UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Fakultät für Elektrotechnik, Informatik und Mathematik
Arbeitsgruppe Codes und Kryptographie

Title

Bachelor's Thesis

in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science

by
Niklas Isermann

submitted to:
Prof. Dr. Johannes Blömer
and
???

Paderborn, July 11, 2022

Eidesstattliche Versicherung

Nachname: _____ Vorname: _____

Matrikelnr.: _____ Studiengang: _____

☐ Bachelorarbeit ☐ Masterarbeit

Titel der Arbeit: Title

☐ Die elektronische Fassung ist der Abschlussarbeit beigefügt.

☐ Die elektronische Fassung sende ich an die/den erste/n Prüfenden bzw. habe ich an die/den erste/n Prüfenden gesendet.

Ich versichere hiermit an Eides statt, dass ich die vorliegende Abschlussarbeit (Ausarbeitung inkl. Tabellen, Zeichnungen, etc.) selbstständig und ohne unzulässige fremde Hilfe erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht. Die Abschlussarbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen. Die elektronische Fassung entspricht der gedruckten und gebundenen Fassung.

Belehrung

Wer vorsätzlich gegen eine die Täuschung über Prüfungsleistungen betreffende Regelung einer Hochschulprüfungsordnung verstößt, handelt ordnungswidrig. Die Ordnungswidrigkeit kann mit einer Geldbuße von bis zu 50.000,00 € geahndet werden. Zuständige Verwaltungsbehörde für die Verfolgung und Ahndung von Ordnungswidrigkeiten ist die Vizepräsidentin / der Vizepräsident für Wirtschafts- und Personalverwaltung der Universität Paderborn. Im Falle eines mehrfachen oder sonstigen schwerwiegenden Täuschungsversuches kann der Prüfling zudem exmatrikuliert werden. (§ 63 Abs. 5 Hochschulgesetz NRW in der aktuellen Fassung).

Die Universität Paderborn wird ggf. eine elektronische Überprüfung der Abschlussarbeit durchführen, um eine Täuschung festzustellen.

Ich habe die oben genannten Belehrungen gelesen und verstanden und bestätige dieses mit meiner Unterschrift.

Ort: _____ Datum: _____

Unterschrift: _____

Datenschutzhinweis

Die o.g. Daten werden aufgrund der geltenden Prüfungsordnung (Paragraph zur Abschlussarbeit) i.V.m. § 63 Abs. 5 Hochschulgesetz NRW erhoben. Auf Grundlage der übermittelten Daten (Name, Vorname, Matrikelnummer, Studiengang, Art und Thema der Abschlussarbeit) wird bei Plagiaten bzw. Täuschung der/die Prüfende und der Prüfungsausschuss Ihres Studienganges über Konsequenzen gemäß Prüfungsordnung i.V.m. Hochschulgesetz NRW entscheiden. Die Daten werden nach Abschluss des Prüfungsverfahrens gelöscht. Eine Weiterleitung der Daten kann an die/den Prüfende/n und den Prüfungsausschuss erfolgen. Falls der Prüfungsausschuss entscheidet, eine Geldbuße zu verhängen, werden die Daten an die Vizepräsidentin für Wirtschafts- und Personalverwaltung weitergeleitet. Verantwortlich für die Verarbeitung im regulären Verfahren ist der Prüfungsausschuss Ihres Studiengangs der Universität Paderborn, für die Verfolgung und Ahndung der Geldbuße ist die Vizepräsidentin für Wirtschafts- und Personalverwaltung.

Contents

1 Basic definitions and notation

This file contains example content. It is meant to get you started.

You can remove this example content from the thesis by removing the input statement for example.tex from the thesis_main.tex file.

1.1 Basic notation

Throughout this thesis, we will use the following notation:

- $\mathbb{N} := \{1, 2, \dots\}$ denotes the set of natural numbers (excluding zero).
- For a bit string $s = (s_0, \dots, s_{n-1}) \in \{0,1\}^n$ and $0 \leq i \leq j < n$, we write $s[i:j] := (s_i, \dots, s_{j-1})$ to denote substrings. In particular, $s[i:i]$ is the empty string ε and $s[0:n]$ is the complete string s.
- For two vectors $\vec{u}, \vec{v} \in \{0,1\}^n$, with $\vec{u} = (u_1, \dots, u_n), \vec{v} = (v_1, \dots, v_n)$, the expression $\vec{u} \odot \vec{v}$ denotes the Hadamard product. $(\vec{u} \odot \vec{v})_i = u_i \cdot v_i$.

1.2 Gobbling schemes

Don't try to make sense of this. It's just a syntax example with no real connection to anything.

