\mathsf{out}}_{\mathsf{i}}, \mathbb{v}^{\mathsf{out}}_{\mathsf{i}})||\mathsf{addr}^{\mathsf{out}}_{\mathsf{pk,i}}||\rho^{\mathsf{out}}_{\mathsf{i}}, \mathsf{r}^{\mathsf{out}}_{\mathsf{i}})$

Notice it is necessary that the participant i proves he is using his own $\mathbf{n}^{\mathsf{in}}_{\mathsf{i}}$ to create $\mathbf{n}^{\mathsf{out}}_{\mathsf{i}}$, otherwise a malicious third party $\mathcal{T}$ could use the funds of one participant in the withdrawal of another one, preventing the first to retrieve his funds.

The ledger receiving $(\mathsf{tx}, \pi_{\mathsf{withdraw}})$ check the validity of tx and of the proof $\pi_{withdraw}$. If they are valid he accepts the transaction adding $\mathsf{cm}^{\mathsf{out}}_{\mathsf{i}}$ to CmList.

Optional Each participant adds to the ledger a transaction $\mathsf{tx} = (\mathsf{sn}^{\mathsf{out}}_{\$,\mathsf{i}}, \mathsf{sn}^{\mathsf{out}}_{\odot,\mathsf{i}}, \mathsf{cm}^{\mathsf{new}}_{1,\mathsf{i}}, \mathsf{cm}^{\mathsf{new}}_{2,\mathsf{i}}, \pi)$ transferring the funds from his temporary address $\mathsf{addr}^{\mathsf{out}}_{\mathsf{i}}$ to his permanent address.

VII. Comparison to other work

Why not use ZEXE?:

ZEXE is an extension to Zerocash that allows users not only to create an anonymous payment system, but also to perform any private, decentralized calculation. In ZEXE, both the inputs to the calculation and the type of calculation itself are private. We don't need this level of anonymity because, in our context, the ledger is entirely dedicated to a few specific tasks that are fully known to users, namely the exchange of energy tokens and coins and the performance of a double auction. This superfluous advantage of ZEXE in our case comes at the expense of efficiency. Indeed, ZEXE uses proof composition techniques (which, moreover, are not post-quantum for the most efficient) that are still costly in the current state of the art.

Other privacy-preserving auction protocol:

[8] proposes several variants of VCG auction mechanism implementations on the Ethereum blockchain, achieving different level of privacy. Each of those variants deals with several trade off between *Bidder's Anonimity*, *Bidder's Condifentiality*, *Winner's Anonimity* and *Winner's Confidentiality* against the other parties. In comparison, recall that our scheme provides completely all those privacy guarantees against other participants. The primary privacy concern in [8] arises from the use of Ethereum as the underlying blockchain for the smart contract. Ethereum, not inherently designed for privacy, fails to conceal the identity of the fund recipient (winner) at their address. Despite existing solutions to add a privacy layer [9] [10], this remains a notable weakness. On the other hand, their protocol is simpler and less cost full in computational resources, both for auctioneers and for bidders . It also requires less actions of the bidders. However, the cryptographic primitives used are not post-quantum.

[11] proposes an anonymous fair auction scheme similar to the one we propose with additional use of ring signature, except that there is no precise description of the interaction of the protocol with of the underlying anonymous transaction infrastructure which is not specified. In addition, their scheme is not usable as such for double-auction, because it is suppose that all the asset to be trade are in possession of the auctioneer, in contrast we deal with the case were auctioneer do not own directly the assets to be trade.

[12] propose anonymous double auction, but neither precises the interaction with the underlying anonymous cryptocurrency, making unclear if *Winner's Anonimity* is verified or not. It has the advantage, over our protocol 2 or our protocol 1 without MPC to be resistant against bidder-auctioneer collusion using multiple auctioneer and secret sharing spread the trust. [12] relies by design over discret logarithm problem, making it not post-quantum resistant.

VIII. Conclusion

We proposed two variants of a general anonymous sealed-bid exchange mechanism protocol based on Zerocash-like infrastructure. The first requires the auctioneer to be absolutely trusted, but this auctioneer can be replace by MPC between participants, avoiding any necessity to having resort to a third party, at cost of efficiency. The second scheme overcome this difficulty using altogether zero-knowledge proofs and an almost not trusted third party/auctioneer to obtain a protocol usable in practice. Finally, we implemented this second scheme for the particular use-case of the energy market with double auction proposed by Horta [1] in the context of smart-grid deployment, answering a crucial left-open question of privacy protection in peer to peer energy trading.

References

[1] J. Horta, "Innovative paradigms and architecture for future distribution electricity networks supporting the energy transition," Ph.D. dissertation, 04 2018.

[2] G. Oded, *Foundations of Cryptography: Volume 2, Basic Applications*, 1st ed. USA: Cambridge University Press, 2009.

[3] O. Goldreich, *Foundations of Cryptography*. Cambridge University Press, 2001, vol. 1.

[4] J. Groth, "Simulation-sound nizk proofs for a practical language and constant size group signatures," in *Advances in Cryptology – ASIACRYPT 2006*, X. Lai and K. Chen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 444–459.

[5] S. D. Warren and L. D. Brandeis, "The right to privacy," *Harvard Law Review*, vol. 4, no. 5, pp. 193–220, 1890. [Online]. Available: http://www.jstor.org/stable/1321160

[6] F. Reid and M. Harrigan, *An Analysis of Anonymity in the Bitcoin System*. New York, NY: Springer New York, 2013, pp. 197–223. [Online]. Available: https://doi.org/10.1007/978-1-4614-4139-$7_1$0

[7] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "*Zerocash: Decentralized Anonymous Payments from Bitcoin*," Cryptology ePrint Archive, Paper 2014/349, 2014. [Online]. Available: https://eprint.iacr.org/2014/349

[8] L. Massoni Sguerra, P. Jouvelot, F. Coelho, E. j. Gallego Arias, and G. Memmi, "The price of smart contract privacy," 2023.

[9] A. Rondelet and M. Zajac, "Zeth: On integrating zerocash on ethereum," 2019.

[10] B. Bünz, S. Agrawal, M. Zamani, and D. Boneh, "Zether: Towards privacy in a smart contract world," in *Financial Cryptography and Data Security*, J. Bonneau and N. Heninger, Eds. Cham: Springer International Publishing, 2020, pp. 423–443.

[11] G. Sharma, D. Verstraeten, V. Saraswat, J.-M. Dricot, and O. Markowitch, "Anonymous fair auction on blockchain," in *2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2021, pp. 1–5.

[12] X. Jia, L. Wang, K. Cheng, P. Jing, and X. Song, "A blockchain-based privacy-preserving and collusion-resistant scheme (ppcr) for double auctions," *Digital Communications and Networks*, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864823000834