Ausblick

Christoph Bieringer, Simon Schneider

27.10.2022

Verschiedene Schwachstellen, Probleme und Erweiterungsmöglichkeiten, die in dieser Arbeit nicht behandelt werden konnten, werden aufgelistet und beschrieben. Diese könnten Ausgangspunkte für zukünftige Arbeiten am System sein.

Inhalt

Fazit dieser Arbeit

Das aktuelle System weißt, wie in der Analyse gezeigt werden konnte, zahlreiche konzeptionelle, technische und methodische Mängel auf. Im Rahmen dieser Arbeit konnten allerdings einige davon bereits beseitigt werden. Zukünftige Arbeiten könnten das System noch weiter verbessern und damit eines Tages einen funktionierenden Prototypen hervorbringen. Eine (unvollständige) Reihe an Vorschlägen für zukünftige Arbeiten bietet der Rest dieses Dokuments. Die Vorschläge sind dabei in etwa aufsteigend nach Umfang und Abstraktion sortiert, d.h. die Liste reicht von relativ kleinen Veränderungen am aktuellen System über mögliche Erweiterungen bis hin zu umfassenden theoretischen Betrachtungen.

Unterstützung anderer Wahlformen

Eine mögliche Erweiterung für das System besteht darin, Unterstützung für verschiedene Wahlformen (und angepasste Frontend-Oberflächen) hinzuzufügen. Neben der aktuellen Auswahl aus einer Liste von Alternativen wären dies auch Ja/Nein-Fragen mit Checkboxen oder sog. Write-In-Ballots mit Texteingabefeldern.

Öffentliche Verifizierbarkeit der Auszählung

In der nach dem Ende dieser Arbeit übergebenen Version des Systems ist es den Benutzern zwar mittlerweile möglich, die Integrität der Blockchain zu verifizieren, die Auszählung der Stimmen können sie allerdings nicht selbst nachvollziehen. Dies wäre allerdings technisch möglich.

Hierzu müsste der Wahlserver nur noch die zum entschlüsseln benötigten Informationen (den privaten Administratorschlüssel sowie den Inhalt von directory_server3, d.h. die privaten Wählerschlüssel) öffentlich (z. Bsp. über weitere API-Endpoints) zur Verfügung stellen (erst nach Ende der Wahl, um den schon in der Analyse beschriebenen Angriff mittels der Auszählfunktion auszuschließen). Dann könnte das Frontend oder ein selbstgeschriebener Client mithilfe dieser Daten die in der Blockchain gespeicherten Stimmen selbst entschlüsseln und auszählen. Da in der Blockchain nur die persönlichen Hashwerte (die ja mit einem benutzerdefinierten PIN „gesalzen“ wurden) gespeichert sind, lassen sich von den so verfügbaren Informationen auch keine Rückschlüsse auf die Wahl einzelner Personen ziehen.

Auf diese Weise wäre ein weiterer wichtiger Schritt in Richtung einer vollständig verifizierbaren und damit vertrauenswürdigen Abstimmung gemacht. Vor der endgültigen Implementierung dieser Funktionalität sollte aber evtl. untersucht werden, ob dadurch andere, bisher noch nicht entdeckte Sicherheitslücken geöffnet würden.

Verlagerung der Kryptographie hin zum Client

Weiteres Verbesserungspotenzial bietet die Verlagerung von kryptographischen Berechnungen hin zum Client. In der aktuellen Architektur wird ein Großteil der sensiblen Informationen im Klartext zwischen Wahlserver/Register und Frontend ausgetauscht, obwohl dazu keine theoretische Notwendigkeit besteht. Aufgrund dieser Entscheidung muss der Benutzer Wahlserver und Register absolut vertrauen können, da diese seine Stimme im Klartext sehen.

Hier wäre evtl. zu prüfen, ob eine Möglichkeit besteht, gleiche oder ähnlich sichere kryptografische Funktionen wie in der aktuellen Implementierung auch auf dem Client auszuführen (bspw. mittels der Web Crypto API, die mittlerweile in allen gängigen Browsern unterstützt wird). Wenn dies der Fall ist, könnte die Ver- und Entschlüsselung der Stimme bei der Kommunikation mit Wahlserver und Register vom Client selbst durchgeführt werden. Die restlichen Komponenten des Systems würden

dann nur noch die verschlüsselte Stimme sehen, sodass der Benutzer ihnen weniger Vertrauen entgegenbringen muss.

