KU LEUVEN

FACULTY OF
ENGINEERING SCIENCE

The best master's thesis ever

First Author
Second Author

Thesis submitted for the degree of
Master of Science in Cybersecurity

Supervisor
Prof. dr. ir. Knows Better

Assessors
Ir. Kn. Owsmuch
K. Nowsrest

Assistant-supervisors
Ir. An Assistent
A. Friend

Academic year 2024 – 2025

Published by First Author and Second Author,
Faculty of Engineering Science, Kasteelpark Arenberg 1 bus 2200, B-3001 Leuven

Preface

I would like to thank everybody who kept me busy the last year, especially my promoter and my assistants. I would also like to thank the jury for reading the text. My sincere gratitude also goes to my wive and the rest of my family.

First Author
Second Author

Contents

Abstract

The `abstract` environment contains a more extensive overview of the work. But it should be limited to one page.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

List of Figures and Tables

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

DL	Discrete Logarithm
PPT	Probabilistic Polynomial Time
NIZK	Non-Interactive Zero Knowledge
PoK	Proof of Knowledge
AoK	Arguement of Knowledge
PSSS	Packed Shamir Secret Sharing
PVSS	Publicly Verifiable Secret Sharing
PPVSS	Pre-Constructed Publicly Verifiable Secret Sharing
PPPVSS	Packed Pre-Constructed Publicly Verifiable Secret Sharing

Symbols

q	prime number
$\mathbb{G}$	Cyclic group of order q
$\mathbb{Z}_q$	Modular ring with q elements
$\mathbb{Z}_q[X]$	Univariate polynomial ring in the variable X with coefficients in $\mathbb{Z}_q$
λ	Security Parameter
$negl$	Negligible function
$\mathcal{O}$	Big-O notation

Chapter 1

Literature Review

In 1979, Shamir introduced a threshold secret sharing scheme called Shamir Secret Sharing scheme [14], which is now a well-known and widely used secret sharing scheme to this day because of its numerous applications in cryptography. It was first of its kind to have Information Theoretic (IT) security under certain assumptions against passive adversaries who can only see the secret shares of the parties they have corrupted. In reality, however, the adversaries are usually stronger than just being passive, moreover, they possess the power to manipulate the share values of the corrupted parties itself. Shamir's scheme is not tailored to defend against active adversaries as one cannot verify the correctness of the shares. This led to numerous inventions of Verifiable Secret Sharing (VSS) schemes, which not only does allow the parties to verify the correctness of the shares shared by the dealer but also allows the parties to verify the correctness of the shares when opened by the parties during the reconstruction phase. Because of the feature of verifiability, VSS schemes can defend the applications against active adversaries.

There are many VSS schemes ([8], [9]) in the literature which are based on Shamir Secret Sharing scheme. Throughout the years, many advancements have been made in the field of VSS schemes, and as of writing this report the efficient VSS schemes are Π_F, Π_P and Π_{LA} [2], each of which have distinct security features. In VSS, only shareholders can actually verify the correctness of the shares. Certain applications demand to have verifiability feature available to anyone, which is solved by Publicly Verifiable Secret Sharing (PVSS) schemes. PVSS is an extension of VSS, where the correctness of the shares can be verified by anyone. Many cool applications exist today which use PVSS schemes, such as, e-voting [13], randomness beacons [5], etc. In [3], authors have noticed that the Schoenmakers' PVSS scheme used for the e-voting application in [13] is actually more than a PVSS scheme, and they coined the term Pre-Constructed Publicly Verifiable Secret Sharing (PPVSS) scheme. PPVSS is a special type of PVSS where the dealer additionally publishes a commitment to the secret itself. The authors have also shown that any PVSS scheme can be transformed into a PPVSS scheme with minimal changes, and constructed a PPVSS Λ_{RO} from the PVSS Π_S [2] as an example, where they used Λ_{RO} to build an efficient e-voting application.

With PPVSS, one can build versatile applications and also can improve the efficiency of existing applications. In ALBATROSS [6], authors built a randomness beacon application using a PVSS. We have an intuition that an efficient randomness beacon application can be built using a scheme based on PPVSS on certain conditions. In this report, we will introduce Packed PPVSS (PPPVSS) along with its security proofs and give an example based on Λ_{RO}, which will be used to improve ALBATROSS in many cases.

Chapter 2

Preliminaries

2.1 Notation

Let $\mathbb{G}$ be a cyclic group of prime order q with hard Discrete Log (DL) and its generator being g. Also, we write $\mathbb{Z}_q[X]_d$ to denote the set of all d degree polynomials univariate in X with coefficients in the finite field $\mathbb{Z}_q$.

