SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Informatics

Formalizing the Kate-Zaverucha-Goldberg Polynomial Commitment Scheme

Tobias Rothmann

SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY —
INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis in Informatics

Formalizing the Kate-Zaverucha-Goldberg Polynomial Commitment Scheme

Eine Formalisierung des Kate-Zaverucha-Goldberg Polynomiellen Commitment Verfahrens

Author:	Tobias Rothmann
Supervisor:	Prof. Tobias Nipkow
Advisor:	Katharina Kreuzer
Submission Date:	April 15, 2024

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, April 15, 2024   Tobias Rothmann

Acknowledgments

Abstract

Contents

1 Introduction

1.1 Section

Citation test [Lam94] [Lan02].

Acronyms must be added in `main.tex` and are referenced using macros. The first occurrence is automatically replaced with the long version of the acronym, while all subsequent usages use the abbreviation.

E.g. `\ac{TUM}, \ac{TUM}` $\Rightarrow$ Technical University of Munich (TUM), TUM

For more details, see the documentation of the `acronym` package[1].

1.1.1 Subsection

See Table 1.1, Figure 1.1, Figure 1.2, Figure 1.3.

Table 1.1: An example for a simple table.

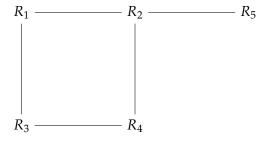| A	B	C	D
1	2	1	2
2	3	2	3

Figure 1.1: An example for a simple drawing.

---

[1] `https://ctan.org/pkg/acronym`

Figure 1.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 1.3: An example for a source code listing.

2 related work

3 Prelimiaries

In this section, we introduce the notation used throughout the paper, and capture the most important preliminaries in definitions. We start with the mathematical notation and concepts used in this paper.

3.1 Mathematical Prelimiaries

We let p and q denote prime numbers if not explicitly stated otherwise. Groups are written in a multiplicate manner with the $\cdot$ operator and the abbreviation ab for $a \cdot b$. Furthermore, we use the notation $\mathbb{Z}_p$ for a finite field of prime order p (note that the integers modulo p are isomorph to any finite field of prime order p [Lan02]).

Definition 3.1.1 (cyclic group). Let $\mathcal{G}$ be a group of prime order p. We call a group cyclic iff: $\exists g \in \mathcal{G}. \forall e \in \mathcal{G}. \exists n \in \mathbb{N}. e = g^n$, which is equivalent to $\mathcal{G} = \{1, g, g^2, ..., g^{p-1}\}$ [Lan02].

From now on we write **g** for an arbitrary but fixed generator of a respective cyclic group.

Definition 3.1.2 (pairings). Let $\mathcal{G}$ and $\mathcal{H}$ be two groups of prime order p. A pairing is a function: $\mathcal{G} \times \mathcal{G} \to \mathcal{H}$, with the following two properties:

- **Bilineraity:** $\forall g, h \in \mathcal{G}. \forall a, b \in \mathbb{Z}_p. e(g^a, h^b) = e(g, h)^{ab}$

- **Non-degeneracy:** $\neg(\forall g, h \in \mathcal{G}. e(g, h) = 1)$

[KZG10]

Now that we have introduced the mathematical preliminaries we will tend to the cryptographic preliminaries.

3.2 Cryptographic Prelimiaries

In this section, we will introduce the security notions that we use in this paper and the concepts behind them.

We start with the definition of a negligable and a poly-bounded function from which we will define our adversarial model, against which we will prove security in this paper.

Definition 3.2.1. Let f be a function: $\mathbb{Z}_{\geq 0} \to \mathbb{R}$. We call f negligable iff: $\forall c \in \mathbb{R}_{>0}. \exists n_0. \forall n \geq n_0. |f(n)| < 1/n^c$
   [BS23]

Boneh and Shoup state "Intuitively, a negligible function $f : \mathbb{Z}_{\geq 0} \to \mathbb{R}$ is one that not only tends to zero as $n \to \infty$, but does so faster than the inverse of any polynomial." [BS23]

Definition 3.2.2. Let f be a function: $\mathbb{Z}_{\geq 0} \to \mathbb{R}$. We call f poly-bounded iff: $\exists c, d \in \mathbb{R}_{>0}. \forall n \in \mathbb{N}_0. |f(n)| \leq n^c + d$
   [BS23]

Note, we will define (probabilistic) algorithms for some security parameter κ and bound their performance using the notion of negligibility and poly-boundedness with respect to κ.

We capture the security of our cryptograhpic-system in games against a (efficient) Adversary. Typically, the Adversary has to break a security property in those games (e.g. decrypt a cyphertext). However, before we formally define games, we define what an Adversary is.

Definition 3.2.3 (efficient Adversary). An Adversary is a probabilistic algorithm, that takes a security parameter κ as its first argument. We call an Adversary efficient if its running time is poly-bounded in κ except for negligible probability (with respect to κ) [BS23].

Besides this definition, we will use a stronger definition of Adversaries, namely that of the Adversary in the Algebraic Group Model (AGM) [FKL17].

Definition 3.2.4 (AGM Adversary). Let $\mathbb{Z}_p$ be a finite field of prime order p and $\mathbb{G}$ a cyclic group of prime order p. An Adversary in the AGM is an Adversary as in definition 3.2.3, that furthermore outputs a vector $\vec{z} \in \mathbb{Z}_p^t$ for every element e from $\mathbb{G}$ in its output, such that $e = \prod_1^t g_i^{z_i}$, where $g \in \mathbb{G}^t$ is the vector of all elements of $\mathbb{G}$ that the Adversary has seen so far [FKL17].
   The efficiency definition is analogue to definition 3.2.3

Now that we have defined Adversary models, we define games.

Definition 3.2.5 (games). Games are probabilistic algorithms with access to an Adversary and output a boolean value [BS23]. Formally we write games as a sequence of functions and Adversary calls [BS23].

Notatioinwise we write ' ← ' followed by a set for uniform sampling from that set, ' ← ' followed by a probability mass function (e.g. an Adversary result) to sample from that function space, and ' = ' for an assignment of a deterministic value. Moreover, we write ' : ' followed by a condition to assure that the condition has to hold at this point. To give an example, think of the following game as "sampling a uniformly random a from $\mathbb{Z}_p$, get the probabilistic result from $\mathcal{A}$ as b, computing c as F applied to a and b, and assert that P holds for c":

Example Game

$$a \leftarrow \mathbb{Z}_p,$$
$$b \leftarrow \mathcal{A},$$
$$c = F\ a\ b,$$
$$: P\ c$$

Abbreviations

TUM Technical University of Munich

List of Figures

List of Tables

Bibliography

[BS23]   D. Boneh and V. Shoup. *A Graduate Course in Applied Cryptography*. `http://toc.cryptobook.us/book.pdf`. 2023.

[FKL17]   G. Fuchsbauer, E. Kiltz, and J. Loss. *The Algebraic Group Model and its Applications*. Cryptology ePrint Archive, Paper 2017/620. `https://eprint.iacr.org/2017/620`. 2017.

[KZG10]   A. Kate, G. M. Zaverucha, and I. Goldberg. "Constant-Size Commitments to Polynomials and Their Applications." In: *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security*. Vol. 6477. Lecture Notes in Computer Science. Springer, 2010, pp. 177–194. DOI: `10.1007/978-3-642-17373-8_11`.

[Lam94]   L. Lamport. *LaTeX : A Documentation Preparation System User's Guide and Reference Manual*. Addison-Wesley Professional, 1994.

[Lan02]   S. Lang. *Algebra*. Vol. 3. Springer New York, NY, 2002.