Eine solche Änderung würde große Teile des bisher geschriebenen Codes, sowohl im Frontend wie im Backend, betreffen und wäre vermutlich sehr aufwändig. Der Gewinn durch das zusätzliche Maß an Vertraulichkeit wäre aber beträchtlich und ein wichtiger Schritt hin zu einem ernstzunehmenden Prototypen.

Verbesserung des Mix-Netzwerks

Die aktuelle Version des Wahlservers implementiert nur einen Teil der für ein vollständiges Mix-Netzwerk benötigten Funktionalität. Insbesondere werden die Nachrichten zwar zufällig von einem Wahlserver zum nächsten geroutet, sodass der eigentliche Empfänger (das Register) nicht mehr direkt auf den Absender (den einzelnen Wähler) schließen kann, aber keine weiteren Verschleierungsmaßnahmen getroffen.

Ein Angreifer, der den Verkehr zwischen Wählern, Wahlservern und Registern zumindest teilweise mithören kann, kann also über Timing, Nachrichtenlänge etc. trotz der Verwendung des Mix-Netzwerks u.U. herausfinden, welche Nachricht von welchem Wähler stammt. Mit dieser Information könnte er in der Lage sein, einzelne Wähler zu deanonymisieren.

Aus diesem Grund implementieren reale Mix-Netzwerke neben dem Routing über mehrere Mix-Server hinweg noch weitere Features, etwa dass Verzögern und umordnen der empfangenen Nachrichten sowie das Weiterleiten von Nachrichten in größeren Batches. Solche Features würden auch das Voting-System deutlich verbessern und kommen damit ebenfalls für eine zukünftige Erweiterung in Frage.

Bessere Sicherung des Administrator-Schlüssels

Eine mögliche Erweiterung besteht darin, den Administrator-Schlüssel (bzw. dessen privaten Teil) besser zu schützen. Wie in der Analyse bereits angemerkt wurde, stellt die Verfügbarkeit des Administratorschlüssels ein Problem dar, da er von einem kompromittierten Wahlserver aus in falsche Hände gelangen könnte. Böswillige Akteure könnten dann mit ihm u.a. versuchen, Wähler zu deanonymisieren).

Um dieses Risiko zu vermeiden, könnte beispielsweise der private Teil des Administratorschlüssels mittels Secret-Sharing-Methoden auf mehrere Beteiligte verteilt werden (bspw. unterschiedliche Teile der Wahlbehörde oder unabhängige Organisationen). Dies würde einen Missbrauch stark erschweren, da nun mehrere Parteien zusammenarbeiten müssen, um den privaten Schlüssel zu berechnen. Für herkömmliche Secret-Sharing-Methoden wird allerdings zumindest für das Setup eine vertrauenswürdige Partei benötigt, die das Secret (in diesem Fall den privaten Administratorschlüssel) kennt. Das Problem, dass der Schlüssel in falsche Hände geraten könnte, wird also nur ein Stück verschoben und müsste nach wie vor anderweitig (Vertrauen, Sicherheitsprozeduren, Zugriffsbeschränkungen etc.) behoben werden. Mittels *Distributed Key Generation* ist es aber auch möglich, ein Schlüsselpaar zu erzeugen, ohne dass eine Stelle den privaten Schlüssel vollständig kennt (dieser wird also überhaupt erst berechnet, wenn genug Teilnehmer dies möchten). Auch andere Methoden zur Sicherung des Administratorschlüssels sind denkbar.

Untersuchung des Systems auf Schwächen gegenüber Quantencomputern

Im Zuge des technischen Fortschritts werden Quantencomputer stetig robuster und leistungsfähiger. In nicht allzu ferner Zukunft könnten diese in der Lage sein, aktuelle kryptografische Primitive anzugreifen. Jedes ernstzunehmende kryptografische System sollte vor diesem Hintergrund also die Frage beantworten können, wie resistent es gegenüber Angreifern ist, die über einen Quantencomputer verfügen.