2.2 Coding Theory

This subsection is a brief recall of linear codes and their properties.

Definition 2.2.1 (Linear Code). *If $\mathcal{C}$ be a vector subspace of $\mathbb{Z}_q^n$ with dimension k, then $\mathcal{C}$ is said to be a **linear code**(/ linear $q-$ary code) of length n and dimension k.*

In the remainder of the subsection, we let $\mathcal{C}$ be a linear $q-$ary code of length n and dimension k.

Definition 2.2.2 (Dual Code). *The vector subspace $\mathcal{C}^\perp$ is called a Dual (Code) of $\mathcal{C}$ if it is orthogonal to $\mathcal{C}$.*

Definition 2.2.3 (Generating Matrix). *The $k \times n-$matrix $\mathcal{G}$ is said to be a generating matrix of $\mathcal{C}$ if it generates $\mathcal{C}$, more precisely, the rows of G form a basis for $\mathcal{C}$. Also, $\mathcal{G}$ is said to be in its **standard form** if it is of the form*

$$\mathcal{G} = \begin{bmatrix} I_k & P \end{bmatrix},$$

where I_k is the $k \times k$ identity matrix and P is some $k \times (n-k)$ matrix.

Definition 2.2.4 (Parity Check Matrix). *Consider the linear transformation ϕ as follows:*

$$\phi : \mathbb{Z}_q^n \to \mathbb{Z}_q^{n-k},$$

*where kernel of ϕ is $\mathcal{C}$. Then the matrix associated to ϕ, $\mathcal{H}$, is called the **parity check matrix** of $\mathcal{C}$.*

Lemma 2.2.1. *If $\mathcal{G}$ being a generating matrix of $\mathcal{C}$ and say it is in its standard form, i.e., $\mathcal{G} = \begin{bmatrix} I_k & P \end{bmatrix}$, then $\mathcal{H}$ being a parity check matrix of $\mathcal{C}$ is given by*

$$\mathcal{H} = \begin{bmatrix} -P^T & I_{n-k} \end{bmatrix},$$

where I_{n-k} is the $(n-k) \times (n-k)$ identity matrix and P^T is the transpose of P.

2.2.1 Reed Solomon Codes

2.3 Packed Shamir Secret Sharing

(n, t, ℓ)-Packed Shamir secret sharing ([11],[4]) scheme is a threshold secret sharing scheme which is a variant of (n, t)-Shamir's secret sharing scheme [14]. In a nutshell, the $t + \ell - 1$ degree secret polynomial with coefficients in $\mathbb{Z}_q$ which evaluates to ℓ secrets is secret shared amongst n parties such that any $t+\ell$ parties can reconstruct back the secret polynomial. Recall that Shamir's secret sharing scheme requires at least $t + 1$ parties to reconstruct the secret polynomial in contrast to the $t + \ell$ parties in the Packed Shamir secret sharing scheme. The scheme is summarized in the Figure 2.1.

2.4 Sigma Protocols

The agenda of this subsection is to give a brief formal background about some important primitives used in the PVSS ,Π_S [2], and the PPVSS ,Λ_{RO} [3], schemes. Let X and W be two sets with R being a relation on $X \times W$, and $L = \{x \in X : \exists w \in W, xRw\}$ be the language defined by R where xRw says that w is a witness for a given $x \in L$. Also, let $\mathcal{R}$ be a PPT algorithm such that $\mathcal{R}(1^\lambda)$ outputs pairs (x, w) with $x \in L$ and xRw where λ is a security parameter.

Given a relation R and its corresponding language L, a **Sigma ($\sum$) Protocol** is a 3-round *interactive* protocol between two Probabilistic Polynomial Time (PPT) algorithms, a prover P and a verifier V. For some $x \in L$ with xRw, in the first round P sends a commitment a to V. To which V sends a challenge d to P in the second round and finally P responds back with the response z to V in the third round. V outputs **true** or **false** upon the proof verification on transcript $trans := (a, d, z)$. Informally, with a $\sum-$protocol a prover P tries to convince a verifier V that they know a witness w for a given statement $x \in L$ without revealing any information about w. To state it formally, a $\sum-$protocol is supposed to satisfy *completeness*, *Honest Verifier Zero Knowledge* (HVZK) and *Special Soundness* which are defined as follows.