Gobbling schemes are useful whenever the the Hadamard product of two random bit vectors needs to be hidden from polynomially bounded adversaries. They are an important building block for twaddle signatures, which we investigate in this thesis.

This is an example for a todonote. They're super useful!

1.2.1 Syntax definition

Our definition for gobbling schemes is taken from [Eti14] with minor syntactic changes. See Section 1.1 for basic notation.

note the citation

note the reference

Definition 1.1 (Gobbling scheme) *A* gobbling scheme Π *consists of the following three probabilistic polynomial-time algorithms:*

- $pk \leftarrow \mathsf{Setup}(1^\lambda)$ *on input a unary security parameter* λ, Setup *generates a public key* pk.
- $\vec{q} \leftarrow \mathsf{Gobble}(pk, \vec{u}, \vec{v})$ *generates a* gobbled *vector* $\vec{q} \in \{0,1\}^n$ *given a key* pk *and two vectors* $\vec{u}, \vec{v} \in \{0,1\}^n$.

note that *pk* and Setup are macros defined in defs.tex.

- $\vec{z} \leftarrow \mathsf{Ungobble}(pk, \vec{q})$ *given a gobbled vector* $\vec{q} \in \{0,1\}^n$ *outputs an ungobbled vector* $\vec{z} \in \{0,1\}^{2n}$.

Π *is* correct *if there exists a negligible function* μ *such that for all* $\lambda, n \in \mathbb{N}$,

$$1 - \mu(\lambda) \leq \Pr[\vec{z} = \vec{u} \odot \vec{v} \,|k \leftarrow \mathsf{Setup}(1^\lambda); \vec{u}, \vec{v} \leftarrow \{0,1\}^n; \\ \vec{z} \leftarrow \mathsf{Ungobble}(k, \mathsf{Gobble}(k, \vec{u}, \vec{v}))]$$

Intuitively, correctness guarantees ...

1.2.2 Security definition

...

2 Abstract

3 Introduction

A famous problem in the context of MPC is Yao's millionaire's problem. In Yao's millionaire's problem there are two millionaires Alice and Bob. We will call Alice's wealth x and Bob's wealth y. Alice and Bob want to know who of them is has more money. i.e. they want to compute the function F(x,y) := $\begin{cases} Alice\,is\,richer & y \leq x \\ Bob\,is\,richer & y > x \end{cases}$. Yet neither of them is willing to trust the other and tell him how much money he has. Yao's millionaire's problem can be generalised into the general MPC problem. Instead of Bob and Alice, we now consider n parties $p_0, \ldots, p_{n-1}$ and each party i holds an arbitrary input x_i for an arbitrary function F($x_0, \ldots, x_{n-1}$), that all parties have agreed upon. A MPC protocol π is protocol, that allows $p_0, \ldots, p_{n-1}$ to compute F($x_0, \ldots, x_{n-1}$) without revelling any information about $x_0, \ldots, x_{n-1}$. Before we can formalize our security goal of "not revelling $x_0, \ldots, x_{n-1}$.", it is necessary to talk about our adversary and its capability's. An adversary has the ability to corrupt one or more party's. Once a party is corrupted the adversary get full information about every message the party send our receives, this also includes the messages from the time before the party had been corrupted. There are multiple categorizations of adversary's and their capability's. On such categorization is the distinction between passive and active adversary's. A passive adversary can not force a corrupted party to deviate from the protocol in an any way. A active adversary has the power to force a corrupted party to deviate from the protocol in an arbitrary way. So if for example the protocol would at some point require that each party choses an integer between 1 and n uniformly at random. Then a passive adversary would have no choice but to choose the integer between 1 and n uniformly at random. On the contrary an active adversary would be able to force a corrupted party to chose the value 42 or any other value that the adversary considers to be advantageous for him. TODO simulation based security definition

Andrew Yao proposed a solution for Yao's millionaire's problem in 1982 []. It has also been shown that MPC is Turing-complete[]. This means that for any function f that can be computed with a Turing machine. There exists a MPC protocol π that can compute f.

4 related work

4.1 conclave

2019, end-to-end security by mix max, "SMCQL most similar existing system"

4.2 aby3

In [MRR20] .. 3 Party setting , honest majority passiv adversary O(n) constant round, inner left ,full join, set union,set minus, single table operation like where or aggregate.prototype LAN VS WAN

4.3 smcql

Bibliography

[Eti14] Y. Eti. On the Importance of Correct Stirring. *International Journal of Cookie Theory*, 13(1):1–247, 2014.

[MRR20] Payman Mohassel, Peter Rindal, and Mike Rosulek. *Fast Database Joins and PSI for Secret Shared Data*, page 1271–1287. Association for Computing Machinery, New York, NY, USA, 2020.