Chair for Embedded Systems
Universität Augsburg

Master's Thesis

Implementation of an IoT based Electronic Voting Machine

Gabriel Cmiel

Gutachter/Examiner:	Prof. Dr. Vorname Nachname
Zweitgutachter/Second examiner:	Prof. Dr. Vorname Nachname
Betreuer/Supervisor:	Prof. Dr. Sebastian Altmeyer
Date:	17th October 2024

written at
Chair for Embedded Systems
Prof. Dr. Sebastian Altmeyer
Institute of Computer Science
University of Augsburg
D–86135 Augsburg, Germany
https://www.Informatik.uni-augsburg.de

Contents

Abstract v

List of Abbreviations vii

1. Introduction 1
1.1. Ziele dieser Arbeit 1
1.2. Überblick 1

2. Background 3
2.1. Cryptography 3
2.2. ElectionGuard 5
2.2.1. Election Verification 6

3. Hauptteil/Main Part 7
3.1. Hardware 7
3.1.1. Breadboard Prototype 7
3.1.2. Cryptographic Hardware accelerators 8
3.2. Implementation/Software Architecture 8
3.2.1. Model 8
3.2.2. View 9
3.2.3. Adapter 9
3.3. Verification 9
3.4. Unterkapitel 10
3.4.1. Dritte Gliederungsebene 10

4. Conclusions 13

List of figures ix

List of tables xi

Bibliography xiii

A. Appendix xv

Abstract

Eine kurze Zusammenfassung der Ausarbeitung.

List of Abbreviations

EAC Election Assitance Commission

E2E End-to-end

VVSG Voluntary Voting System Guidelines

ZK zero-knowledge

PRG Pseudo-random generator

NIZK Non-interactive zero-knowledgte proofs

KDF Key derivation function

MAC Message authentication code

HMAC Hash Message Authentication code

NIST National Institute of Standards and Technology

Glossary

E2E End-to-end. vii

EAC Election Assistance Commission. vii

VVSG Voluntary Voting System Guidelines. vii

1. Introduction

In der Einleitung wird die Arbeit motiviert und die Relevanz dieser herausgearbeitet.

1.1. Ziele dieser Arbeit

Die Ziele der Arbeit werden hier erläutert.

1.2. Überblick

Der Autor führt einen potentiellen Leser durch die Arbeit und beschreibt kurz, was den Leser in den folgenden Kapiteln erwartet.

2. Background

2.1. Cryptography

Cryptography is the science of securing information through encryption. Encryption or ciphering refers to the process of making a message incomprehensible [Ert07, p. 18] The security of all cryptographic methods is essentially based on the difficulty of guessing a secret key or obtaining it by other means. It is possible to guess a key, even if the probability becomes very small as the length of the key increases. It must be pointed out that there is no absolute security in cryptography [Ert07, p. 25].

Practically all cryptographic methods have the task of ensuring one of the following security properties are met [Ert07, p. 18].

- **Confidentiality** The aim of confidentiality is to make it impossible or difficult for unathorized persons to read a message [Ert07, p. 18].

- **Authenticity** Proof of identity of the message sender to the recipient, i.e. the recipient can be sure that the message does not originate from another (unauthorized) sender [Ert07, p. 18]

- **Integrity** The message must not be altered (by unauthorized persons) during transmission. It retains its integrity [Ert07, p. 18].

- **Non-repudiation** The sender cannot later deny having sent a message [Ert07, p. 18].

Cryptographic algorithms are mathematical equations, i.e. mathematical functions for encryption and decryption [Ert07, p. 19]. A cryptographic algorithm for encryption can be used in a variety of ways in different applications. To ensure that an application always runs in the same and correct correct way, cryptographic protocols are defined. In contrast to the cryptographic algorithms, the protocols are procedures for controlling the flow of transactions for certain applications. [Ert07, p. 22].

The idea of combining cryptographic methods with voting systems is not new. In 1981, David Chaum published a cryptographic technique based on public key cryptography that hides who a participant communicates with, aswell as the content of the communication. The untracable mail system requires messages to pass through a cascade of mixes (also known as a Mix Network) **chaum**. Chaum proposes that the techniques can be used in elections in which an individual can correspond with a record-keeping organisation or an interested party under a unique pseudonym. The unique pseudonym has to appear in a roster of accetable clients. A interested party or record keeping organisation can verify that the message was sent by a registered voter. The record-keeping organisation or the interested party can also verify that the message was not altered during transmission. **chaum**.

In this use case, the properties of Confidentiality, Authenticity, Integrity and Non-repudiation are ensured. However, to be worthy of public trust, an election process must give voters and observers compelling evidence that the election was conducted properly without breaking voters confidentiality. The problem of public trust is further exacerbated to now having to trust election software and hardware, in addition to election officials, and proceedurs.