Definition 2.4.1 (Completeness)**.** *A $\sum-$protocol is said to be* ***complete*** *for $\mathcal{R}$ if the verifier V always accepts the honest prover P for any $x \in L$.*

Definition 2.4.2 (HVZK)**.** *A $\sum-$protocol is said to be* ***HVZK*** *for $\mathcal{R}$ if there exist a PPT algorithm S that simulates trans of the scheme corresponding to a given*

Packed Shamir Secret Sharing

Given ℓ secrets to share amongst n parties, where at most t of them can be *(passively)* corrupt, the (n, t, ℓ)-Packed Shamir secret sharing scheme description is as follows:

Sharing Algorithm:

- Dealer constructs the secret polynomial $f \in \mathbb{Z}_q[X]_{t+l-1}$ via the lagrange interpolation by choosing $t + \ell$ elements in $\mathbb{Z}_q$ where ℓ of them are secrets, $\{s_i\}_{i=0}^{\ell-1}$, with $f(-i) = s_i$ for all i and remaining t are chosen uniformly at random in $\mathbb{Z}_q$.
- Each party P_i receives their share $f(i)$ from the Dealer for each $i \in \{1, \ldots, n\}$

Reconstruction Algorithm:

- Any Q set containing at least $t + \ell$ parties can use the lagrange interpolation to compute $\{s_i\}_{i=0}^{\ell-1}$ as follows:

$$s_m = \sum_{i \in Q} f(i) \left[\prod_{j \in Q, j \neq i} \frac{-m - j}{i - j} \right] \qquad , m \in \{0, \ldots, \ell - 1\}$$

- The secrets $\{s_i\}_{i=0}^{\ell-1}$ are outputted as the result.

FIGURE 2.1: Packed Shamir Secret Sharing

$x \in L$ with any witness w of x. That is, given $x \in L$,

$$trans(P(x, w) \leftrightarrow V(x)) \approx trans(S(x) \leftrightarrow V(x)) \qquad \textit{, for any witness } w \textit{ of } x.$$

Where $trans(P(\cdot) \leftrightarrow V(\cdot))$ is the transcript of the $\sum -$protocol amongst P and V and $\approx$ denotes the indistinguishability of the two transcripts.

Definition 2.4.3 (Special Soundness)**.** *A $\sum -$protocol is said to satisfy **Special Soundness** for $\mathcal{R}$, if there exists a PPT extractor $\mathcal{E}$ for any two valid transcripts, (a, d, z) and (a, d', z'), corresponding to a given $x \in L$ with only a unique witness w and $d \neq d'$ such that $\mathcal{E}(a, d, z, d', z')$ outputs the witness w.*

It is shown that a public-coin, complete, HVZK, special soundness $\sum -$protocol can be made into a Non Interactive Zero Knowledge (NIZK) Proof of Knowledge (PoK) or Argument of Knowledge (AoK) in the Random Oracle (RO) model using Fiat-Shamir transform [10]. In the following subsections, we recall two important NIZK PoK schemes which are used in Π_S and Λ_{RO} schemes.

2.4.1 Chaum-Pedersen Protocol for DL Equality

Recall $\mathbb{G}$ being the cyclic group of prime order q with hard Discrete Logarithm (DL). For some $g, h \in \mathbb{G}$ consider the following relation:

$$R_{DLEQ} = \{(g, h, a, b), x : a = g^x, b = h^x\}.$$

In [7], Chaum and Pedersen proposed a NIZK PoK scheme for the DL Equality relation, R_{DLEQ}. Informally, a prover P can convince a verifier V that they know x such that it can be used with both g and h to obtain a and b respectively. This protocol is widely used in many cryptographic applications like threshold decryption, e-voting and Randomness Beacons. We summarize the protocol in Figure 2.2.

Chaum-Pedersen Protocol for DLEQ

Let $(g, h, a, b) \in L_{DLEQ}$ be a statement with its corresponding witness being x where L_{DLEQ} is the language defined by the relation R_{DLEQ}.

Prover

- Samples $r \in_R \mathbb{Z}_q$ uniformly at random and sets $c_1 = g^r$ and $c_2 = h^r$.
- Sets $d \leftarrow \mathcal{H}(a, b, c_1, c_2)$, where $\mathcal{H}$ is an agreed upon Random Oracle (RO).
- Sets $z \equiv r + dx \pmod q$ and returns the proof(/transcript) $\pi := (d, z)$.

Verifier

- Checks if $d \leftarrow \mathcal{H}(a, b, \frac{g^z}{a^d}, \frac{h^z}{b^d})$ and outputs **true** or **false** accordingly.

