Client and Server Verifiable Additive Homomorphic Secret Sharing

Enforcing Clients to Act Honestly in Sever Verifiable Additive Homomorphic Secret Sharing by including a Range proofs

Master's thesis in Computer science and engineering

Hanna Ek

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2021

Master's thesis 2021

Client and Server Verifiable Additive Homomorphic Secret Sharing

Enforcing Clients to Act Honestly in Sever Verifiable Additive Homomorphic Secret Sharing by including a Range proofs

Hanna Ek

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2021

A Chalmers University of Technology Master's thesis template for LaTeX
Enforcing Clients to Act Honestly in Sever Verifiable Additive Homomorphic Secret Sharing by including a Range proofs
Hanna Ek

Supervisor: Georgia Tsaloli and Katerina Mitrokotsa, Department of Computer Science and Engineering
Examiner: Katerina Mitrokotsa, Department of Computer Science and Engineering

Master's Thesis 2021
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Description of the picture on the cover page (if applicable)

Typeset in LaTeX
Gothenburg, Sweden 2021

A Chalmers University of Technology Master's thesis template for LaTeX
Enforcing Clients to Act Honestly in Sever Verifiable Additive Homomorphic Secret Sharing by including a Range proofs

Hanna Ek
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Abstract text about your project in Computer Science and Engineering.

Keywords: Computer, science, computer science, engineering, project, thesis.

Acknowledgements

Here, you can say thank you to your supervisor(s), company advisors and other people that supported you during your project.

Name Familyname, Gothenburg, March 2021

Contents

List of Figures

List of Tables

1
Introduction

1.1 Contribution

1.2 Organisation

2

Theory

This chapter will present the theory that is relevant to construct a client and server verifiable VAHSS construction. First the *preliminaries* needed is described, this includes notation that is used, theorems/definitions, assumptions and several cryptographic preliminaries and concepts. Then the VAHSS construction, which this reports aims to extend to include verification of clients honesty, originally described in [12, **?**] is presented. Finally two different range proof constructions is described in detail. These two constructions will be refereed to as *signature-based range proof* and *bulletproof*.

2.1 Cryptographic preliminaries

This sections aim to present all relevant background to the cryptographic constructions presented in section 2.2 and 2.3.

Notation and setup

First lets define some notation that will be used trough out the paper. Consider n clients and m servers, to simplify notation define the two sets $\mathcal{N} = \{1, ..., n\}$ and $\mathcal{M} = \{1, ..., m\}$. Let c_i and x_i for $i \in \mathcal{N}$ denote the clients (data providers) and their respective data. Denote the servers by s_j, where $j \in \mathcal{M}$.

Let $\mathbb{F} = \mathbb{Z}_N$ denote a finite field, where N is a large prime and let $\mathbb{G}$ denote he unique subgroup of order q. Define $g \in \mathbb{G}$ to be a group generator and $h \in \mathbb{G}$ a group element such that no one knows $log_g\, h$. The two group elements g, h can either be chosen by a trusted party or by one of the participates using a *coin-flipping* protocol [9].

The notation $x \in_R \mathbb{Y}$, means that an element x in chosen at random from the set $\mathbb{Y}$.

Definitions and theorems

Definition 1 (**Euler's totient function**)**.** *The function $\Phi(n)$ is defined as the counter of the number of integers in the set $\{1, ..., n\}$ that are relative primes to n. Therefore if n is a prime number $\phi(n) = n - 1$.*

Theorem 1 (Euler's Theorem)**.** *For all integers x and n that are co-prime it holds that: $x^{\Phi(n)} = 1$ $(mod\ n)$, where $\Phi(n)$ is Euler's totient function.*

From Theorem 1 it follows that for arbitrary y it holds that $x^{y\Phi(n)} = 1 \,(\text{mod n})$.

Definition 2 (Pseudorandom Function (PRF))**.** *ehjs*

Assumptions

In this section cryptographic assumptions that is used in the constructions presented later rely on will be presented. A remark is that this assumptions will not hold in the presence of quantum computers which means that the constructions presented here is not secure post quantum.

Assumption 1 (**Discrete logarithmic assumption**)**.** *Let $\mathbb{G}$ be a group of prime order q, a generator $g \in \mathbb{G}$ and an arbitrary element $y \in \mathbb{G}$, it is infeasible to find $x \in \mathbb{Z}_q$, such that $y = g^x$*

Assumption 2 (**q-strong Diffie Hellman Assumption**)**.** *Given a group $\mathbb{G}$, a random generator $g \in \mathbb{G}$ and powers $g^x, ..., g^{x_q}$, for $x \in_R \mathbb{Z}_p$ and $q = |\mathbb{G}|$. It is then infeasible for an adversary to find $(c, g^{\frac{1}{x+c}})$, where $c \in \mathbb{Z}_p$.*

Homomorphic Secret Sharing

The idea behind secret sharing [10] is to split a secret x into m shares x_i s.t $i \in \{1, ..., m\}$, where individual shares reviles no information about the original secret x. To reconstruct the value x a party would need to combine at least τ shares for some threshold value τ and any subset of shares smaller than τ reviles no information about the original secret x, this is called a (τ, m)-threshold scheme. In this paper $\tau = \textit{number of shares} = m$. Furter this paper will consider additive secret sharing scheme, thus the shares will have the property; $x = \sum_{i=1}^{\tau} x_i$.

