Strictly-Black-Box Zero-Knowledge and Efficient Validation of Financial Transactions

Michael O. Rabin[1], Yishay Mansour[2⋆], S. Muthukrishnan[3], and Moti Yung[4]

[1] Harvard University, Hebrew University
[2] Tel-Aviv University.
[3] Google Inc. and Rutgers University.
[4] Google Inc. and Columbia University.

Abstract. Zero Knowledge Proofs (ZKPs) are one of the most striking innovations in theoretical computer science. In practice, the prevalent ZKP methods are, at times, too complicated to be useful for real-life applications. In this paper we present a practically efficient method for ZKPs which has a wide range applications. Specifically, motivated by the need to provide an upon-demand efficient validation of various financial transactions (e.g., the high-volume Internet auctions), we have developed a novel secure and highly efficient method for validating correctness of the output of a transaction while keeping input values secret. The method applies to input values which are publicly committed to by employing generic commitment functions (even input values submitted using tamper-proof hardware solely with input/ output access can be used.) We call these: strictly black box [SBB] commitments. Hence these commitments are typically much faster than public-key ones, and are the only cryptographic/ security tool we give the poly-time players, throughout. The general problem we solve in this work is: Let SLC be a publicly known staight line computation on n input values taken from a finite field and having k output values. The inputs are publicly committed to in a SBB manner. An Evaluator performs the SLC on the inputs and announces the output values. Upon demand the Evaluator, or a Prover acting on his behalf, can present to a Verifier a proof of correctness of the announced output values. This is done in a manner that (1) The input values as well as all intermediate values of the SLC remain information theoretically secret. (2) The probability that the Verifier will accept a false claim of correctness of the output values can be made exponentially small. (3) The Prover can supply any required number of proofs of correctness to multiple Verifiers. (4) The method is highly efficient. The application to financial processes is straight forward. To this end (1) we first use a novel technique for representation of values from a finite field which we call "split representation", the two coordinates of the split representation are generically committed to; (2) next, the SLC is augmented by the Prover into a "translation" which is presented to the Verifier as a

⋆ This research was supported in by The Israeli Centers of Research Excellence (I-CORE) program, (Center No. 4/11), by a grant from the Israel Science Foundation (ISF), by a grant from United States-Israel Binational Science Foundation (BSF), and by a grant from the Israeli Ministry of Science (MoS).

sequence of generically committed split representations of values; (3) using the translation, the Prover and Verifier conduct a secrecy preserving proof of correctness of the announced SLC output values; (4) in order to exponentially reduce the probability of cheating by the Prover and also to enable multiple proofs, a novel highly efficient method for preparation of any number of committed-to split representations of the n input values is employed. The extreme efficiency of these ZK methods is of decisive importance for large volume applications. Secrecy preserving validation of announced results of Vickrey auctions is our demonstrative example.

1 Introduction

Many current methods for validation of auctions employ "additive homomorphic encryptions." The Paillier encryption [8], in particular, employs a public integer $n = p \cdot q$ where p, q, are large primes constituting the private key. A value $x \in [0, n-1]$ is encrypted, using n, as $C = E(x, r)$ where r is a help value. Given two ciphers $C_i = E(x_i, r_i), i = 1, 2$, then $(C_1 \cdot C_2) \text{mod } n^2 = E((x_1 + x_2) \text{mod } n, (r_1 \cdot r_2) \text{mod } n)$. In [1-7] numerous applications of Paillier encryption to secure auctions are given. In [11] Paillier encryption is applied to combinatorial clock-proxy auctions. There are drawbacks to the use of homomorphic encryptions for verifying financial processes. Practicality, [10] employs Paillier encryption for providing secrecy preserving proofs of correctness of Vickery auctions. A proof of correctness of a 100 bidder auction required 800 minutes.

This work uses values $x \in F_p$ where p is a prime. A 128-bit prime is adequate for all applications. A value x is randomly represented by a vector $X = (u, v)$ such that $(u+v) mod p = x$, where $u \in F_p$ is randomly chosen. A value x appearing in a proof of correctness of outputs of a straight line computation is represented by a vector X and is committed to by $COM(X) = (COM(u), COM(v))$ where COM is any generic information- theoretic hiding and computationally binding commitment function. In proofs of correctness, the Prover never opens/reveals both coordinates of a commitment $COM(X)$ to a representation of a value x appearing the proof.

The method of using split representations of values and generic commitments appears in Kilian's [19] who credits it to [20]. However, Kilian only treats values in F_2, i.e. bits, and implements correctness proofs for boolean operations on bits. Also, he deals with ZKPs by a Prover without inputs from others, such as bidders in the case of auctions.