FIGURE 2.2: Chaum-Pedersen NIZK PoK for DLEQ

2.4.2 NIZK PoK for Polynomial DL

Recall $\mathbb{G}$ being the cyclic group of prime order q with hard Discrete Logarithm (DL) and g being its generator. Consider the following relation for some polynomial $f \in \mathbb{Z}_q[X]_t$ with degree $t < n$:

$$R_{PDL} = \{(g, x_1, \ldots, x_n, F(x_1), \ldots, F(x_n)), f(X) : F(x_i) = g^{f(x_i)}, 1 \leq i \leq n\}.$$

In [2], Baghery formally introduced a NIZK PoK scheme for the Polynomial DL relation, R_{PDL}, which is a generalization of Schnorr's ID protocol [12]. Informally, a prover P can convince a verifier V that they know a t degree polynomial f such that it can be used with g to obtain $F(x_i)$ for $1 \leq i \leq n$. This protocol is used to construct the PPVSS Λ_{RO} [3], which was essential in building an efficient e-voting protocol. We summarize the protocol in Figure 2.3.

A NIZK PoK for Polynomial DL

Let $(g, x_1, \dots, x_n, F(x_1), F(x_n)) \in L_{PDL}$ be a statement with its corresponding witness being $f \in \mathbb{Z}_q[X]_t$ where L_{PDL} is the language defined by the relation R_{PDL}.

Prover

- Samples $r \in_R \mathbb{Z}_q[X]_t$ uniformly at random and sets $\Gamma_i = g^{r(x_i)}$ for $1 \leq i \leq n$.
- Sets $d \leftarrow \mathcal{H}(F_1, \dots, F_n, \Gamma_1, \dots, \Gamma_n)$, where $\mathcal{H}$ is an agreed upon Random Oracle (RO).
- Sets $z(X) \equiv r(X) + df(X) \pmod q$ and returns the proof(/transcript) $\pi := (d, z(X))$.

Verifier

- First checks if z is a t degree polynomial in $\mathbb{Z}_q[X]_t$. If so, they proceed with the next step.
- Checks if $d \leftarrow \mathcal{H}(F_1, \dots, F_n, \frac{g^{z(x_1)}}{F_1^d}, \dots, \frac{g^{z(n)}}{F_n^d})$.
- If first two steps are correct then they output **true**, otherwise **false**.

FIGURE 2.3: A NIZK PoK for Polynomial DL based on Schoenmakers' PVSS

2.5 Publicly Verifiable Secret Sharing (PVSS)

Publicly Verifiable Secret Sharing (PVSS) is an extension of Non-Interactive Verifiable Secret Sharing (NI-VSS) scheme. Unlike NI-VSS where only the parties who possess the secret shares can verify the correctness of the secret sharing, anyone including external entities can verify the correctness of the secret sharing in PVSS.

2.6 Pre-Constructed Publicly Verifiable Secret Sharing (PPVSS)

PPVSS was first introduced in [3], which is used as a building block to construct a new e-voting protocol based on Schoenmakers' PVSS [13]. Interestingly, the authors in [3] observed that the original e-voting protocol published in 1999 by Schoenmakers is unusually efficient to be just based on a PVSS, which led them to discover that Schoenmakers PVSS is actually a PPVSS. What sets PPVSS apart from standard PVSS schemes is that it can be used to construct versatile applications, such as e-voting, and can also improve efficiency of some existing protocols. The subtle difference between PPVSS and PVSS is that the secret itself is committed by the prover along with all its corresponding secret shares.

2.7 Conclusion

The final section of the chapter gives an overview of the important results of this chapter. This implies that the introductory chapter and the concluding chapter don't need a conclusion.

Chapter 3

The Next Chapter

Randomness Beacon using PPPVSS

Our protocol with PPPVSS is run between a set $\mathcal{P}$ of n parties $P_1, \ldots, P_n$ who have access to a public ledger where they can post information for later verification. It is assumed that the Setup phase of Π_{PPPVSS} is already done and the public keys pk_i of each party P_i along with $\{\mathbb{P}_i\}_{i=1}^{l}$ being Commitment keys (or public keys of target people) to encrypt the l secrets are already registered in the ledger. In addition, the parties have agreed on a Vandermonde $(n-2t)\times(n-t)$-matrix $M = M(\omega, n-2t, n-t)$ with $\omega \in \mathbb{Z}_q^*$.