Homomorphic hash functions

Let $\mathcal{H}$ be a cryptographic hash function [13], $\mathcal{H} : \mathbb{F} \rightarrow \mathbb{G}$. Any such function should satisfy the following two properties:

- **Collision-resistant** It should be hard to find $x, x' \in \mathbb{F}$ such that $x \neq x'$ and $\mathcal{H}(x) = \mathcal{H}(x')$.
- **One-Way** It should be computationally hard to find $\mathcal{H}^{-1}(x)$.

A homomorphic hash function should also satisfy the following property:

- **Homomorphism** For any $x, x' \in \mathbb{F}$ it should hold that $\mathcal{H}(x \circ x') = \mathcal{H}(x) \circ \mathcal{H}(x')$. Where $\circ$ is either " + " or " $*$ ".

A funtion satisfying the thee properties is $\mathcal{H}_1(x) : \mathbb{F} \rightarrow \mathbb{G}$ and $\mathcal{H}_1(x) = g^x$ [13].

Pedersen Commitment scheme

Define a commitment to an $x \in \mathbb{F}$ as $\mathbb{E}(x, R) = g^x h^R$, where $R \in_R \mathbb{F}$, this commitment is known as *Pedersen commitment* and originally presented in [9]. This commitment satisfies the following theorem;

Theorem 2. *For any $x \in \mathbb{F}$ and for $R \in_R \mathbb{F}$, it follows that $\mathbb{E}(x, R)$ is uniformly distributed in $\mathbb{G}$. If we have two commits satisfying $\mathbb{E}(x, R) = \mathbb{E}(x', R')$ $x \neq x'$ and $x \neq x'$ then it must hold that $R \neq R' \, mod \, q$ and*

$$log_g(h) = \frac{x - x'}{R' - R} \; mod \; N.$$

Proof. The statements of the theorem follows from solving for $log_g(h)$ in $\mathbb{E}(x, R) = \mathbb{E}(x', R')$ □

Theorem 2 implies that if someone knows the discrete logarithm of h with respect to g, i.e $log_g(h)$, he is able to provide two equal commits, $\mathbb{E}(x, R) = \mathbb{E}(x', R')$ such that $x \neq x'$. Note that in he set up of the protocol it was required that this logarithm should be unknown to any party and generated by a trusted third party or by one of the participants using a coin-flipping protocol.

Further note that Pedersen commitment is homomorphic. Hence for arbitrary messages $x_1, x_2 \in \mathbb{F}$, random values $R_1, R_2 \in_R \mathbb{F}$ and the commits $C_i = \mathbb{E}(x_i, R_i), i \in \{1, 2\}$, it holds that $C_1 \cdot C_2 = \mathbb{E}(x_1 + x_2, R_1 + R_2)$.

A final remark is the similarity between the hash function $\mathcal{H}_1$ and the Pedersen commitment $\mathbb{E}$, the hash function can be seen as a generalisation of the Pedersen commitment.

Vector Pedersen Commitment scheme

Bilinear mapping

Zero knowledge proof

[7]

A zero knowledge proof can be made non-interactive by Fiat-Shamir heuristic [1]. A non interactive constructions means that there is no communication between the prover and verifier while constructing the proof. Usually in interactive protocols the verifier sends a challenge $c \in_R \mathbb{F}$ to the prover which is included in the proof to convince the verifier of its correctness. The Fiat-Shamir heuristic uses a challenge that instead of being randomly chosen by the verifier is the a hash of the transcripts up to this point. This heuristic allows to convert an interactive protocol to a non-interactive while preserving its secure and full zero-knowledge in the random oracle model.

2.2 Verifiable additive homomorphic secret sharing

This section will describe the verifiable additive homomorphic secret sharing (VAHSS) constructions presented in [?, 12]. An additive homophobic secret sharing construction assumes n clients/data providers and m servers. The clients split the secret x_i into m shares, x_{ij} and sends one share to each server. The servers receives shares from all n clients and computes the partial function $y_j = \sum_{i=1}^{n} x_{ij}$ and publishes the result. The final result $y = \sum_{j=1}^{m} y_j$ can then be computed by any party. In verifiable additive homomorphic secret sharing a proof σ that verifies that $y = \sum_{j=1}^{n} y_j = \sum_{j=1}^{m} \left(\sum_{i=1}^{n} x_{ij} \right) = \sum_{i=1}^{n} \left(\sum_{j=1}^{m} x_{ij} \right) = \sum_{i=1}^{n} x_i$ is generated and published. This allows any party to verify the correctness of the severs computations. Remark that the individual secrets x_i is never revealed in the protocol.