In [9] there appears the first use of random split representations of values $x \in F_p$ for a general prime p as well as information theoretic value hiding proofs for straight line computations involving $(x + y) mod p = z$, $(x \cdot y) mod p = z$, and the predicate $x \leq y$. The method of [9] allows a Prover to provide only a single ZK proof of correctness. Once this is done and commitments to components values are opened, another proof is impossible. A 100 bidder Vickery auction is ZK verified by this method in 4 minutes.

The present work greatly improves over [9] in several ways. The ZK proofs of correctness are considerably simpler and faster. The probability of a Verifier

accepting a false claim by the Prover is smaller and better analyzed. Most importantly, the Prover can supply to different Verifiers any required number of verifications. A 100 bidder Vickery auction is ZK verified by this new method in 2 ms. This new work enables repeated ZK verifications of large and large volume auctions and other financial processes.

The full version of the paper will include a full discussion of the SBB ZK proofs, as well as all the omitted proofs.

2 Representation, Commitment, and Translation

2.1 Validation Domain: The financial application domain settings

Consider as an example a Vickrey auction. An Auctioneer AU receives sealed bids $x_1, \cdots, x_n$ from bidders $P_1, \cdots, P_n$. At closing time the bid values are revealed to AU and he determines that, say, P_1 is the winner and he should pay x_2. Proving correctness of the announced result involves proving $x_1 > x_2$ and $x_2 \geq x_3, \cdots, x_2 \geq x_n$. After announcement of the result, the AU acting as a Prover, wants to prove correctness of the result to a Verifier, say one or all of the bidders, or to a judge, and this without revealing any of the bid values.

In general, the above example and many other real life situations are captured by a model where an Evaluator Prover EP who receives inputs from participants (i.e., they post commitments to the values and decommit privately to EP). EP, in turn, computes certain output values from these inputs (the computational procedure is efficient and public), and announces these "outputs." Later on, possibly upon demand, the EP provides a ZKP of correctness of the output values (given the public commitments) to a Verifier.

2.2 Inputs and Straight Line Computations

In our setting we assume that all inputs, constants, intermediate values and outputs of the EP's calculations are values smaller than $p/32$ where p is a known prime number, say $p \sim 2^{128}$. Our computations are in the finite field F_p so that for $x, y \in F_p$, $x + y$ and $x \cdot y$ are abbreviations for $(x + y) \bmod p$ and $x \cdot y \bmod p$. For financial applications the range of values $0 \leq x < 2^{128}$ is adequate.

Definition 1. *A straight line computation (*SLC*) on inputs $x_1, \ldots, x_n$ in F_p with k outputs $x_{L+1}, \ldots, x_{L+k}$ is a sequence*

$$\texttt{SLC} = x_1, \ldots, x_n, x_{n+1}, \ldots, x_{L+1}, \ldots, x_{L+k} \tag{1}$$

where for all $m > n$ there exist $i, j < m, L$ such that

- $x_m = x_i + x_j$, *or*
- $x_m = x_i \cdot x_j$, *or*
- $x_m = x_i$, *or*
- $x_m = TruthValue(x_i \leq x_j)$. *These x_m are restricted to output values.*

An example of the SLC for the output $x_1+\ldots+x_n$ is: $x_1, \ldots, x_n, x_{n+1}, \ldots, x_{2n-1}$, where

$$x_{n+1} = x_1 + x_2, x_{n+2} = x_{n+1} + x_3, \ldots \tag{2}$$

We now come to the main construct for enabling ZKP's for the correctness of the results $x_{L+1}, \ldots, x_{L+k}$ of the SLC in the above definition (with the strictly-black-box restriction as a SBB proof). It also presents a key component of our overall input representation.

Definition 2. *Let $x \in F_p$ be a value. A random representation $RR(x)$ of x is a vector $X = (u, v)$ where $u, v \in F_p, u$ was chosen randomly (notation $u \leftarrow F_p$) and $x = (u + v)$ mod p. For a vector $X = (u, v)$ we denote $\mathtt{val}(X) = (u + v) mod\, p$.*

The method for creating a $\mathrm{RR}(x) = (u, v)$ of x is to randomly chose $u \leftarrow F_p$ and set $v = (x - u) \mathrm{mod}\ p$. Note that from u (or v) by itself, no information about x can be deduced.