1. **Commit:** For $1 \leq j \leq n$:
   - Shareholder P_j executes the Distribution phase of the PPPVSS as Dealer for $\ell = n-2t$ secrets, publishing commitments (/encryptions) of secrets, $y^j_{-(l-1)}, \ldots, y^j_{-1}, y^j_0$, and encryptions of shares $\{y^j_i\}_{i=1}^{n}$ along with π^j_{proof}, which is a NIZK PoK for proving the correctness of committed(/encrypted) secrets and encrypted secret shares on the public ledger, also learning the secrets $h^{s^j_0}, ..., h^{s^j_{-(l-1)}}$ and their corresponding exponents $s^j_0, \ldots, s^j_{-(l-1)}$.
2. **Reveal:**
   - Each shareholder checks the validity of the proof π^j_{proof}, i.e., the **verification phase of PPPVSS protocol**.
   - After a set $\mathcal{C}$ containing at least $n-t$ shareholders publish their shares in the public ledger, $P_j \in \mathcal{C}$ reveals l secrets.
   - Every shareholder verifies the validity of secrets by reproducing the commitments using the commitment keys (/public keys of target people).
   - At this point, if every party in $\mathcal{C}$ has opened their secrets correctly, go to step 4' in Figure **??**. Otherwise, proceed to step 3 in Figure **??**.

Figure 4.1: Commit and Reveal phase of the Randomness Beacon using PPPVSS

Chapter 4

Revisiting a Randomness Beacon Protocol

See table 4.1 for an overview.

- In ALBATROSS, a dealer(as a part of **commit**) should compute $n(\mathbb{E}_x + \mathbb{P}_e)$ commitments and to give a proof he should do an additional $n(\mathbb{P}_e + \mathbb{E}_x)$. Also, on dealer should do $l(\mathbb{P}_e + \mathbb{E}_x)$ for computing secrets and keeping it to himself. In total dealer needs to do $(2n + l)[\mathbb{E}_x + \mathbb{P}_e]$.
  - In **Reveal**, a verifier should compute $2n\mathbb{E}_x$ which internally requires additional $n\mathbb{P}_e$, i.e., in total it requires $(n-1)n(2\mathbb{E}_x + \mathbb{P}_e)$ computations for each verifier.
    - In **Robust case** where t dealers do not open their polynomials, a verifier should verify $n - t$ polynomials of honest dealers, i.e., for each honest dealer, a verifier has to do $n\mathbb{P}_e$ to evaluate secret share exponents and does $n\mathbb{E}_x$ to get secret shares and cross checks them in the public ledger. Also, finally the verifier computes $l\mathbb{P}_e$ to get secret exponents and get l secrets by doing $l\mathbb{E}_x$. As there are $n - t$ honest dealers, the verifier has to compute $(n - t)(n + l)(\mathbb{E}_x + \mathbb{P}_e)$.
    - In **Honest case**, everyone would have been honest and so each verifier has to do $(n - 1)(n + l)(\mathbb{E}_x + \mathbb{P}_e)$.
  - **Recovery** phase only exists if some party does not open the polynomial leading to PVSS reconstruction phase, in the worst case there should be reconstruction for the secrets of t malicious parties. Given a malicious shareholder who has not opened the secret polynomial, each shareholder/reconstructor has to decrypt their share, which requires $1\mathbb{E}_x$ and should give a DLEQ proof that they have decrypted correctly, which additionally requires $2\mathbb{E}_x$; Also the re-constructor should verify DLEQ proofs of correct share decryption from $n - t$ honest shareholders requiring them to do $4(n - t)\mathbb{E}_x$. In total, each re-constructor requires $[3 + 4(n - t)]t\mathbb{E}_x$.
- Using PPPVSS in randomness beacon protocol, a dealer(as a part of **commit**) requires to do $(n + l)[\mathbb{E}_x + \mathbb{P}_e]$ and $(l - 1)\mathbb{M}_G$ to compute $\{y_i\}_{i=0}^n$. For generating the proof that y_i's are valid encryptions of the secret shares and also y_0 is a commitment of the l secrets, the dealer should do $(n + l)[\mathbb{E}_x + \mathbb{P}_e]$ which internally requires additional $(l - 1)\mathbb{M}_G$. In total, a dealer has to do $2[(n + l)[\mathbb{E}_x + \mathbb{P}_e] + (l - 1)\mathbb{M}_G]$.
  - In **Reveal**, a verifier should do $(n + l)(2\mathbb{E}_x + \mathbb{P}_e)$ and $(l - 1)\mathbb{M}_G$ for each proof. In total, a verifier has to do $(n-1)(n+l)[2\mathbb{E}_x+\mathbb{P}_e]+(n-1)(l-1)\mathbb{M}_\mathbb{G}$.
    - In **Robust case** with t malicious parties not opening the secret polynomials, a verifier should do $l\mathbb{E}_x + (l - 1)\mathbb{M}_G$ to verify each proof, so in total each verifier should do $(n - t - 1)[l\mathbb{E}_x + (l - 1)\mathbb{M}_G]$.
    - In **Honest case** where everyone is honest, a verifier will do $(n - 1)l(\mathbb{E}_x + \mathbb{M}_G)$.
  - The computational complexity of each re-constructor in **Recovery** phase is exactly same as in the case of ALBATROSS.