Construction

In this section a VAHSS construction is presented. The construction consists of the six PPT algorithms: **ShareSecret**, **PartialEval**, **PartialProof**, **FinalEval**, **FinalProof** and **Verify**. The clients/data providers executed the step **ShareSecret**, the servers **PartialEval** and **PartialProof** and the last three steps can run by anyone. The complete construction looks like:

Construction 1 : Verifiable additive homomorphic secret sharing

- **ShareSecret** $(1^\lambda, i, x_i) \rightarrow (\tau_i, \{x_{ij}\}_{j \in \mathcal{M}})$
  Pick uniformly at random $\{a_i\}_{i \in \{1,..,t\}} \in \mathbb{F}$ and a t-degree polunomial p_i on the form $p_i(X) = x_i + a_1 X + ... + a_t X^t$. Let $H : x \rightarrow g^x$, (g generator the multiplicative group of $\mathbb{F}$), be a collision-resistant homomorphic hash function. Let $R_i \in \mathbb{F}$ be the output of a PRF. We require $R_n \in \mathbb{F}$ to satisfy $R_n = \phi(N) \lceil \frac{\sum_{i=1}^{n-1} R_i}{\phi(N)} \rceil - \sum_{i=1}^{n-1} R_i$. Compute $\tau_i = H(x_i + R_i)$, and put $x_{ij} = \lambda_{i,j} p_i(\theta_{ij})$. The algorithm published τ_i and sends $x_{i,j}$ to server j for $j \in \mathcal{M}$.
- **PartialEval** $(j, \{x_{ij}\}_{i \in \mathcal{N}}) \rightarrow y_j$
  Compute and publish $y_j = \sum_{i=1}^{n} x_{ij}$.
- **PartialProof** $(j, \{x_{ij}\}_{i \in \mathcal{N}}) \rightarrow \sigma_j$
  Compute and publish $\sigma_j = \prod_{i=1}^{n} g^{x_{ij}} = g^{\sum_{i=1}^{n} x_{ij}} = g^{y_j} = H(y_j)$.
- **FinalEval** $(\{y_j\}_{j \in \mathcal{M}}) \rightarrow y$
  Compute and output $y = \sum_{i=1}^{n} y_j$.
- **FinalProof** $(\{\sigma_j\}_{j \in \mathcal{M}}) \rightarrow \sigma$
  Compute and output $\sigma = \prod_{j=1}^{n} \sigma_j = \prod_{j=1}^{m} g^{y_j} = g^{\sum_{j=1}^{m} y_j} = g^y = H(y)$.
- **Verify** $(\{\tau_i\}_{i \in \mathcal{N}}, \sigma, y) \rightarrow \{0, 1\}$
  Compute and output $\sigma = \prod_{i=1}^{n} \tau_i \wedge \prod_{i=1}^{n} \tau_i = H(y)$.

The above described construction satisfies the correctness, security and verifiability requirements presented below, this is stated in Theorem 3

Theorem 3. *The VAHSS construction above satisfies the correctness, security and verifiability requirements described below.*

Proof. See section 4.1 in [?]. □

Correctness, Security and Verifiability

A HSS/additive-HSS construction should satisfy the two requirements: *Correctness* and *Security*. A verifiable additive HSS should also satisfy *Verifiability*. The requirements are defined as:

- **Correctness** It must hold that $\Pr\Big[\mathbf{Verify}(pp, \sigma, y) = 1\Big] = 1$. This means that with probability 1 the output y from the construction is accepted given all parties where honest and the protocol were executed correctly.
- **Security** Let T define the set of corrupted servers with $|T| < m$, i.e at least one honest server, and $\mathrm{Adv}(1^\lambda, \mathcal{A}, T) := \Pr[b' = b] - 1/2$, i.e the advantage of $\mathcal{A} = \{\mathcal{A}_1, \mathcal{D}\}$ in guessing b in the following experiment:
  1. The adversary $\mathcal{A}_1$ gives $(i, x_i, x_i') \leftarrow \mathcal{A}_1$ to the challenger, where $i \in [n], x_i \neq x_i'$ and $|x_i| = |x_i'|$.
  2. The challenger picks a bit $b \in 0, 1$ uniformly at random chooses and computes $(\hat{\text{share}}_{i1}, ..., \hat{\text{share}}_{im}, \tau_i) \leftarrow \mathbf{ShareSecret}(1^\lambda, i, \hat{\mathbf{x}}_i)$, where $\hat{\mathbf{x}}_i$ is $\hat{\mathbf{x}}_i = \begin{cases} x_i, & \text{if } b = 0 \\ x_i' & \text{else} \end{cases}$.
  3. Given the shares from the corrupted servers T and $\hat{\tau}_i$ the adversary distinguisger outputs a guess $b' \leftarrow \mathcal{D}(\hat{\text{share}}_{j|s_j \in T}, \hat{\tau}_i)$.

  A construction is t-secure if for all $T \subset \{s_1, ..., s_m\}$ with $|T| < t$ if $\mathrm{Adv}(1^\lambda, \mathcal{A}, T) < \varepsilon(\lambda)$ for some negligible $\varepsilon(\lambda)$.
- **Verifiability** Let $\mathcal{A}$ denote any PPT and T denote the set of corrupted servers with $T \leq m$. Note that if $|T| = m$, the verifiability property holds but not the security property. The verifiability property requires that any $\mathcal{A}$ who can modify the input shares to all servers $s_j \in T$ can cause a wrong value to be excepted as $y = f(x_1, ..., x_n)$ with negligible probability.