Die Suche nach Systemen, die einem Angriff mit Quantencomputern widerstehen können, ist ein aktives und aktuelles Forschungsfeld und wird oft unter dem Begriff der Post-Quantum-Kryptographie zusammengefasst. Eine erste Übersicht bietet bspw. (Bernstein und Lange 2017). Zentral sind dabei insbesondere folgende zwei Punkte:

1. Shors Algorithmus ermöglicht es einem Quantencomputer, sowohl das Faktorisieren von Zahlen als auch das Berechnen diskreter Logarithmen deutlich schneller durchzuführen als herkömmliche Rechner. Sobald ausreichend leistungsfähige Quantencomputer zur Verfügung stehen, könnte ein Angreifer mit ihnen auf diese Weise zahlreiche derzeit gängige asymmetrische Verschlüsselungs- und Signierverfahren angreifen. Hiervon betroffen sind neben RSA und ElGamal insbesondere auch ECC-basierte Systeme, wie sie im hier betrachteten System verwendet werden.
2. Grovers Algorithmus ermöglicht eine ausreichend dimensionierten Quantencomputer die unstrukturierte Suche über einer Menge von M Elementen in O($\sqrt{M}$) Schritten. Das Verfahren ist äußerst allgemein und betrifft damit insbesondere auch symmetrische Chiffren wie etwa das im betrachteten System verwendete AES, aber auch die Umkehrung von Hash-Funktionen wie etwa SHA-256 (ebenfalls im aktuellen System verwendet). Die Reduzierung der benötigten Ausführungsschritte auf $\sqrt{M}$ entspricht bei einem Kryptosystem demselben Sicherheitsverlust wie bei einer Halbierung der verwendeten Schlüssel- bzw. Outputlänge.

Für die Weiterentwicklung lassen sich hieraus bereits erste Schlüsse ziehen. Die verwendeten symmetrischen Chiffren, Hash-Funktionen und andere von Grovers Algorithmus betroffene Elemente sollten:

1. Gesucht und aufgelistet
2. Auf Sicherheitsanforderungen und aktuelle Schlüssellänge (oder ähnliche Parameter) untersucht
3. Bei Bedarf durch Versionen mit größeren Schlüsseln/Outputs/… ersetzt werden.

Bei den asymmetrischen Verfahren, die durch Shors Algorithmus bedroht werden, ist die Sachlage wesentlich komplizierter. Aktuell werden von verschiedenen Stellen neue Post-Quantum-Verfahren untersucht, die diese ersetzen könnten, bspw. im Rahmen des NIST Post-Quantum-Cryptography-Standardization-Prozesses[1]. Solche Verfahren eignen sich i.d.R. aber nur bedingt für den Einsatz in Produktivsystemen. Gründe hierfür sind neben tlw. sehr hohen Rechenleistungs- oder Speicherverbrauch und mangelnder Verfügbarkeit von Implementierungen auch die allgemein geringe Erfahrung mit ihnen. Aufgrund dieser könnten viele der Verfahren Schwachstellen bergen, die erst im Laufe der Zeit (mit genaueren Untersuchungen) deutlich werden. So wurde bspw. der sog. Supersingular Isogeny Key Exchange, der im Rahmen des NIST-Prozesses vorgeschlagen wurde, erst kürzlich auf herkömmlicher PC-Hardware erfolgreich gebrochen (vgl. Castryck und Decru 2022). Vor

[1] Online verfügbar unter https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization.

einer evtl. Verwendung von Post-Quantum-Verfahren im Wahlsystem sind also noch viele Fragen zu klären.