4.1.1 Computational Cost analysis

The dealer has to do a bit more work in the case of our protocol in contrast to ALBATROSS

4.2 Communication Complexity

Protocol	**Commit** *(by Dealer)*	**Reveal** *(by Dealer)*	**Recovery** *(by shareholder)*
ALBATROSS	$nG + (t+l)\mathbb{Z}_q$	$(t+l)\mathbb{Z}_q$	$1G + 1\mathbb{Z}_q + 1R_o$
with PPPVSS	$(n+1)G + (t+l)\mathbb{Z}_q$	$l\mathbb{Z}_q$	$1G + 1\mathbb{Z}_q + 1R_o$

Table 4.2: Communication cost of dealer and (each) shareholder, R_o being the random oracle, G =group of order q and $\mathbb{Z}_q$ = modular group of order q, where q is a large prime

See table 4.2 for an overview.

- In ALBATROSS, a dealer (as a part of **commit**) should send n group elements as commitments, $t+l$ elements in $\mathbb{Z}/q\mathbb{Z}$ that defines the polynomial used in the ZKP and 1 extra element in $\mathbb{Z}/q\mathbb{Z}$ from RO.
  - In **Reveal**, an honest dealer would broadcast $t+l$ coefficients in $\mathbb{Z}/q\mathbb{Z}$ concerning the secret polynomial.
  - If some party has not revealed their polynomial, then in **Recovery** phase a re-constructor using PVSS reconstruction protocol should broadcast 1 element in group which is being the decrypted secret, for the proof of correct decryption, they have to broadcast 3 more group elements along with a polynomial which requires $t+l$ coefficients in $\mathbb{Z}/q\mathbb{Z}$ and 1 group element from RO.
- Using PPPVSS in randomness beacon protocol, a dealer (as a part of **commit**) should send $n+1$ group elements as commitments, $t+l$ elements in $\mathbb{Z}/q\mathbb{Z}$ that defines the polynomial used in the ZKP and 1 extra element in $\mathbb{Z}/q\mathbb{Z}$ from RO.
  - In **Reveal**, an honest dealer would broadcast l elements in $\mathbb{Z}_q$ concerning the exponents to construct the secret.
  - If some part has not revealed their secrets, then the communication cost of each re-constructor is exactly same as in the case of ALBATROSS.

4.2.1 Communication Cost analysis

Randomness Beacon using PPPVSS (cont.)

3. **Recovery:** Let $\mathcal{C}_a$ be the set containing at most t malicious shareholders(as Dealers) who did not open the exponents corresponding to their l secrets, $\{h^{s_i^k}\}_{i=0}^{-(l-1)}$ for each $P_k \in \mathcal{C}_a$, in *Reveal* phase.

   - Every shareholder P_j should decrypt the secret share of each malicious shareholder(Dealer) in $\mathcal{C}_a$, and give a DLEQ NIZK PoK which asserts that the decryption is performed correctly, $h^{s_j^k}$ and NIZK PoK for $(g, h^{s_j^k}, pk_j, y_j^k) \in L$ for each $P_k \in \mathcal{C}_a$, i.e., each shareholder should perform the *pessimistic* reconstruction phase of PPPVSS for every shareholder(Dealer) who has not revealed the exponents corresponding to their secrets.

4 **Output:** Let T be the $(n-t) \times l$ matrix with rows indexed by the shareholders in $\mathcal{C}$ and where the row corresponding to $P_a \in \mathcal{C}$ is $(h^{s_0^a}, ..., h^{s^a_{-(l-1)}})$.