2.3 Constructions for verifying clients input

A range proof is constructed to prove the following statement about a secret σ in a Pedersen commitment without revealing anything else regarding σ:

$$\{(g, h \in \mathbb{G}, C; \sigma, R \in \mathbb{Z}_p) \ : \ C = g^\sigma h^R \wedge \sigma \in \text{"}\textit{predetermined allowed range}\text{"}\}$$

The range which σ is proved to belong to vary between different constructions and will be more precisely defied below for the separate constructions.

A range proof consists of two parties the *prover*, denoted $\mathcal{P}$, and the verifier, denoted $\mathcal{V}$. After successfully performing a range proof $\mathcal{P}$ has convinced $\mathcal{V}$, that the secret σ in a commitment C is in an predetermined allowed range (or set) without the verifier learning anything else about σ. Note that this means that a range proof is a zero knowledge proof, and this property needs to be proved for each the following constructions.

In the subsections below theory and construction of some range proofs will be presented, these will then be compared and evaluated in section 3.

%subsectionSquare based

2.3.1 Signature-based constructions

The first two constructions that will be considered here are based on the range proofs presented in [3] and adjusted to a non-interactive construction described by

[8]. The transformation from a interactive protocol to a non interactive is done via the Fiat-Schit principle [**?**]. Non-interactive means that there communication between the prover and verifies while XXX Find.

Construction 2 is a non interactive set membership proof of a Pedersen commitment $C = g^\sigma h^R$, where σ is the secret and $R \in_R \mathbb{F}$ is chosen uniformly at random. The construction allows a prover that knows the secret σ to prove for a verifyer, who knows only C that $\sigma \in \Phi$ for some predetermined set Φ without revealing any other information regarding the secret σ. This construction fulfils the zero knowledge requirements described in **??** and is a non interactive zero knowledge set membership proof.

Construction 2 : Non interactive set membership proof

Goal: Given a Pedersen commitment $C = g^\sigma h^R$ and a set Φ, prove that the secret σ belongs to the set Φ without revealing anything else about σ.

- **SetUp** $(g, h, \Phi) \rightarrow (y, \{A_i\}_{i\in\Phi})$
  Pick uniformly at random $\sigma \in_R \mathbb{G}$. Define $y = g^\sigma$ and $A_i = g^{\frac{1}{\sigma+i}}$ $\forall i \in \Phi$, publish y and $\{A_i\}_{i\in\Phi}$.
- **Prove** $(g, h, C, \Phi) \rightarrow$ *proof* $= (V, a, D, z_\sigma, z_\tau, z_R)$
  Pick uniformly at random $\tau \in_R \mathbb{F}$, choose from the set $\{A_i\}$ the element A_σ and calculate $V = A_\sigma^\tau$. Pick uniformly random three values $s, t, m \in_R \mathbb{F}$. Put $a = e(V, g)^{-s} e(g, g)^t$, $D = g^s h^m$ and $c = \text{Hash}(V, a, D)$. Finally compute $z_\sigma = s - \sigma c$, $z_R = m - Rc$ and $z_\tau = t - \tau c$ then construct and publish *proof* $= (V, a, D, z_\sigma, z_R, z_\tau)$.
- **Verify** $(g, h, C = g^\sigma h^R, \textit{proof}) \rightarrow \{0, 1\}$ Check if $D \overset{?}{=} C^c h^{z_R} g^{z_\sigma} \wedge a \overset{?}{=} e(V, g)^c e(V, g)^{-z_\sigma} e(g, g)^{z_\tau}$. If the equality holds the prover has shown that $\sigma \in \Phi$ then return 1 otherwise return 0, since $\sigma \notin \Phi$.

The above construction can be turned into a efficient zero knowledge range proof by rewriting the secret σ into base u such that,

$$\sigma = \sum_{j=0}^{l} \sigma_j u^j.$$

Optimal choice of the two parameters u, l is described in [**?**]. Using this notation if follows that if $\sigma_j \in [0, u)$ $\forall j \in [0, l]$, then $\sigma \in [0, u^l)$. The following construction is a modification of construction 2 into a non interactive zero knowledge range proof using the above decomposition of the secret σ.

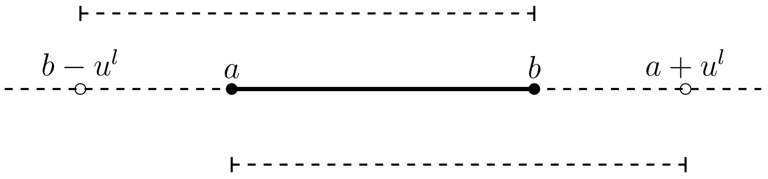

Figure 2.1: Illustration of generalisation to arbitrary intervals $[a, b]$ for range proofs

Construction 3 : Non interactive range proof

Goal: Given a Pedersen commitment $C = g^\sigma h^R$ and two parameters u, l, prove that the secret $\sigma = \sum_{j=0}^{l} \sigma_j u^j$ belongs to the interval $[0, u^l)$ without revealing anything else about σ.