2.3 Generic Commitment Schemes

We can use any commitment scheme we wish (i.e., a generic commitment). We can use physically secure and physically binding scheme (based on physical assumptions) or any of the two kinds of commitments. Let us define one for concreteness: We express our work in terms of a generic commitment function COM, which is information theoretic hiding and computationally binding and is used as a "black box".

Definition 3. *An information theoretic hiding and computationally binding generic (black-box) commitment for values $u \in F_p$, is a function $COM : F_p \times [0, m - 1] \longrightarrow R$, where R is a set of m elements, such that:*

For any fixed $u \in F_p$, $\{COM(u, r) | r \in [0, m - 1]\} = R$. I.e. for a fixed u, the mapping $COM(u, r)$ is 1-1 from $[0, m - 1]$ onto R.

It is assumed that COM is computationally collision-free. I.e. finding two different pairs (u_1, r_1), (u_2, r_2) such that $COM(u_1, r_1) = COM(u_2, r_2)$ is not possible by a polynomial-time algorithm.

To commit to a value $u \in F_p$, a committer Alice randomly selects a random help value $r \in [0, m - 1]$, obtains from the Black Box the commitment value $c = COM(u, r)$ and submits c to the receiver Bob or posts it.

To de-commit c, Alice submits to Bob, or posts, the pair (u, r). Bob, or anyone else, has the result COM(r, u) computed by the Black Box on the decommitted values and verifies the equality of the commitment value c to the newly obtained value $COM(u, r)$.

Because $m = |R|$, and the above 1-1 property for fixed $u \in F_p$, this commitment is clearly information theoretic hiding.

Remark: Note that if the scheme is implemented via a physical envelope it is also information theoretically binding (the only way to get the value is to open the envelope). The literature often employs the highly structured Discrete-Log-based Pedersen commitment function [18].

Definition 4. *Let $X = (u, v)$ be a representation of $x = (u + v)$ mod p, then a commitment to X is defined as $\texttt{COM}(X) = (COM(u, r_1), COM(v, r_2))$, where , $r_1, , r_2$ are randomly chosen help values.*

2.4 The Main Theorem: SBB ZK Arguments for SLC

Theorem 1. *Let EP be computationally bounded prover. Having posted generic black-box information-theoretic hiding "split commitments"*

$$\texttt{COM}(X_1), \cdots, \texttt{COM}(X_n), \qquad (*)$$

of representations $X_1, \cdots X_n$ of values $x_1, \cdots, x_n$ in F_p , the EP can create a translation

$$TR = \ COM(X_1), \cdots, \texttt{COM}(X_n), \texttt{COM}(Y_{n+1}), \cdots, \texttt{COM}(Y_M), x_{L+1}, \cdots, x_{L+k}, (**)$$

of the public SLC (1) so that:

1. *Using the translation TR, the EP can conduct a two round interactive Zero Knowledge Argument for the statement that $x_{L+1}, \cdots, x_{L+k}$, are the correct output values of the publicly known SLC (1) [namely, completeness holds].*
2. *The proof is information theoretic hiding [i.e., there is a ZK simulator in the SBB model, and in fact if run over commitment of canonical input the visible transcript has the same distribution, i.e., the proof is Witness Indistinguishable].*
3. *The probability of EP cheating is at most 3/4 [i.e., this is the soundness error, and it is implies validity as in proof of correctness (probability of extracting is bounded by this value)].*
4. *Finally, the length TR is at most $11 \cdot L$.*

3 Proving Correctness of Additions and Equalities

We now show how the EP can prove to a Verifier correctness of an equation (3) for posted commitments (4). Let $X = (u_1, v_1), Y = (u_2, v_2)$, and $Z = (u_3, v_3)$, be random representations of the values x, y, and z. Note that

$$val(X) + val(Y) = val(Z) \qquad (3)$$

if and only if there exists a value w such that $X + Y = Z + (w, -w)$.
The EP has prepared commitments

$$\begin{aligned} COM(X) &= [COM(u_1, r_1), COM(v_1, s_1)], COM(Y) \\ &= [COM(u_2, r_2), COM(v_2, s_2], COM(Z) \\ &= [COM(u_3, r_3), COM(v_3, s_3)] \end{aligned} \qquad (4)$$

The EP posts the commitments (4) or sends them to the Verifier and claims that the hidden vectors X, Y, Z, satisfy the equation (3).

When challenged to prove this claim, the EP posts or sends to Verifier the above value w. The Verifier now presents to EP a randomly chosen challenge $c \leftarrow \{1, 2\}$.