In 2021, the U.S. Election Assitance Commission (EAC) adopted the Voluntary Voting System Guidelines (VVSG) 2.0. [Com21]. The VVSG is intended for designers and manufacturers of voting systems. Currently, the VVSG is titled as "Recommendations to the EAC" because it's not yet the final version that voting system manufacturers will follow. **https://www.nist.gov/itl/voting/vvsg-introduction**. The VVSG 2.0 currently states only two methods for achieving software independence. The first through the use of independent voter-verifiable paper records, and the second through cryptographic E2E verifiable voting systems. The VVSG 2.0 states that a voting system need to be software independent through the use of independent voter-verifiable paper records **https://www.eac.gov/sites/default/files/TestingCertificatio**
However, due to the lack of E2E verifiable voting systems available within the current market, there are no verified E2E cryptographic protocols.**https://www.eac.gov/sites/default/**
The U.S. Election Assistance Commission, in collaboration with the National Institute of Standards and Technology initialised an Call for proposals to solicit, evaluate, and approve protocols used in E2E cryptographically verifiable voting systems. **https://www.eac.gov/voting-equipment/end-end-e2e-protocol-evaluation-process**.

Submitted protocols must support the following properties

- Cast as Intended: Allow voters to confirm the voting system correctly interpreted their ballot selections while in the polling place via a receipt and provide evidence such that if there is an error or flaw in the interpretation of the voters' selections.

- Recorded as Cast: Allow voters to verify that their cast ballots were accurately

recorded by the voting system and included in the public records of encoded ballots.

- Tallied as Recorded: Provide a publicly verifiable tabulation process from the public records of encoded ballots.

2.2. ElectionGuard

One of the first pilots to see how E2E verifiable elections works in a real election took place in a district of Preston, Idaho, United States, on November 8, 2022. The Verity scanner from Hart InterCivic was used in this pilot, which was integrated with Microsoft's ElectionGuard. [EAC Report]. ElectionGuard is a toolkit that encapsulates cryptographic functionality and provides simple interfaces that can be used without cryptographic expertise. The principal innovation of ElectionGuard is the seperation of the cryptographic tools from the core mechanics and user interfaces of voting systems. In it's preferred deployment, ElectionGuard doesn't replace the existing vote counting infrastructure but instead runs alongside and produces its own independently-verifiable tallies [Ben+24, pp. 1–2]. The cryptographic design is largely inspired by the cryptographic voting protocol by Cohen (now Benaloh) and Fischer in 1985 and the voting protocol by Cramer, Gennaro and Schoenmakers in 1997 [Ben+24, p. 5].

———————————————- The philosophy of ElectionGuard has been to cover the majority of voting scenarios with an approach that is as simple as possible to understand and verify. It can be used with Precinct Ballot Scanners, Electronic Ballot Markers, Internet Voting, Risk-Limiting Audits and even vote by mail and many more.

In all applications, an election using ElectionGuard begins with a key-generation ceremony in which an election administrator works with guardians to form election keys. Later, usually at the conclusion, the administrator will again work with guardians to produce verifiable tallies. What happens in between, however, can vary widely. [ElectionGuard: a Cryptographic Toolkit to Enable Verifiable Elections].

This thesis focuses on the implementation of the Key Generation Ceremony and the Guardians using the ESP32 microcontroller. The ESP32 is a low-cost, low-power system on a chip microcontroller. It is widely used in IoT applications. [source]

———————————————-

2.2.1. Election Verification

ElectionGuard support the central aspects which a VVSG 2.0 compliant voting system must support Cast as Intended, Recorded as Cast and Tallied as Recorded. [Ben+24, p. 17]. Key element supporting cast-as-intended and recorded-as-cast verifiability is through confirmation codes. Ballots can be ce challenged or cast and both are included in the election record. Voters can check if the expected confirmation code appears in the election records and for the challenged ballots, that it shows the correct selections [Ben+24, p. 18]. Tallied-as-cast verifiability is supported through the inclusion of all ballots and decryption proofs in the election record. Any voter can verify that the ballot is accurately incorporated in the tally, and the decryption proofs demonstrate the validity of the announced tally [Ben+24, p. 18]. To confirm the election's integrity, independent verification software can be used at any time after the completion of an election. [Ben+24, p. 6]. Some, may choose even to write their own verifiers.

3. Hauptteil/Main Part

3.1. Hardware

3.1.1. Breadboard Prototype

Our prototype uses the NodeMCU ESP32 development board by Joy-IT. The board is equipped with the ESP32-WROOM-32 module. ESP32-WROOM-32 is a powerful, generic Wi-Fi+BT+BLE MCU module.