- **SetUp** $(g, h, u, l) \rightarrow (y, \{A_i\})$
  Pick uniformly at random $\chi \in_R \mathbb{Z}_p$. Define $y = g^\chi$ and $A_i = g^{\frac{1}{\chi+i}}\ \forall i \in \mathbb{Z}_u$, publish y and $\{A_i\}$.
- **Prove** $(g, h, u, l, C = g^\sigma h^R) \rightarrow$ *proof*$= (\{V_j\}, \{a_j\}, D, \{z_{\sigma_j}\}, \{z_{\tau_j}\}, z_R)$
  Fist put D to be the identity element in $\mathbb{G}$. Then for every $j \in \mathbb{Z}_l$: pick uniformly at random $\tau_j \in_R \mathbb{Z}_p$ and compute $V_j = A_{\sigma_j}^{\tau_j}$. Then pick uniformly at random three more values $s_j, t_j, m_j \in_R \mathbb{Z}_p$ and compute $a_j = e(V_j, g)^{-s_j} e(g, g)^{t_j}$, $D = D g^{\sigma_j s_j} h^{m_j}$ Given these computations for all $j \in \mathbb{Z}_l$ let $c = \text{Hash}(\{V_j\}, \{a_j\}, D)$. Then for all $j \in \mathbb{Z}_l$ compute $z_{\sigma_j} = s_j - \sigma_j c$,$z_{\tau_j} = t_j - \tau_j c$. Compute $z_R = m - Rc$, where $m = \sum_{j \in \mathbb{Z}_l} m_j$. Finally publish *proof*$= (\{V_j\}, \{a_j\}, D, \{z_{\sigma_j}\}, \{z_{\tau_j}\}, z_R)$
- **Verify** $(g, h, C, \mathit{proof}) \rightarrow \{0, 1\}$
  Check if $D \overset{?}{=} C^c h^{z_R} \prod_{j \in \mathbb{Z}_l} g^{z_{\sigma_j}} \wedge a_j \overset{?}{=} e(V_j, g)^c e(V_j, g)^{-z_{\sigma_j}} e(g, g)^{z_{\tau_j}}$ for all $j \in \mathbb{Z}_l$. If the equality holds the prover has shown that $\sigma \in [0, u^l]$ then return 1 otherwise return 0, since $\sigma \notin [0, u^l)$.

The construction can be generalised to be used for verifying a secret σ is in an arbitrary interval $[a, b]$ where $a > 0,\ b > a$. By verifying that $\sigma \in [a, a + u^l)$ and $\sigma \in [b - u^l, b)$, it must hold that $\sigma \in [a, b]$. Figure 2.1 illustrates the intuition and correctness of the transformation. Proving $\sigma \in [a, a + u^l)$ and $\sigma \in [b - u^l, b)$ can easily be transferred into proving $\sigma - a \in [0, u^l)$ and $\sigma - b + u^l \in [0, u^l)$, since both a, b are public. Therefore to extend the construction two a arbitrary interval the steps **Prove** and **Verify** will have to be run twice but the step **SetUp** will only have to be executed once. In this approach for general intervals `AND` operation will have to be executed to verify that the secret is a interval where the lower bound is non-zero. In [4] an optimised implementation reduces the complexity with a factor 2 compared to [3] for the case of non-zero lower bound of the allowed interval. This rather small reduction of a factor 2, makes a considerable difference for a verifier that checks multiple clients honesty, which is the case in VAHSS.

2.3.2 Bulletproofs

The construction of bulletproofs presented in [2] is presented. This construction is based on a inner-product argument.

Notation

The description and construction or bulletproofs requires some additional notation which will be presented here. First let lowercase bold font denote vectors, i.e $\mathbf{a} \in \mathbb{F}$ is a vector with element $a_1, .., a_n \in \mathbb{F}$, and uppercase bold font denote matrices, i.e $\mathbf{A} \in \mathbb{F}^{n \times m}$ is a matrix and a_{ij} the element of $\mathbf{A}$ at row i and column j. Given this notation denote scalar multiplication with a vector as $\mathbf{b} = c \cdot \mathbf{a} \in \mathbb{F}$, where $c \in \mathbb{F}$ and $\mathbf{b} = (b_1, ..., b_n)$ where $b_i = c \cdot a_i$. Denote the euclidean inner product of two vectors as $\langle \mathbf{a}, \mathbf{b} \rangle$ and Hadamard product as $\mathbf{a} \circ \mathbf{b}$.

Further consider vector polynomials $p(X)$ of degree d on the form $p(X) = \sum_{i=0}^{d} \mathbf{p_i} \cdot X^i \in \mathbb{Z}_p^n[X]$, where the coefficients $\mathbf{p_i} \in \mathbb{Z}_p^n$. The inner product of two vector polynomials, $l(X), r(X)$ is defined as,

$$\langle l(X), r(X) \rangle = \sum_{i=0}^{d} \sum_{j=0}^{n} \langle l_i, r_j \rangle \cdot X^{i+j} \in \mathbb{Z}_p[X].$$

The following is equivalent: evaluating two polynomials at x then taking the inner product versus taking the inner product polynomial at x.

Let $\mathbf{a}||\mathbf{b}$ denote the concatenation of two vectors and for $0 \leq l \geq n$ use python notation to denote sections of vectors such that $\mathbf{a}_{[:l]} = (a_1, ..., a_l)$ and $\mathbf{a}_{[l:]} = (a_{l+1}, ..., a_n)$.