   - Each computes the $l \times l$-matrix $R = M \circ T$ by applying FFTE to each column $T^{(j)}$ of T, resulting in column $R^{(j)}$ of R (since $R^{(j)} = M \circ T^{(j)}$ and M is Vandermonde) for $j \in [0, l-1]$.
   - Shareholders output the l^2 elements of R as final randomness.

4' **Alternative Output:** if every party in $\mathcal{C}$ has opened her secrets correctly in step *Reveal*, then:

   - Shareholders compute $R = M \circ T$ in the following way: Let S be the $(n-t) \times l$ matrix with rows indexed by the shareholders in $\mathcal{C}$ and where the row corresponding to $P_a \in \mathcal{C}$ is $(s_0^a, ..., s^a_{-(l-1)})$. Then each party computes $U = M \circ S \in \mathbb{Z}_q^{l \times l}$ (using the standard FFT in $\mathbb{Z}_q$ to compute each column) and $R = h^U$.
   - Shareholders output the l^2 elements of R as final randomness.

FIGURE 4.2: Recovery and Output phase of the Randomness Beacon using PPPVSS

Chapter 5

Conclusion

The final chapter contains the overall conclusion. It also contains suggestions for future work and industrial applications.

Appendices

Appendix A

The First Appendix

Appendices hold useful data which is not essential to understand the work done in the master's thesis. An example is a (program) source. An appendix can also have sections as well as figures and references[1].

A.1 More Lorem

Quisque facilisis auctor sapien. Pellentesque gravida hendrerit lectus. Mauris rutrum sodales sapien. Fusce hendrerit sem vel lorem. Integer pellentesque massa vel augue. Integer elit tortor, feugiat quis, sagittis et, ornare non, lacus. Vestibulum posuere pellentesque eros. Quisque venenatis ipsum dictum nulla. Aliquam quis quam non metus eleifend interdum. Nam eget sapien ac mauris malesuada adipiscing. Etiam eleifend neque sed quam. Nulla facilisi. Proin a ligula. Sed id dui eu nibh egestas tincidunt. Suspendisse arcu.

A.1.1 Lorem 15–17

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi.

In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam congue neque id dolor.

A.1.2 Lorem 18–19

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

Nulla non mauris vitae wisi posuere convallis. Sed eu nulla nec eros scelerisque pharetra. Nullam varius. Etiam dignissim elementum metus. Vestibulum faucibus, metus sit amet mattis rhoncus, sapien dui laoreet odio, nec ultricies nibh augue a enim. Fusce in ligula. Quisque at magna et nulla commodo consequat. Proin accumsan imperdiet sem. Nunc porta. Donec feugiat mi at justo. Phasellus facilisis ipsum quis ante. In ac elit eget ipsum pharetra faucibus. Maecenas viverra nulla in massa.

A.2 Lorem 51

Maecenas dui. Aliquam volutpat auctor lorem. Cras placerat est vitae lectus. Curabitur massa lectus, rutrum euismod, dignissim ut, dapibus a, odio. Ut eros erat, vulputate ut, interdum non, porta eu, erat. Cras fermentum, felis in porta congue, velit leo facilisis odio, vitae consectetuer lorem quam vitae orci. Sed ultrices, pede eu placerat auctor, ante ligula rutrum tellus, vel posuere nibh lacus nec nibh. Maecenas laoreet dolor at enim. Donec molestie dolor nec metus. Vestibulum libero. Sed quis erat. Sed tristique. Duis pede leo, fermentum quis, consectetuer eget, vulputate sit amet, erat.

Appendix B

The Last Appendix

Appendices are numbered with letters, but the sections and subsections use arabic numerals, as can be seen below.

B.1 Lorem 20-24

Nulla ac nisl. Nullam urna nulla, ullamcorper in, interdum sit amet, gravida ut, risus. Aenean ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim. Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetuer quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

Maecenas non massa. Vestibulum pharetra nulla at lorem. Duis quis quam id lacus dapibus interdum. Nulla lorem. Donec ut ante quis dolor bibendum condimentum. Etiam egestas tortor vitae lacus. Praesent cursus. Mauris bibendum pede at elit. Morbi et felis a lectus interdum facilisis. Sed suscipit gravida turpis. Nulla at lectus. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Praesent nonummy luctus nibh. Proin turpis nunc, congue eu, egestas ut, fringilla at, tellus. In hac habitasse platea dictumst.