Assume that c = 1. The EP de-commits /reveals to Verifier $u_j, r_j, j = 1, 2, 3$. The Verifier checks the commitments, i.e. computes $\texttt{COM}(u_j, r_j), j$ = 1,2,3 and compares to the posted first coordinates of $\texttt{COM}(X), \texttt{COM}(Y), COM(Z)$.

The Verifier next checks that $u_1 + u_2 = u_3 + w$. If $c = 2$ was chosen, then the Verifier checks that $v_1 + v_2 = v_3 - w$. The following two theorems are immediately obvious.

Theorem 2. *If the equation (3) is not true for the vectors committed in* $\texttt{COM}(X)$, $\texttt{COM}(Y)$, $COM(Z)$, *then Verifier will accept with probability at most* $1/2$ *the claim that (3) holds.*

Proof. Under our assumption about the COM function being computationally binding, the EP can open the commitments for $u_j, v_j, j = 1, 2, 3$, in only one way. Now, if (3) does not hold then at least one of the equations $u_1 + u_2 = u_3 + w$, or $v_1 + v_2 = v_3 - w$ is not true. So the probability that a random challenge $c \leftarrow \{1, 2\}$ will not uncover the falsity of the claim (3) is less than $1/2$.

Theorem 3. *The above interactive proof between EP and Verifier reveals nothing about the values* $\texttt{val}(X)$, $\texttt{val}(Y), \texttt{val}(Z)$ *beyond, if successful, that the claim that (3) is (actually may be) true.*

Proof. We note that the interactive proof involves only the revelation of either all the first coordinates of the "split representation" based commitments, or of all the second coordinates, of X, Y, Z. Assume that Verifier's challenge was $c = 1$. The only revealed values were random u_1, u_2, u_3, w which satisfy $u_1 + u_2 = u_3 + w$. Because the commitment function $C(\ ,\ )$ is information theoretically hiding, the un-opened second coordinates in the commitments (4) of $\texttt{COM}(X), \texttt{COM}(Y), \texttt{COM}(Z)$, are consistent with any three values $v_{1,1}, v_{2,2}, , v_{3,3}$, satisfying $v_{1,1} + v_{2,2} = v_{3,3} - w$. Thus the interactive proof is consistent with any three vectors X_1, Y_1, Z_1 satisfying the sum equality (3)– any consistent triple can serve as an alternative input as in Witness Indistinguishable (WI) proofs.

It is clear how to similarly create and conduct ZK Arguments for the correctness of a claim $\texttt{val}(X) = \texttt{val}(Y)$, for given commitments $\texttt{COM}(X), \texttt{COM}(Y)$.

4 Proving Correctness of Multiplications.

For proving correctness of the operations of multiplication $x_m = x_i \cdot x_j$ in the SLC, the EP will have presented to Verifier commitments $\texttt{COM}(X_m)$, $\texttt{COM}(X_i)$, $\texttt{COM}(X_j)$ for random representations of the values x_m, x_i, x_j. The EP has to prove to Verifier that

$$\texttt{val}(X_i) \cdot val(X_j) = val(X_m) \tag{5}$$

Let $X_i = (u_1, v_1)$, $X_j = (u_2, v_2)$, and $X_m = (u_3, v_3)$. The EP prepares auxiliary vectors $Z_0 = (u_1 \cdot u_2, v_1 \cdot v_2)$, $Z_1 = (u_1 \cdot v_2 + w_1, p - w_1)$, $Z_2 = (u_2 \cdot v_1 + w_2, p - w_2)$, where w_1, w_2 are randomly chosen values. The EP augments the commitments presented to Verifier into:

$$\texttt{COM}(X_m), \texttt{COM}(X_i), \texttt{COM}(X_j), \texttt{COM}(Z_0), \texttt{COM}(Z_1), \texttt{COM}(Z_2) \tag{6}$$

Clearly (5) holds if the following Aspects 0-4 hold true for the vectors committed in (6):

- Aspect 0: Z_0 is $(u_1 \cdot u_2, v_1 \cdot v_2)$.
- Aspect 1: $\texttt{val}(Z_1) = u_1 \cdot v_2$.
- Aspect 2: $\texttt{val}(Z_2) = u_2 \cdot v_1$.
- Aspect 4: $\texttt{val}(X_m) = val(Z_0) + val(Z_1) + val(Z_2)$.