For $k \in \mathbb{Z}_p^*$ let $\mathbf{k}^n = (1, k, k^2, ..., k^{n-1})$, i.e the vector containing the n fist powers of k.

Also let $\mathbf{g}, \mathbf{h} \in \mathbb{G}^n$ be two vectors. Given such a vector $\mathbf{g}$ and a vector $\mathbf{a} \in \mathbb{Z}_p^n$ write $C = \mathbf{g}^{\mathbf{a}} = \prod_{i=1}^{n} g_i^{a_i} \in \mathbb{G}$. C can be interpreted as a commitment to the vector $\mathbf{a}$.

Remark that is this section n denotes the dimension of the room not the number of clients, further remark that the dimension of the room is the length of the bit representation of the secret in the Pedersen vector commitment considered below.

Inner product argument

The bulletproof construction is based on the inner product argument presented in this section. The inner product argument is a argument of knowledge of $\mathbf{s}, \mathbf{r}$ in a Pedersen vector commitment $P_v = \mathbf{g}^{\mathbf{s}} \mathbf{h}^{\mathbf{r}}$ satisfying a given inner product. (To differ from the Pedersen vector commitment considered here and the Pedersen commitment in the range proofs the exponents in the commit are denoted $\mathbf{s}, \mathbf{r}$ instead of σ, R, and the commitment by P_v) More formally the argument is a proof system of the statement,

$$\{(\mathbf{g}, \mathbf{h} \in \mathbb{G}^n, P_v \in \mathbb{G}, c \in \mathbb{Z}_p;\ \mathbf{s}, \mathbf{r} \in \mathbb{Z}_p^n) :\ P_v = \mathbf{g}^{\mathbf{s}} \mathbf{h}^{\mathbf{r}} \wedge\ c = \langle \mathbf{s}, \mathbf{r} \rangle\}$$

Which can be shown to be equivalent to a proof of the statement,

$$\{(\mathbf{g}, \mathbf{h} \in \mathbb{G}^n, u, P_v \in \mathbb{G};\ \mathbf{s}, \mathbf{r} \in \mathbb{Z}_p^n) :\ P_v = \mathbf{g}^{\mathbf{s}} \mathbf{h}^{\mathbf{r}} u^{\langle \mathbf{s}, \mathbf{r} \rangle}\}.$$

The construction to prove the inner product argument is presented in Construction 4. The construction presented is modified compared to the one presented in [2] to be non-interactive using the Fiat-Shamir heuristic by [8].

Construction 4 : Inner-product argument

Goal: Given a Pedersen commitment $C = g^{\sigma} h^{R}$ and two parameters u, l, prove that the secret $\sigma = \sum_{j=0}^{l} \sigma_j u^j$ belongs to the interval $[0, u^l)$ without revealing anything else about σ.

- **Prove** $(\mathbf{g}, \mathbf{h}, P_v = \mathbf{g}^{\mathbf{s}} \mathbf{h}^{r}, c, \mathbf{r}, \mathbf{s}) \rightarrow \; \mathit{proof}_{IP} = (\mathbf{g}, \mathbf{h}, P_v', u^x, \mathbf{s}, \mathbf{r}, \mathbf{l}, \mathbf{r})$
  Let $x = \text{Hash}(\mathbf{g}, \mathbf{h}, P_v, c) \in \mathbb{Z}_p^*$ and compute $P_v' = u^{x \cdot c} P$. Then define the two vectors $\mathbf{l}, \mathbf{r}$.
  - If the dimension of the vectors $\mathbf{g}, \mathbf{h}, \mathbf{s}, \mathbf{r}$ is one drop the bold font in the notation and publish the proof $\mathit{proof}_{IP} = (g, h, P_v', u^x, s, r, \mathbf{l}, \mathbf{r})$.
  - Otherwise: Let $n' = n/2$ and define $c_L = \langle a_{[:,n']}, b_{[n',:]} \rangle$ and $c_R = \langle a_{[n',:]}, b_{[:,n']} \rangle$. Then use these variables to calculate $L = \mathbf{g}_{[n':]}^{\mathbf{a}_{[:n']}} \mathbf{h}_{[:n']}^{\mathbf{b}_{[n':]}} u^{c_L}$ and $R = \mathbf{g}_{[:n']}^{\mathbf{a}_{[n':]}} \mathbf{h}_{[n':]}^{\mathbf{b}_{[:n']}} u^{c_R}$. Further store the current values of $L, R \in \mathbb{G}$, by appending them to the vectors $\mathbf{l}$ resp $\mathbf{r}$. Now update $x = \text{Hash}(L, R)$, and recalculate $\mathbf{g} = \mathbf{g}_{[:n']}^{x^{-1}} \mathbf{g}_{[n':]}^{x}$, $\mathbf{h} = \mathbf{h}_{[:n']}^{x} \mathbf{h}_{[n':]}^{x^{-1}}$ and the commitment $P' = L^{x^2} P R^{x^{-2}}$. Finally update the exponents $\mathbf{s}, \mathbf{r}$ to $\mathbf{s} = \mathbf{s}_{[:n']} + \mathbf{s}_{[n':]} x^{-1}$ and $\mathbf{r} = \mathbf{r}_{[:n']} x^{-1} + \mathbf{r}_{[n':]} x$. Run the step $\mathbf{Prove}(\mathbf{g}, \mathbf{h}, P_v', n', \mathbf{r}, \mathbf{s})$ with the updated variables. Note that the vectors $\mathbf{g}, \mathbf{h}, \mathbf{s}, \mathbf{r}$ now have the dimension $n' = n/2$, hence performing the recursion until one-dimensional vectors will require $log\, n$ iterations.
- **Verify** $(g, h, C, \mathit{proof}) \rightarrow \{0, 1\}$
  For $i \in \{0, log(n)\}$ put $n = n/2$ and $x = \text{Hash}(\boldsymbol{l}[i], \boldsymbol{r}[i])$, then update the vectors $\boldsymbol{g}$ and $\boldsymbol{h}$ as well as the variable P according to, $\boldsymbol{g} = \boldsymbol{g}_{[:,n]}^{x^{-1}} \boldsymbol{g}_{[n,:]}^{x}$, $\boldsymbol{h} = \boldsymbol{h}_l[:, n]^x \boldsymbol{h}_{[n,:]}^{x^{-1}}$ and $P = L^{x^2} P R^{x^{-2}}$. After iterating over all i the dimension of the vectors $\boldsymbol{g}, \boldsymbol{h}$ is one and we can drop the bold font. Accept if $c = \langle s, r \rangle$ and $P = g^s h^r$.
  Accept if