At the core of the module is the ESP32-D0WDQ6 chip [Sys23, p. 6]. This chip, and therefore this module. is not recommended for new designs anymore due to chip revisions. The ESP32-D0WDQ6 is based on chip revision v.1.0 or v1.1. [24a, p. 11]. The ESP32-WROOM-32 module could therefore be replaced by the newer ESP32-WROOM-32E module. The ESP32-WROOM-32E module uses a ESP32-D0WD-V3 or ESP32-D0WDR2-V3 chip which are based on chip revision v3.0 or v3.1 which fixes some Hardware bugs [Sys24c, p. 1], [24a, p. 11], [Sys24b, pp. 3–4]. Compared to the ESP32-WROOM-32 module, the ESP32-WROOM-32E module has versions that provide additional 2MB PSRAM, support higher operating ambient temperatures and veWrsions with higher integrated SPI flash sizes (8MB or 16MB) [Sys24c, p. 2], [Sys23, pp. 6–7].

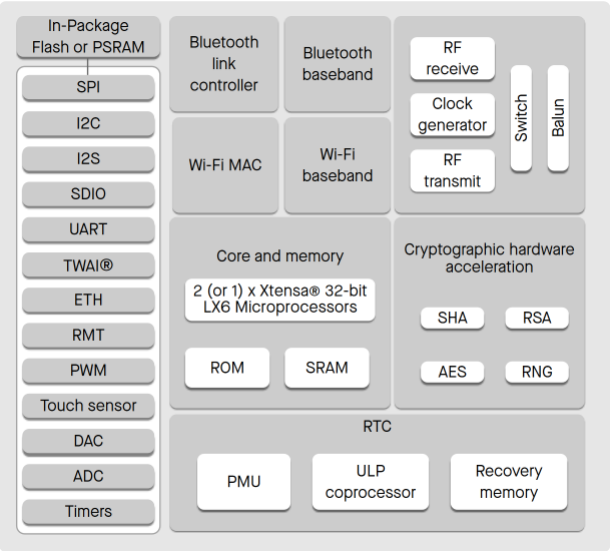

Figure 3.1.: ESP32 Functional Block Diagram

3.1.2. Cryptographic Hardware accelerators

3.2. Implementation/Software Architecture

3.2.1. Model

Difference with other Implementations

Since ElectionGuards original specification in 2019, there have been several implementations of ElectionGuard that have been used in various applications . The current roadmap of ElectionGuard targets a c++ implementation of the ElectionGuard 2.0 specification. [https://www.electionguard.vote/overview/Roadmap/]. Earlier implementations include a Python reference implementation of the ElectionGuard 1.0 specification and an encryption engine in C++ with a C wrapper.

ESP32 development support development of applications in C, C++ and Micropython. [source]. Initialy, we could try to port the encryption engine written in c++ over to ESP32. This would allow us to use the existing codebase and focus on the integration of the encryption component with the hardware. The encryption engine would be responsible for generating an elgamal keypair and the subsequent exchange of cryptographic proofs and cryptographic keys.

The modular exponentiation at the heart of most ElectionGuard operations imposes the highest computaitional cost among all computations and is the limiting factor in any performance analysis. Using fast libraries for modular arithmetic is crucial to achieve good performance so that the latency due to Key generation and ZK proof generation doesn not impact usability. [source]

The c++ implementation uses Microsoft's HACL* - a performant C implementation of a wide variety of cryptographic primitives which have been formally verified for correctness [source.]. For performance reasons the implementation of the c++ encryption engine uses pre-computed tables to make encryption substantially faster. This is possible because most exponentiations in ElectionGuard have fixed base, either the generator g or the election public key K. The pre-computed tables contain certain powers of these bases. The Python reference implementation uses a more straightforwar approach by using GnuMP. [source].

The ESP32 is equipped with hardware accelerators of general algorithms such as SHA and RSA and it also supports independent arithmetic, such as Big Integer Multiplication and Big Integer Modular Multiplication. [esp tech reference,4.1.19]. The hardware accelerators greatly improve operation speed and reduce software complexity.

Performance

- With/Without HW Accerleration - Single/Dual Core

3.2.2. View

3.2.3. Adapter

3.3. Verification

Bachelor- und Masterarbeiten können sowohl in deutsch als auch in englisch geschrieben werden. Die sprachliche Ausarbeitung wird bewertet, was bei der Wahl der Sprache berücksichtigt werden sollte. Im folgenden werden ein paar Hinweise zur Ausarbeitung mit LaTeX gegeben.

