22. Nachwuchswissenschaftler*innenkonferenz (NWK)

https://doi.org/10.52825/ocp.v2i.139

Publiziert: 15. Dez. 2022

Entwicklung und Evaluation eines anonymitätsfokussierten Feedbacktools auf Ring-Signatur-Basis

Martin Schuster[1*]

[1] Hochschule Mittweida

* Korrespondenz: mschust3@hs-mittweida.de

Kurzfassung. Für die Blockchain Academy Mittweida sollte ein Feedback-System entwickelt werden, welches das Feedback für die angebotenen Kurse, anonym und Duplikation frei speichern kann. Es sollte möglich sein, ohne die Anonymität der Nutzer aufzuheben, festzustellen zu können, ob bereits Feedback eines Nutzers eingereicht wurde. Dafür wurde die Technologie der Linkable-Ring-Signaturen verwendet. Mit dieser Technologie können Nutzer zu einem Ring zusammengefasst werden und anonym Feedback einreichen. Um das Interesse und das Verständnis für diese Technologie abzufragen, wurde ein Video erstellt, welches das Verfahren vorstellt und dieses einer Probandengruppe präsentiert. Die Evaluation ergab, dass das Interesse an der vorgestellten Lösung vorhanden ist und das Konzept vermittelt werden konnte.

1 Einleitung

Mit der zunehmenden Digitalisierung von analogen Vorgängen hat sich eine sicherheitskritische Problematik entwickelt. Diese besteht darin, die Anonymität des Nutzers zu gewährleisten und gleichzeitig sicherzustellen zu können, dass diese Anonymität nicht missbräuchlich ausgenutzt wird. Einen möglichen Lösungsansatz dafür bieten die Ring-Signaturen. Diese erlauben es, stellvertretend für eine Gruppe von mehreren Personen Daten zu signieren, ohne dabei die eigene Identität preiszugeben. Es existieren vollständige anonyme Varianten und jene, die eine Verknüpfung von mehreren Signaturen eines Senders erlauben. Dadurch wird die Technologie für viele Anwendungsbereiche wie beispielsweise Meinungsäußerungs-, E-Voting- oder E-Cash-Systeme interessant. Für die Blockchain Academy Mittweida, war es das Ziel, solch ein System zum Aufnehmen von Feedback für die Kurse zu entwickeln. Um das Interesse an dieser Lösung zu erfragen und das Verständnis für Technologie zu fördern galt es ein Erläuterungsvideo zu erstellen. Die Forschungsfrage, die sich dabei stellt, ist, ob man mit einer visuellen Erläuterung das Verständnis für eine Technologie bei Personen fördert und dadurch die Akzeptanz diese zu nutzen erhöht werden kann.

2 Methodik

2.1 Ring Signaturen

Ring-Signaturen, 2001 erstmals von Rivest, Shamir und Tauman vorgestellt, ermöglichen es, dem Nutzer eine Nachricht stellvertretend für seine zugehörige Gruppe zu signieren, ohne dabei seine eigene Identität preiszugeben. Die Gruppe an möglichen Unterzeichnern wird als Ring bezeichnet, der signierende wird als Unterzeichner betitelt und alle anderen als Nicht-

Unterzeichner. Es wird vorausgesetzt, dass alle Beteiligten über einen öffentlichen und privaten Schlüssel P_k S_k, welcher das Signaturverfahren und seinen Verifikationsschlüssel beinhaltet, verfügen. Ein Ring-Signaturverfahren ist durch zwei Prozessschritte definiert. [1]

1. Ring-Unterzeichnung (*ring-sign(m,P1,P2,...,Pr,z,Sz)*), diese produziert die Ring-Signatur *σ* für die Nachricht *m*, mit den gegebenen öffentlichen Schlüsseln *P1,P2,...,Pr* der *r* Ring Mitglieder, zusammen mit dem geheimen Schlüssel *Sz* des Unterzeichners *z*.
2. Ring-Signatur-Verifizierung (*ring-verify(m, σ)*) diese nimmt die Nachricht m und die Signatur *σ* entgegen und gibt ein wahr oder falsch aus, je nachdem ob es eine valide Signatur ist oder nicht.

Der Originalentwurf von Rivest, Shamir und Tauman hatte das Hauptaugenmerk auf der Anonymität des Unterzeichners und sah keinen Widerruf dieser Anonymität vor. [1] Da diese Eigenschaft nicht für alle Anwendungszwecke zielführend beziehungsweise nützlich ist, wurde das Verfahren erweitert. Mit den Linkable-Ring-Signaturen beispielsweise wurde eine Nachvollziehbarkeit, ob ein Mitglied bereits eine Nachricht aus der Gruppe gesendet hat, integriert. [2] Es haben sich noch weitere Varianten wie Tracable-, Threshhold- oder Revocable-Ring-Signaturen entwickelt und zu diesen jeweils noch weitere Abwandlungen, sodass es bereits eine Vielzahl an Varianten gibt, die für unterschiedliche Einsatzzwecke verwendet werden können.