In the interactive proof/verification either Aspects 0 and 4 are checked together, or Aspect 1, or Aspect 2 are separately checked. The Verifier randomly chooses with probability 1/2 to verify Aspect 0 and the addition in Aspect 4. He randomly chooses $c \leftarrow \{1, 2\}$. Say $c = 1$. The EP reveals the first coordinates of X_m, X_i, X_j and Z_0. Aspect 0 is verified. Aspect 4 is verified in the manner of verification of addition, see Section 3. If the EP ' s claim is false with respect to Aspect 0 or Aspect 4, then the probability of Verifier accepting is at most $3/4 = 1 - (1/2) \cdot (1/2)$.

The Verifier chooses to check either Aspect 1 or Aspect 2, each with probability 1/4. Say Aspect 1 was chosen by Verifier. The EP reveals the first coordinate u_1 of X_i and the second coordinate v_2 of X_j and both coordinates of Z_1 and checks the equality of Aspect 1. Note that if Aspect 1 is false and is chosen for verification then Verifier will never accept. Similarly for Aspect 2. Consequently, if (5) is false and the proof of correctness (5) presented by EP to Verifier is false in Aspect 1, or Aspect 2, then the probability that Verifier will accept is at most 3/4.

Altogether we have:

Theorem 4. *If the product claim is false then the probability that the Verifier will accept* EP *s proof of correctness is at most 3/4*

To achieve the information-theoretic ZK property of the above interactive proof of correctness we require an additional step in EP 's construction of the posted proof (6). We note that the same x_i can appear in the SLC (1) as left factor and as right factor. One example arises if the SLC has an operation $x_m = x_i \cdot x_i$. In this case verifying Aspect 1 will reveal both coordinates of X_i and hence reveal the value $x_i = val(X_i)$.

When preparing a proof of correctness of SLC the EP creates for every x_i involved in multiplications two random vector representations XL_i and XR_i.

The proof of correctness of the multiplication $x_m = x_i \cdot x_j$ will be:

$\texttt{COM}(X_m), \texttt{COM}(XL_i), \texttt{COM}(XR_j), \texttt{COM}(Z_0), \texttt{COM}(Z_1), \texttt{COM}(Z_2),$

where now $XL_i = (u_1, v_1)$, $XR_j = (u_2, v_2)$. It is clear that even if $i = j$, and Aspect 1 is checked, u_1 and v_2 are independent random values from F_p. Similarly if SLC contains another multiplication $x_k = x_s \cdot x_i$ it as well as $x_m = x_i \cdot x_j$ are verified wrt Aspect 1. For the first multiplication XR_i will be employed, for the second multiplication XL_i will be used. Thus again independent random first coordinate of XR_i and and second coordinate of XL_i are revealed. These considerations lead to a proof of:

Theorem 5. *If the* SLC *comprises only the operations* $+$ *and* $\cdot$ *(and no comparisons* $TruthValue(x_i \leq x_j)$*) then the* EP *can prepare a proof of correctness that is information-theoretically hiding and, by Theorem* 3*, if false will be accepted by Verifier with probability at most* 3/4*.*

Proof: Once the commitments (*) to the representations of the input values $x_1, \cdots, x_n$ are posted or sent to the Verifier, the EP prepares a translation TR for the SLC (1) as follows. Successively, after $X_1, \cdots, X_{m-1}$ were created, if $x_m = x_i + x_j$ then EP creates a $\text{RR}(x_m) = X_m$. Thus, by induction, $\texttt{val}(X_i) + val(X_j) = val(X_m)$. If $x_m = x_i \cdot x_j$ then EP creates the vectors Z_0, Z_1, Z_2, X_m as in the proof of correctness of multiplications, Section 4. Now $\texttt{val}(X_i) \cdot val(X_j) = val(X_m)$. In addition EP creates for every x_i appearing in the SLC (1) as a first and second factor in a multiplication, once X_i was created, another $\text{RR}(x_i)$.

The EP now has a translation TR for the SLC (1). He now creates commitments to all the vectors beyond the already posted commitments (*) and posts or sends those to the Verifier.

In the interactive proof of correctness, the Verifier chooses with probability 1/2 to simultaneously verify all additions, equalities, and Aspect 0 for all multiplications. Verifier randomly chooses $c \leftarrow \{1, 2\}$. Say $c = 1$. The EP reveals the first coordinates of all the X_i, and of all the Z_0, Z_1, Z_2 and all the w values required for proving correctness of additions. Using these first coordinates and w values the Verifier checks all equations. Similarly if $c = 2$. If the TR is false with respect to any addition, equality, or Aspect 0 of any multiplication, then the probability of Verifier accepting is at most 3/4. The Verifier chooses with probability 1/4 to simultaneously verify all Aspects 1 of all multiplications, and with probability 1/4 to simultaneously verify all Aspects 2 of all multiplications. If the TR is false in Aspect 1 or Aspect 2 for any multiplication then the probability of Verifier accepting the correctness of TR and hence the correctness of the results of the SLC (1) is at most 3/4. The above arguments lead to proving completeness, validity and statistical ZK/WI.