	LCD	Board	Description
1	VCC	3.3V	
2	GND	GND	
3	GND	GND	
4	NC		
5	NC		
6	NC		
7	CLK	D14	SPI-CLK
8	SDA	D13	SPI-MOSI
9	RS	D34	Any GPIO PIN
10	RST	D35	Any GPIO PIN
11	CS	D15	SPI-SS

Table 3.1.: PINOUT LCD

3.4. Unterkapitel

3.4.1. Dritte Gliederungsebene

Falls in einem Kapitel mehrere Gliederungsebenen verwendet werden sollte darauf geachtet werden, dass mindestens drei Punkte pro ebene existieren.

1	2	3
4	5	6

Table 3.2.: Beispieltabelle

Hier wird die Beispieltabelle 3.2 referenziert.

Figure 3.2.: Bildunterschrift mit Quellenangabe [LCD]

Hier wird die Beispielbild 3.2 referenziert.

$$\sum_{x=0}^{10} x = 55 \tag{3.1}$$

Hier wird die Beispielformel 3.1 referenziert.

kursiv, **fett**, unterstrichen

Abkürzungen müssen im Abkürzungsverzeichnis angelegt werden. Erste Verwendung einer **ABK!** (**ABK!**) jede weitere Verwendung der **ABK!**.

4. Conclusions

Im Schlusskapitel wird die Arbeit und ihre Ergebnisse zusammengefasst sowie ein Ausblick gegeben.

List of Figures

3.1. ESP32 Functional Block Diagram 8
3.2. Bildunterschrift mit Quellenangabe 10

List of Tables

3.1. PINOUT LCD 10
3.2. Beispieltabelle 10

Bibliography

[23] *End-to-End Verifiability in Real-World Elections*. Tech. rep. Microsoft, Jan. 2023.

[24a] *ESP32 Series Datasheet*. v4.5. Espressif Systems. 2024. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[24b] *ESP32 Technical Reference Manual*. v5.2. Espressif Systems. Aug. 2024. URL: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf.

[Ben+24] Josh Benaloh et al. *ElectionGuard: a Cryptographic Toolkit to Enable Verifiable Elections*. Cryptology ePrint Archive, Paper 2024/955. https://eprint.iacr.org/2024/955. 2024. URL: https://eprint.iacr.org/2024/955.

[BN] Josh Benaloh and Michael Naehrig. *ElectionGuard Specification*. v1.0. Microsoft Research.

[Com21] United States Election Assistance Commission. *U.S. Election Assistance Commission Adopts New Voluntary Voting System Guidelines 2.0*. Accessed on 2024-01-10. Feb. 2021. URL: https://www.eac.gov/news/2021/02/10/us-election-assistance-commission-adopts-new-voluntary-voting-system-guidelines-20.

[Ert07] Wolfgang Ertel. *Angewandte Kryptographie*. 3., aktualisierte Auflage. München: Carl Hanser Verlag, 2007. ISBN: 978-3-446-41195-1.

[eta19] Andrew Banks et.al. *MQTT Version 5.0*. Rev1.0. LCDWiki. Mar. 2019.

[Gmb20] SIMAC Electronics GmbH. *SBC-Button 2 Datasheet*. SIMAC Electronics GmbH. Pascalstr. 8, 47506 Neukirchen-Vluyn, Oct. 2020.

[Joy18] Joy-it. *NodeMCU ESP32 Datasheet*. Joy-it. Sept. 2018.

[LCD] LCDWiki. *1.8inch Arduino SPI Module MAR1801 User Manual*. Rev1.0. LCDWiki.

[ST17] National Institute of Standards and Technology. *VVSG Introduction*. Accessed on 2024-01-10. May 2017. URL: https://www.microsoft.com/en-us/research/project/electionguard/.

[Sys23] Espressif Systems. *ESP32-WROOM-32 Datasheet*. Version 3.4. Not Recommended For New Designs (NRND). Espressif Systems. Shanghai, China, Feb. 2023. URL: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf.

[Sys24a] Espressif Systems. *ESP32 Series Datasheet*. Version 4.7. Espressif Systems. Shanghai, China, Sept. 2024. URL: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf.

[Sys24b] Espressif Systems. *ESP32 Series Datasheet*. Version 4.7. Espressif Systems. Shanghai, China, Sept. 2024. URL: https://www.espressif.com/sites/default/files/documentation/esp32_errata_en.pdf.

[Sys24c] Espressif Systems. *ESP32-WROOM-32E Datasheet*. Version 1.7. Espressif Systems. Shanghai, China, Sept. 2024. URL: https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf.

A. Appendix