Inner product rang proof

Based on the inner product argument in this section the construction a range proof called *bulletproof*, will be described. This construction allows a prover, given a Pedersen commitment $C = g^{\sigma} h^{R}$ to convince a verifier that the secret σ belongs to the interval $[0, 2^n)$. To do this the prover needs to convince the verifier that:

- $\boldsymbol{\sigma} \in \{0, 1\}^n$ is the binary representation of σ, or equivalently that $\langle \boldsymbol{\sigma}, \mathbf{2}^n \rangle = \sigma$.
- $\bar{\boldsymbol{\sigma}}$ is the component-wise complement of $\boldsymbol{\sigma}$. This is equivalent to show that $\bar{\boldsymbol{\sigma}}$ satisfies the two conditions: $\bar{\boldsymbol{\sigma}} \circ \boldsymbol{\sigma} = \mathbf{0}^n$ and $\bar{\boldsymbol{\sigma}} = \boldsymbol{\sigma} - \mathbf{1}^n \bmod 2$.

These three equations can be rewritten and summarised in proving the following statement;

$$\left\langle \boldsymbol{\sigma} - z \cdot \mathbf{1}^n, \boldsymbol{y}^n \circ (\bar{\boldsymbol{\sigma}} + z \cdot \mathbf{1}^n) + z^2 \cdot \mathbf{2}^n \right\rangle = z^2 \cdot \sigma + \delta(y, z), \tag{2.1}$$

where $\delta(y, z) = (z - z^2) \cdot \langle \mathbf{1}^n, \boldsymbol{y}^n \rangle - z^3 \langle \mathbf{1}^n, \mathbf{2}^n \rangle \in \mathbb{Z}_p$.. The values z and y are either chosen at random from the set $\mathbb{Z}_p$ by the verifier in an interactive construction or are the hash of other values in a non-interactive construction. Here a non-interactive construction will be considered, for further definition of the variables z, y see Construction **??**.

If a inner product argument presented in Construction 4 was used to prove the inner product defined in equation (2.1) it would leak information about σ, since information about the two vectors $\boldsymbol{\sigma}, \bar{\boldsymbol{\sigma}}$ is revealed and they contain information about the binary representation of σ. Hence two new vectors $\boldsymbol{s}_1, \boldsymbol{s}_2$ are introduced and will serve as blinding vectors and help construct a zero-knowledge range proof even if the inner product argument is not a zero knowledge construction. Given this idea, the inner product in (2.1) is tweaked to include the two blinding vectors,

$$t(X) = \langle l(X), r(X) \rangle = t_0 + t_1 \cdot X + t_2 \cdot X^2 \tag{2.2}$$

$$l(X) = \boldsymbol{\sigma} - z \cdot \mathbf{1}^n + \boldsymbol{s}_1 \cdot X \tag{2.3}$$

$$r(X) = \boldsymbol{y}^n \circ (\bar{\boldsymbol{\sigma}} + z \cdot \mathbf{1}^n + \boldsymbol{s}_2 \cdot X) + z^2 \cdot \mathbf{2}^n, \tag{2.4}$$

$$\tag{2.5}$$

Clearly $t_0 = z^2 \cdot \sigma + \delta(y, z)$ which is equal to the right hand side of equation (2.1).

3
Methods

Evaluate and choose one to implement, do implementation and construct proofs

3.1 Comparison of constructions for verifying clients input

In this section the different constructions presented in section 2.3 will be evaluated and compared in order to decide which method is best to combine with the VHASS scheme described in Construction**??** to check clients input. Each range proof constructions pros and cons will be discussed separately but tables for comparison will also be presented. Then a final comparison will be made.