5 Proving Inequalities $x \leq y$, When $x, y < p/32$

Let $b^2 < p/32$ be an explicit bound on all values x_i, x_j in the SLC(1) for which $x_i \leq x_j$ needs to be proved. In the application to auctions, where the inputs $x_1, \cdots, x_n$ are bids, it is required that all bids are bounded by $p/32$.

We note that for integers $x, y < p/2$ we have $x \leq y$ iff $(y - x) \bmod p < p/2$. Thus if the EP proves to a verifier these three inequalities for split representations X, Y, Z of the values x, y, z, then he has proved that as integers $x \leq y$. Following [15], given $0 \leq z \leq b$ the EP can supply within the framework of `SLC` proofs of correctness, a proof that $(b) \leq z \leq 2b$ (i.e., as an integer $p - b \leq z < p$ or $0 \leq z \leq 2b$). Such a proof verification can be made part of Aspect 4 of the verification. All such inequalities can be simultaneously proved.

How do we get rid of the $p - b \leq z < p$ possibility?

Lagrange proved that every integer x is the sum of four squares of integers, $x = z_1^2 + z_2^2 + z_3^2 + z_4^2$. Rabin in 1977 MIT lectures and [17] gave an efficient polynomial-time algorithm for computing such a representation. For numbers $x \leq 2^{32}$, Schorn's Python implementation computed $60,000$ representations in 1 second.

[16] proposed using Lagrange in the context of proving range statements for encrypted numbers.

We apply Lagrange and [17] in our context of `SLC`s.

Given $0 \leq x \leq b^2 < p/32$, the `EP` computes $z_1, \ldots, z_4$ such that $x = z_1^2 + z_2^2 + z_3^2 + z_4^2$. Each z_i is between 0 and b. The numbers $x, z_1, \ldots, z_4$ are represented as usual in a translation TR by pairs $X, Z_1, \ldots, Z_4$.

`EP` incorporates in the `SLC` steps for enabling verification that $-b \leq \texttt{val}(Z_i) \leq 2b$ and that $\texttt{val}(X) = \texttt{val}(Z_1)^2 + \ldots + \texttt{val}(Z_4)^2$. This implies $0 \leq x \leq 16b^2$. Now $32b^2 < p$, i.e. $16b^2 < p/2$.

Theorem 6. *Given a* `SLC` *including inequalities, the* `EP` *can create a proof of correctness of the whole* `SLC` *and present it to* `V`*. The verification by* `V` *information-theoretically hides all input and intermediate values. If the proof is false then the probability that* `V` *will accept is at most* $3/4$.

Proof: For every x_i appearing in an inequality

$$x_m = TruthValue(x_i \leq x_j)$$

and every difference $x_j - x_i$ linked to such an inequality, the `EP` calculates the Lagrange sum of 4 squares representation. For each such sum of 4 squares $x = z_1^2 + z_2^2 + z_3^2 + z_4^2$ the `EP` creates random vector representation Z_j of $z_j, 1 \leq j \leq 4$ as well as random representations S_j of $(z_j)^2, 1 \leq j \leq 4$. The proof of correctness of the `SLC` now reduces to proof of correctness of a `SLC` involving only the operations $+$ and $\cdot$, so that Theorem 4 applies.

This establishes Main Theorem 1, but to use the result with negligible probabilities we need amplification, thus copying!

6 Exponential Reduction of Probability of Cheating and Multiple Proofs of Correctness.

The use of a single translation of a SLC in a proof of correctness allows a probability of 3/4 for the EP to cheat the Verifier. This soundness probability is unacceptable in real-life applications. The way to reduce the margin of uncertainty is for the EP to add redundancy and present to the Verifier m translations $TR_1, \ldots, TR_m$ of the SLC. The Verifier randomly and independently challenges the EP for each T_j to verify Aspect 4 with probability 1/2 or one of Aspects 1, 2, each with probability 1/4. Recall that the EP 's construction of the whole translation for the SLC starts with commitments $\mathtt{COM}(X_1), \ldots, \mathtt{COM}(X_n)$, where X_j is a random representation of the input value $x_j, 1 \leq j \leq n$. As explained in the Overview, $\mathtt{COM}(X_1), \ldots, \mathtt{COM}(X_n)$, were submitted by $P_1, \ldots, P_n$. To reduce the probability of cheating and to allow multiple proofs of correctness, each participant P_j has to submit multiple random representations (i.e., "the redundant split commitment") of his input value x_j. The whole protocol proceeds as follows. $P_1, \ldots, P_n$ submit input values $x_1, \ldots, x_n$ to EP : $P_i, 1 \leq i \leq n$, prepares $3k$ random representations $Y_1^{(i)}, \ldots, Y_{3k}^{(i)}$ of his value x_i. P_i submits commitments $\mathtt{COM}(Y_1^{(i)}), \ldots, \mathtt{COM}(Y_{3k}^{(i)})$ to the EP . EP posts all commitments from all $P_i, 1 \leq i \leq n$, denoted $\mathcal{Y}$:

