What does the integrity of a Blind Auction scheme mean?

It means that…

1. Bidders should stick to their chosen value throughout the process.

2. The revealed bid is accessible only to the owner.

3. The owner does not collude with any bidder, i.e., sorts the secret bids honestly.

Traditionally…

The problem of integrity was to be reduced to a problem of trust.

And trust was deferred to…

a government…

an organization…

a digital platform…

In short, the problem was reduced to an issue of who to trust!

How not to rely on a central authority?

Generally, cryptography is the primary assistant of eliminating issues of trust in protocols that involve multiple untrusted parties.

Is it up to task in our use case?

Well…

1. Bidders should stick to their chosen value throughout the process.

Cryptographic Commitment!

2. The revealed bid is accessible only to the owner.

Encryption!

3. The owner does not collude with any bidder, i.e., sorts the secret bids honestly.

Hmmm…

There have been three approaches to VC…

- Audit-Based Schemes

Do it, then prove it!

- Secure Co-Processors

It is impossible to cheat me!

- Homomorphic Encryption

It is useless to cheat me!

Audit-Based Schemes?

It is possible to prove strict adherence to agreed-upon computations without leaking secret values!

We have explored two variants of Zero-Knowledge Proofs that seemed promising for our purposes…

Significant progress in recent years (partial list)

	size of proof π	verifier time	setup	post-quantum?
Groth'16	≈ 200 Bytes $O_\lambda(1)$	≈ 1.5 ms $O_\lambda(1)$	trusted per circuit	no
Plonk / Marlin	≈ 400 Bytes $O_\lambda(1)$	≈ 3 ms $O_\lambda(1)$	universal trusted setup	no
Bulletproofs	≈ 1.5 KB $O_\lambda(\log\|C\|)$	≈ 3 sec $O_\lambda(\|C\|)$	transparent	no
STARK	≈ 100 KB $O_\lambda(\log^2\|C\|)$	≈ 10 ms $O_\lambda(\log^2\|C\|)$	transparent	yes

How to implement a ZKP approach that will work in our setting?

To speak concretely, how to commit to encrypted values, then after revealing them, generate a proof that establishes a total order relationship among values that are two cryptographic layers away?

This would require several proofs!

That seemed overkill for an apparently simple problem.

So, we explored other options…

Homomorphic Encryption Schemes?

Fully Homomorphic Schemes allow sorting over encrypted values easily.

Again, we thought there had to be a more specialized approach…

So, we explored two homomorphic encryption schemes that leak only order information: OPE, and ORE.

Using OREs… (Our Actual Protocol)

(Bidding Phase)

Bidder will decide on a bid X, and compute a set of values…

$$Y = Enc_{ORE}(X) \qquad C = SHA_{256}(Y)$$

$$Z = RSA_{owner}(Y) \qquad D = SHA_{256}(Z)$$

Bidder will publish C and D as his bid commitment.

(Revealing Phase)

Bidder will publish Z.

(Sorting Phase)

Owner will decrypt all Zs, sort them, then publish the winning Y.

This scheme somewhat relaxes our initial requirement!

Interestingly, this protocol hints at a possible protocol that uses only one ZKP! While this would offer us our original promise, it comes at the cost of being less efficient, and complicated to implement…