The aspects that will be considered in the evaluation of the range proofs and their compatibility with the VHASS construction is presented is the below list;

- Proof size
- Communication complexity
- Flexibility of range
- Assumptions and requirements
- Computation complexity for prover resp. verifier

Remark that all the range proof considered aim to prove that the secret in a Pedersen commitment is in an allowed range. Thus to combine any of the range proofs with the VHASS construction, the clients needs beyond previously computed and published values also publish a Pedersen commitment. This is investigated further in section **??**. Another remark is that all range proofs considered have be made non interactive using the Fiat-Shamir heuristic, even if they were originally presented as interactive constructions.

3.1.1 Signature-based range proof

flexible, sets and arbitrary range proofs. sends XXX. Signature is $\mathcal{O}(n)$ or using $\sigma = \sum_{k=1}^{j} \sigma_j u^j$ we have $\mathcal{O}(\frac{n}{log\, n - log\, log\, n})$
Third party?

3.1.2 Bulletproof

bullet is $\mathcal{O}(log\, n)$. Bulletproof not general range. No third party? logarithmic size, linear prove and verifucation time'?

bullet proof
signature
square

Table 3.1: Caption

3.2 Additive homomorphic secret sharing with verification of both clients and severs

Construction 5 XX

Require:

Ensure:

1.

3.3 Proofs

Theorem 4 (Correctness)**.** $Pr[]$

Proof. To prove XXX it is sufficient to show that $\sigma = \prod_{i=1}^{n} \tau_i \wedge \prod_{i=1}^{n} \tau_i = \mathcal{H}(y)$. For y and σ we have the same construction as in [11]. Hence by construction we have:

$$y = \sum_{j=1}^{m} y_j = \sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_{ij} p_i(\theta_{ij}) = \sum_{i=1}^{n} \overbrace{\Big(\sum_{j=1}^{m} \lambda_{ij} p_i(\theta_{ij}) \Big)}^{p_i(0)} = \sum_{i=1}^{n} p_i(0) = \sum_{i=1}^{n} x_i, \tag{3.1}$$

and for σ it holds that:

$$\sigma = \prod_{j=1}^{m} \sigma_j = \prod_{j=1}^{m} g^{y_j} = g^{\sum_{j=1}^{m} y_j} = g^y = \mathcal{H}(y)$$

For the τ_i, whose construction has been modified compared to [11] we have:

$$\prod_{i=1}^{n} \tau_i = \prod_{i=1}^{n} \mathbb{E}(x_i, R_i) = \prod_{i=1}^{n} g^{x_i} h^{R_i} = g^{\sum_{i=1}^{n} x_i} h^{\sum_{i=1}^{n} R_i} \overset{(3.1)}{=} g^y h^{\sum_{i=1}^{n-1} R_i + R_n} =$$
$$= g^y h^{\phi(N) \left\lceil \frac{\sum_{i=1}^{n-1} R_i}{\phi(N)} \right\rceil} \overset{*}{=} g^y = \mathcal{H}(y) \quad \textit{*- since } h \textit{ is co-prime to } N.$$

□

3.4 Implementation

4

Results

4.1 Runtime and complexity

5

Conclusion

5.1 Discussion

Limit, only considerd range proof using pedersen commitment scheme.

FFS for intervals: Need comunication between servers. We do not want

5.2 Conclusion

Bibliography

[1] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security*, CCS '93, page 62–73, New York, NY, USA, 1993. Association for Computing Machinery.

[2] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 315–334, 2018.

[3] J. Camenisch, R. Chaabouni, and a. shelat. Efficient protocols for set membership and range proofs. In J. Pieprzyk, editor, *Advances in Cryptology - ASIACRYPT 2008*, pages 234–252, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[4] R. Chaabouni, H. Lipmaa, and A. Shelat. Additive combinatorics and discrete logarithm based range protocols. In R. Steinfeld and P. Hawkes, editors, *Information Security and Privacy*, pages 336–351, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[5] G. Couteau, T. Peters, and D. Pointcheval. Removing the strong rsa assumption from arguments over the integers. In J.-S. Coron and J. B. Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 321–350, Cham, 2017. Springer International Publishing.

[6] D. Froelicher, J. R. Troncoso-Pastoriza, J. S. Sousa, and J. Hubaux. Drynx: Decentralized, secure, verifiable system for statistical queries and machine learning on distributed datasets. *CoRR*, abs/1902.03785, 2019.

[7] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[8] E. Morais, T. Koens, C. van Wijk, and A. Koren. A survey on zero knowledge range proofs and applications. *SN Applied Sciences*, (946), 2019.

[9] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In J. Feigenbaum, editor, *Advances in Cryptology — CRYPTO '91*, pages 129–140, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.

[10] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, Nov. 1979.

[11] G. Tsaloli, G. Banegas, and A. Mitrokotsa. Practical and provably secure distributed aggregation: Verifiable additive homomorphic secret sharing. *Cryptography*, 4(3), 2020.

[12] G. Tsaloli and A. Mitrokotsa. Sum it up: Verifiable additive homomorphic secret sharing. In J. H. Seo, editor, *Information Security and Cryptology – ICISC 2019*, pages 115–132, Cham, 2020. Springer International Publishing.

[13] H. Yao, C. Wang, B. Hai, and S. Zhu. Homomorphic hash and blockchain based

authentication key exchange protocol for strangers. In *2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)*, pages 243–248, 2018.

A

Appendix 1