$$\begin{array}{r} \mathtt{COM}(Y_1^{(1)}), \mathtt{COM}(Y_2^{(1)}), \ldots, \mathtt{COM}(Y_{3k}^{(1)}) \\ \mathtt{COM}(Y_1^{(2)}), \mathtt{COM}(Y_2^{(2)}), \ldots, \mathtt{COM}(Y_{3k}^{(2)}) \\ \ldots \\ \mathtt{COM}(Y_1^{(n)}), \mathtt{COM}(Y_2^{(n)}), \ldots, \mathtt{COM}(Y_{3k}^{(n)}) \end{array}$$

EP creates additional random representations of input values: Every P_i opens (reveals) $Y_1^{(i)}, \ldots, Y_{3k}^{(i)}$ to EP . The EP chooses L (say $L = 20$) and constructs and posts additional $10kL = m$ columns, denoted $\mathcal{X}$:

$$\begin{array}{r} \mathtt{COM}(X_1^{(1)}), \mathtt{COM}(X_2^{(1)}), \ldots, \mathtt{COM}(X_m^{(1)}) \\ \mathtt{COM}(X_1^{(2)}), \mathtt{COM}(X_2^{(2)}), \ldots, \mathtt{COM}(X_m^{(2)}) \\ \ldots \\ \mathtt{COM}(X_1^{(n)}), \mathtt{COM}(X_2^{(2)}), \ldots, \mathtt{COM}(X_m^{(n)}) \end{array}$$

Definition 5. *We call two sequences $X^{(1)}, \ldots, X^{(n)}$ and $Y^{(1)}, \ldots, Y^{(n)}$* value consistent *if $\mathtt{val}(X^{(j)}) = \mathtt{val}(Y^{(j)}), 1 \leq j \leq n$.*

We have by the following a way to replicate values consistently:

Theorem 7. *Given commitments $\mathtt{COM}(X^{(j)}), \mathtt{COM}(Y^{(j)}), 1 \leq j \leq n$, to two sequences of representations for which the* EP *claims value consistency. The* EP *can give a* ZKP *for this claim such that if the claim is false then the probability that* V *will accept is at most* 1/2.

Using this Theorem, we derive the following which will imply SBB proof of correctness with validity probability negligible and statistical security, resulting in our second major Theorem:

Theorem 8. *Interactively with* V, EP *can provide an SBB ZK (WI) proof of knowledge with probability of cheating at most* $(1/2 + 1/e^2)^k + (1/2 + 1/e^2)^{3k}$ *that*

1. *In the* $n \times 3k$ *posted matrix* $\mathcal{Y}$ *of representation of input values, at least* $2k$ *columns are pair-wise value consistent. By definition, the common* $2k$ *majority of values in row* i *is* $P_i's$ *input* x_i.
2. *In the* $n \times m$ *matrix* $\mathcal{X}$ *at least* $(1 - 1/L)m$ *columns are pair-wise value consistent with the majority values of the input matrix.*
3. *The interactive proof involves all input representations of matrix* $\mathcal{Y}$ *and* $6kL$ *columns of the matrix* $\mathcal{X}$. *The remaining untouched* $4kL$ *columns of the matrix* $\mathcal{X}$ *may be used by* EP *to construct* $4L$ *proofs of correctness of announced* SLC *results.*

Proof: In the interactive proof/verification, the Verifier randomly chooses for each of the $3k$ columns C_i of the inputs matrix $\mathcal{Y}$, $2L$ columns of the matrix of the matrix $\mathcal{X}$ constructed by the EP. The EP interactively proves that C_i is value-consistent with each of the $2L$ correspondingly chosen columns. The Verifier accepts only if all those verifications are successful. The proof for claimed probability of soundness will be given in the full version of the paper.