2.1 REST-API-Server

Um die Ring-Signatur Funktionalität möglichst flexible nutzbar zu machen, wurde sich dazu entschlossen, diese als ein REST-API-Server umzusetzen. Ein Vorteil dieses Programmierparadigmas ist die Skalierbarkeit des bereitgestellten Service, da jede Anfrage durch den Client in sich geschlossen ist. Dies bedeutet, dass in jeder Nachricht an den Server und als Antwort vom Server an den Client alle benötigten Informationen zur weiteren Verarbeitung enthalten sind. Die Anfragen an den Server werden dabei durch einen HTTP-Request realisiert. Durch diesen wird geregelt, wie der Client seine Anfrage formuliert und wie der Server auf diese antwortet. Die Anfragen werden dabei mittels der POST Methode realisiert. Diese überträgt die Anfrage im HTTP-Header, was das Senden größere Datenmengen und möglicherweise sensibler Daten ermöglicht. Der Server wurde auf Node.Js-Basis umgesetzt. Als Basis wurde das Express.JS Framework der OpenJS Foundation genutzt. Dieses bietet bereits vorgefertigte Komponenten, um das Erstellen von REST-API-Funktionen zu erleichtern und zu vereinfachen. Es erlaubt durch vorgefertigte Komponenten, sogenannten Routen, gezielt Funktionen bereitzustellen, welche über URLs ausgelöst werden können. Für die Umsetzung der Ring-Signatur Variante wurde sich für die Linkable-Ring-Signatures entschieden. Dafür wurde eine Umsetzung von Victor Taelin genutzt, welcher eine Implementation in JavaScript und PureScript dafür erstellt hat. Diese Umsetzung wurde als Modul in die Node.JS Installation eingefügt und stellt dadurch dem Server die Funktionalität zum Generieren von privaten und öffentlichen Schlüsseln, das Signieren einer Nachricht, das Überprüfen einer Signatur zu einer Nachricht sowie das Überprüfen, ob zwei Signaturen von einem Nutzer stammen, zur Verfügung. Es wurden jeweils eine eigene Route für jede der Funktionen erstellt. Diese erwarten die Daten, welche gesendet werden, im JSON-Format und liefern das Ergebnis ebenfalls im JSON-Format zurück. Dadurch soll ein einheitlicher Workflow mit den Eingabe- und Ausgabedaten gefördert werden.

Um die Fähigkeiten des REST-API-Servers zu testen und eine mögliche Limitierung ermitteln zu können, wurde ein Performance-Test (Abbildung 1) durchgeführt. Es wurde eine Testapplikation auf Node.JS-Basis geschrieben, welche automatisiert Anfragen zur Signierung einer Nachricht an den Server schickt. Die Ergebnisse für die Bearbeitungszeit, die Länge der Signatur und die Anzahl der Ringmitglieder wurde in eine entsprechende Log-Datei geschrieben. Dies geschieht dabei, beginnend mit einem Ringmitglied und nachfolgende in 5 Mitglieder

Schritten, mit bis zu 500 Ringmitgliedern. Die Begrenzung auf 500 Mitglieder hat sich als limitierender Faktor des Servers herausgestellt, da die Länge der Signatur zu lang wird, um diese ohne weitere Modifikationen an der Konfiguration des Servers, zu versenden.

Die Auswertung der Daten hat ergeben, dass das Verfahren wie schon in der initialen Vorstellung von Liu, J. K. und Wong, D. S., ein lineares Verfahren ist. Insgesamt bleibt die Berechnungszeit bei 500 Nutzern mit 2500 Millisekunden beziehungsweise 2,5 Sekunden noch in einem angemessenen Rahmen. Die Länge der Signatur ist bei 500 Nutzer mit 95000 Zeichen sehr lang, sodass bei einer geplanten Speicherung in einer Datenbank dies besonders beachtet werden müsste.