B.2 Lorem 25-27

Vivamus eu tellus sed tellus consequat suscipit. Nam orci orci, malesuada id, gravida nec, ultricies vitae, erat. Donec risus turpis, luctus sit amet, interdum quis, porta sed, ipsum. Suspendisse condimentum, tortor at egestas posuere, neque metus tempor orci, et tincidunt urna nunc a purus. Sed facilisis blandit tellus. Nunc risus sem, suscipit nec, eleifend quis, cursus quis, libero. Curabitur et dolor. Sed vitae sem. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Maecenas ante. Duis ullamcorper enim. Donec tristique enim eu leo. Nullam molestie elit eu dolor. Nullam bibendum, turpis vitae tristique gravida, quam sapien tempor lectus, quis pretium tellus purus ac quam. Nulla facilisi.

Duis aliquet dui in est. Donec eget est. Nunc lectus odio, varius at, fermentum in, accumsan non, enim. Aliquam erat volutpat. Proin sit amet nulla ut eros consectetuer cursus. Phasellus dapibus aliquam justo. Nunc laoreet. Donec consequat placerat magna. Duis pretium tincidunt justo. Sed sollicitudin vestibulum quam. Nam quis ligula. Vivamus at metus. Etiam imperdiet imperdiet pede. Aenean turpis. Fusce augue velit, scelerisque sollicitudin, dictum vitae, tempor et, pede. Donec wisi sapien, feugiat in, fermentum ut, sollicitudin adipiscing, metus.

Donec vel nibh ut felis consectetuer laoreet. Donec pede. Sed id quam id wisi laoreet suscipit. Nulla lectus dolor, aliquam ac, fringilla eget, mollis ut, orci. In pellentesque justo in ligula. Maecenas turpis. Donec eleifend leo at felis tincidunt consequat. Aenean turpis metus, malesuada sed, condimentum sit amet, auctor a, wisi. Pellentesque sapien elit, bibendum ac, posuere et, congue eu, felis. Vestibulum mattis libero quis metus scelerisque ultrices. Sed purus.

Bibliography

[1] D. Adams. *The Hitchhiker's Guide to the Galaxy.* Del Rey (reprint), 1995. ISBN-13: 978-0345391803.

[2] K. Baghery. π: A unified framework for computational verifiable secret sharing. Cryptology ePrint Archive, Paper 2023/1669, 2023.

[3] K. Baghery, N. Knapen, G. Nicolas, and M. Rahimi. Pre-constructed publicly verifiable secret sharing and applications. Cryptology ePrint Archive, Paper 2025/576, 2025.

[4] G. R. Blakley and C. Meadows. Security of ramp schemes. In *Advances in Cryptology, Proceedings of CRYPTO '84, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 242–268. Springer, 1984.

[5] I. Cascudo and B. David. SCRAPE: Scalable randomness attested by public entities. Cryptology ePrint Archive, Paper 2017/216, 2017.

[6] I. Cascudo and B. David. ALBATROSS: publicly AttestabLe BATched randomness based on secret sharing. Cryptology ePrint Archive, Paper 2020/644, 2020.

[7] D. Chaum and T. P. Pedersen. Wallet databases with observers. In E. F. Brickell, editor, *Advances in Cryptology — CRYPTO' 92*, pages 89–105, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[8] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *SFCS '85*, pages 383–395. IEEE, 1985. DBLP's bibliographic metadata records provided through http://dblp.org/search/publ/api are distributed under a Creative Commons CC0 1.0 Universal Public Domain Dedication. Although the bibliographic metadata records are provided consistent with CC0 1.0 Dedication, the content described by the metadata records is not. Content may be subject to copyright, rights of privacy, rights of publicity and other restrictions.; 26th Annual Symposium on Foundations of Computer Science ; Conference date: 21-10-1985 Through 23-10-1985.

[9] P. Feldman. A practical scheme for non-interactive verifiable secret sharing. In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 427–437. IEEE Computer Society, 1987.

[10] A. Fiat and A. Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology — CRYPTO' 86*, pages 186–194, Berlin, Heidelberg, 1987. Springer Berlin Heidelberg.

[11] M. Franklin and M. Yung. Communication complexity of secure computation (extended abstract). In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '92, page 699–710, New York, NY, USA, 1992. Association for Computing Machinery.

[12] C.-P. Schnorr. Efficient identification and signatures for smart cards. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1989.

[13] L. Schoenmakers. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In M. Wiener, editor, *Advances in Cryptology - CRYPTO'99 (Proceedings 19th Annual International Cryptology Conference, Santa Barbara CA, USA, August 15-19, 1999)*, Lecture Notes in Computer Science, pages 148–164, Germany, 1999. Springer.

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