6.1 Putting it all together

As a consequence of Theorem 8 the EP and V now have available $4kL$ unused columns of the matrix 7 and with high probability at least $(1-1/L)4kL$ of these columns are value consistent with the input values $x_1, \ldots, x_n$.

For the interactive ZKP of the correctness of the outputs of the SLC, V randomly chooses k of these columns and presents this choice to the EP . The EP extends each of the k columns (representing commitments to the n inputs to the SLC) to a full proof of correctness according to Theorem 5. The probability of V accepting such a proof for a single translation is $1/L + 3/4$. The $1/L$ terms bounds the probability that the chosen column is not value consistent with the inputs. The probability that the outputs are incorrect and yet V will accept is at most $(1/2 + 1/e^2)^k + (1/2 + 1/e^2)^{3k} + (1/L + 3/4)^k$. This replication of commitment is also the crux of the ability to repeat the proof process.

References

1. M. Abe, K. Suzuki, M+1-st price auction using homomorphic encryption, in: Procillier's. Public Key Cryptography, 2002.
2. F. Brandt, How to obtain full privacy in auctions, Tech. rep., Carnegie Mellon University, 2005. online

3. M. Burmaster, E. Magkos, V. Chrissikopoulos, Uncoercible e-bidding games, Electrocronic Commerce Research 4 (1-2) (2004) 113-125.
4. X. Chen, K. Kim, B. Lee, Receipt-free electronic auction schemes using homomorphic encryption, in: ICISC, 2003.
5. I. Damgard, M. Jurik, A generalisation, a simplification and some applications of P probabilistic public-key system, in: Proc. Public Key Cryptography 01, 2001.
6. M. J. Jurik, Extensions to the paillier cryptosystem with applications to cryptological protocols, Ph.D. thesis, University of Arhus, 2003.
7. H. Lipmaa, N. Asokan, V. Niemi, Secure Vickrey auctions without threshold trust, in: P. 6th International Conference on Financial Cryptography (FC 2002), 2002 87-101.
8. P. Paillier, Public-key cryptosystems based on composite residuosity classes, in: Proc. EUROCRYPT '99, 1999 223-239.
9. M. O. Rabin, R. A. Servedio, C. Thorpe, Highly efficient secrecy-preserving proofs of correctness of computations and applications, in: Proc. IEEE Symposium on Logic in Computer Science, 2007.
10. D. C. Parkes, M. O. Rabin, S. M. Shieber, and C. A. Thorpe. Practical secrecy-preserving, verifiably correct and trustworthy auctions. In Proceedings of the 8th International Conference on Electronic Commerce (ICEC), pages 70-81, 2006.
11. Christopher Thorpe and David C. Parkes. Cryptographic Combinatorial Securities Exchanges. In the Financial Cryptography and Data Security (FC'09), 285-304, 2009.
12. Peter Bogetoft, Dan Lund Christensen, Ivan Damgrd, Martin Geisler, Thomas P. Jakobsen, Mikkel Krigaard, Janus Dam Nielsen, Jesper Buus Nielsen, Kurt Nielsen, Jakob Pagter, Michael I. Schwartzbach, Tomas Toft. Secure Multiparty Computation Goes Live. Financial Cryptography 2009: 325-343.
13. S. Goldwasser, S. Micali, C. Rackoff. The Knowledge Complexity of Interactive Proof Systems, SIAM Journal on Computing vol. 18, No 1, pp. 186- 208, 1989.
14. O. Goldreich, S. Micali, A. Wigderson. Proofs that Yield Nothing but Their Validity, or all Languages in NP have ZKP systems, Journal of the ACM vol. 38, No. 3, pp. 691-729, 1991.
15. E. Brickell, D. Chaum, I. Damgard, and J.V. de G. Gradual and Verifiable Release of a Secret, Proc. CRYPTO 87, vol. LNCS 293, pp. 156-166, 1988.
16. J. Camenish and V. Shoup. Practical Verifiable Encryption and Decryption of Discrete Logarithms, Proc. Crypto 2003.
17. M.O. Rabin and J.O. Shallit. Randomized Algorithms in Number Theory, Comm. In Pure and Applied Mathematics, vol. 39, pp.239-256, 1986.
18. T.P. Pedersen. Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing, Proc. CRYPTO 91, pp. 129-140, Springer, 1991.
19. J. Kilian. A note on efficient zero-knowledge proofs and arguments, In Proceedings of STOC'92, pages 723-732, 1992.
20. G. Brassard, D. Chaum, and C. Crepeau. Minimum disclosure proofs of knowledge. Journal of Computer and System Sciences, 37: pages 156- 189, 1988.