Das derzeitige System vorerst ohne Änderungen weiterzuverwenden stellt allerdings auch keine zufriedenstellende Lösung dar. Aktuelle Fortschritte in der Verfügbarkeit und Leistung von Quantencomputern legen die Befürchtung nahe, dass die weiter oben beschriebenen Angriffe mittelfristig Realität werden könnten. Spätestens dann müssen die herkömmlichen Verfahren ersetzt werden. Bei einem Wahlsystem könnte dies aber nicht ausreichen, da eine Wählerstimme i.d.R. auf unbegrenzte Zeit geheim bleiben sollte[2] und ein Angreifer einfach die verschlüsselten Daten auslesen und speichern könnte, um sie in Zukunft zu entschlüsseln, sobald ihm ein geeigneter Quantencomputer zur Verfügung steht.

Aus diesem Grund ist die Beschäftigung mit dem Thema Post-Quantum-Kryptographie essenziel für eine mögliche Produktivanwendung des Wahlsystems und damit eine wichtige zukünftige Arbeit.

Untersuchung anderer theoretischer Ansätze

Das aktuelle System stellt nur eine von vielen möglichen Architekturen für ein E-Voting-System dar. Ein Vergleich mit anderen Systemen aus Industrie und Forschung könnte dabei helfen, Stärken und Schwächen des gegenwärtigen Entwurfs besser zu verstehen. Auf diese Weise könnten auch mögliche Erweiterungen und Verbesserungen gefunden werden, die in dieser Arbeit noch nicht beschrieben wurden.

Aktuelle Systeme wären (wie bereits erwähnt) u.a.:

- die Voatz-App
- das E-Voting-System der Schweizer Post
- das POLYAS-System.

Mögliche Beispiele für Verfahren aus der Forschung wären etwa:

- andere mögliche Verwendungen für Blockchains (vgl. bspw (Al-Maaitah und Qatawneh 2021) oder (Jafar et al. 2021) für einen Überblick)
- die Verwendung anderer kryptografischer Werkzeuge für Teile des Wahlprozesses, etwa:
    - Zero-Knowledge-Verfahren (vgl. bspw. (Panja und Roy 2018))
    - homomorphe Verschlüsselung (vgl. bspw. (Hirt und Sako 2000)))
- Eine eher methodische statt kryptographische Möglichkeit, den Wahlprozess widerstandsfähiger zu machen, ist die Abgabe von ungültigen, aber nicht direkt als solchen erkennbare Stimmen, sog. Decoy-Ballots (vgl. bspw. Chaum und (Gersbach et al. 2017) für eine mögliche Schwachstelle).

[2] Falls dies nicht der Fall ist (d.h. die Stimmen noch zu Lebzeiten der Wähler publik werden), könnte ein böswilliger Akteur statt Belohnungen/Sanktionen einfach das Versprechen auf bzw. die Drohung mit zukünftigen Belohnungen respektive Sanktionen verwenden, um Wähler zu beeinflussen.

Quellen

Al-Maaitah, S., Qatawneh, M., & Quzmar, A. (2021). E-Voting System Based on Blockchain Technology: A Survey. *2021 International Conference on Information Technology (ICIT)*, 200-205.

Bernstein, D. J., & Lange, T. (2017). Post-quantum cryptography. *Nature, 549(7671)*, 188–194.

Castryck, W., & Decru, T. (2022). An efficient key recovery attack on SIDH (preliminary version). *Cryptology ePrint Archive*.

Chaum, D. Random-Sample Voting: More democratic , better quality , and far lower cost. Online verfügbar unter: https://rsvoting.org/whitepaper/white_paper.pdf (Letzter Zugriff 25.11.2022)

Gersbach, H., Mamageishvili, A., & Tejada, O. (2017). Sophisticated Attacks on Decoy Ballots: The Devil's Menu and the Market for Lemons. *arXiv preprint arXiv:1712.05477*.

Hirt, M., & Sako, K. (2000). Efficient Receipt-Free Voting Based on Homomorphic Encryption. *EUROCRYPT.*

Jafar, U., Aziz, M.J., & Shukur, Z. (2021). Blockchain for Electronic Voting System—Review and Open Research Challenges. *Sensors (Basel, Switzerland),* 21.

Panja, S., & Roy, B.K. (2018). A secure end-to-end verifiable e-voting system using zero knowledge based blockchain. *IACR Cryptol. ePrint Arch.*, 2018, 466.