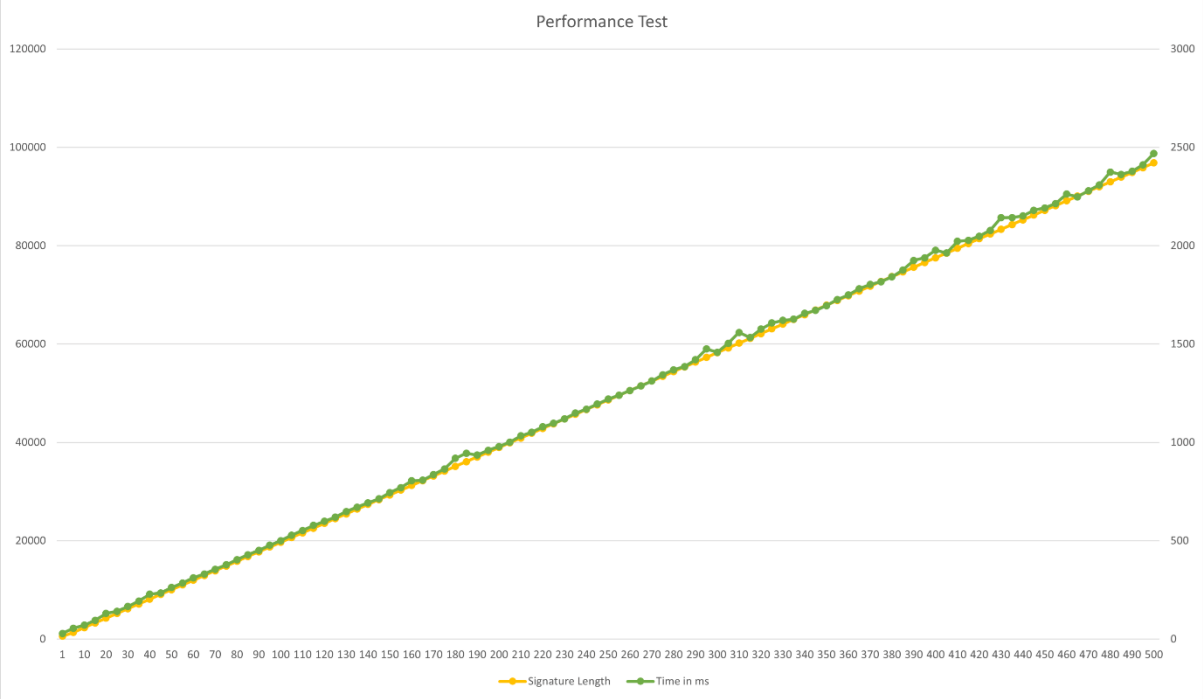

Abbildung 1: REST-API-Server Performance Test

2.3 Visualisierung

Damit das Verständnis und die Akzeptanz für das Vorhaben erprobt werden kann, galt es eine Visualisierung des Themas anzufertigen. Das Ziel, welches für die Visualisierung gesetzt wurde, ist, dass es leicht verständlich und den Empfänger nicht mit zu vielen technischen Details überfordern soll. Dafür wurde zunächst ein grobes Konzept angefertigt, welches den groben Ablauf beschreibt. Anschließend wurde der Text für das Voice Over ausgearbeitet und eingesprochen. Als nächstes wurde Animation mittels Vektorgrafiken erstellt, da diese sich einfacher animieren ließen durch das sogenannte Vektor-Tweening. Dabei werden die Zwischenschritte zwischen zwei Einstellungen automatisch generiert. Abschließend wurde die Animation und die Tonspur in einem Videoschnitt Programm zusammengefügt und das fertige Video auf der Videoplattform YouTube hochgeladen für die Verwendungen in der Evaluation.

3 Evaluation

Die Evaluation wurde in der zweiten Kalenderwoche 2022 durchgeführt. Die Probandengruppe aus 35 Personen, welche zu 77,1% männlich waren, mit 85,7% der Altersgruppe 21 bis 29 Jahre angehörten und mit 77,1% über einen Bachelorabschluss verfügten empfanden das Erläuterungsvideo als gut bis sehr gut Verständlich (91,4%). Das Interesse an der vorgestellten Lösung war bei 57,2% der Befragten gegeben und 94,3% der befragten würden solch ein System benutzen, um ihre Meinung Online mittweilen zu können.

4 Ergebnis

Im Rahmen der Forschungsarbeit konnte eine Prototypische Anwendung entwickelt werden, welche alle geforderten Funktionen enthielt. Aktuell wird diese Anwendung noch nicht im Produktivbetrieb eingesetzt, da die momentane Lösung zu zentralisiert ist sowie die gewünscht Einbindung einer Blockchain Technologie noch ausstehend ist. Die geleistete Arbeit bildet jedoch die Grundlage für eine Fortführung als Masterarbeit Projekt, dabei wird der Ansatz durch die Verwendung einer Browserextension sowie von Smart Contracts dezentralisiert.

Datenverfügbarkeit

Die in der Arbeit verwendeten Daten sind aufgrund von Sicherheitsbedenken nicht öffentlich einsehbar beziehungsweise wurden so weit wie nötig in der Arbeit offengelegt.

Interessenskonflikte

Es liegen keine Interessenskonflikte vor.

Literaturverzeichnis

1. Rivest, R. L., Shamir, A. & Tauman, Y. (2001): How to Leak a Secret. In C. Boyd (Hrsg.), Advances in Cryptology ASIACRYPT 2001 (S. 552–565). Springer Berlin Heidelberg.
2. Fujisaki, E. & Suzuki, K. (2007). Traceable Ring Signature. In T. Okamoto & X. Wang (Hrsg.), Public Key Cryptography PKC 2007 (S. 181–200). Springer Berlin Heidelberg.