Thesis-Seminar im Bachelor-Studium – monatliche Berichterstattung

Berichtszeitraum: 01.04.2019 – 30.04.2019

Thema der Thesis:	Anwendung von Blockchain Technologien zur kryptografischen Absicherung von Auditdaten.
Bearbeiter:	Kevin Hertfelder
Kontakt: kevin.hertfelder@hs-furtwangen.de	

Durchgeführte Arbeiten

Im zweiten Monat habe ich alle grundlegenden Funktionen implementiert und einen ersten testfähigen Prototyen erstellt. Für alle neuen Funktionen wurden parallel Unit Tests erstellt. Die Planung und Implementierung für Verbesserungen und alternative Betriebsverfahren ist auch im Gange. Alternativen soll es bei Persistierungsstrategien und den Hashverfahren geben. Zusätzlich zu der Absicherung durch ein Hashverfahren soll es eine regelmäßige Absicherung durch digital signierte Einträge geben. Eine einfache Implementierung, die einen lokalen RSA benutzt, ist auch bereits abgeschlossen, diese soll aber auch durch alternative Verfahren ersetzt werden können. Ein Ziel bei der Implementierung ist es so viel wie möglich modular und austauschbar zu halten, um einfach und präzise auf Kundenanforderungen oder Gesetzesänderungen reagieren zu können. Wenn es offensichtliche Alternativen für verschiedene Anforderungen gibt, sollen diese als Standardimplementierung direkt angeboten werden können.

Erzielte Ergebnisse

- LifeCycle basierte Erstellung von Auditlogs und deren Verifikation.
- Rest-Schnittstelle zur Verifikation der gesamten, mit der Hashverifikation geschützten Daten.
- Rest-Schnittstelle zur Verifikation einzelner Auditevents.
- Implementierung einer Hashchain zur Absicherung und Überprüfung von Auditevents über ihren Hashwert.
- Implementierung eines Merkletrees mit demselben Ziel wie die Hashchain, aber mit erhöhter Sicherheit durch höhere Vernetzung der Hashwerte.
- Implementierung für direkte Persistierung während der Transaktion für eine direkte Verfügbarkeit und Sicherheit darüber, dass das Persistieren erfolgreich war.
- Implementierung einer aggregierten Persistierung, die über einen Zeitraum eingehende Events aggregiert und diese dann in Bulk abspeichert. Dies erhöht die Performance durch wenige und dementsprechend größere Datenbankzugriffe. Das Aggregieren ist effektiver, je größer die Transaktionsmenge pro Zeit wird.
- Implementierung eines lokalen DSA Verfahrens mit Signierung und Verifikation der Signatur.

Abweichungen / Probleme

Das größte Problem in dieser Phase war die “total order“ über mehrere Instanzen zu gewährleisten. Der Audit-Log benötigt zur Verifikation eine eindeutige Reihenfolge von Events. Zu erst habe ich einen naiven Ansatz verfolgt und wollte das Ordering-Problem über hochauflösende Zeitstempels lösen. Dies stellt aber ein Problem dar, wenn die Uhren der einzelnen Instanzen nicht synchronisiert sind und wenn die Zeitstempel durch Latenzzeiten außer Ordnung geraten. Zum Lösen dieses Problems gibt es zahlreiche Ansätze und die zwei populärsten Lösungen sind entweder eine zentrale Instanz zu bestimmen, die dann für die Ordnung verantwortlich ist, oder ein Kommunikationsprotokoll zwischen den Instanzen zu implementieren, um eine Einigung zwischen allen Instanzen zu garantieren. Die Zweite Variante ist in Open Source Bibliotheken wie Hazelcast oder Apache Ignite implementiert und wurde zur Lösung dieses Problems verwendet. Ein Weiteres Problem ist das Arbeiten mit einer Baumstruktur zur Verifikation. Da dieser beliebig groß werden kann ist es unmöglich den kompletten Baum serverseitig in Memory zu halten. Wenn der Baum nicht

komplett in Memory gehalten wird, fallen viele Vorteile eines Baums, wie log n Zugriffszeiten auf jedes Blatt, weg und der Baum verliert seinen Mehrwert. Aus diesem Grund muss das Verfahren mit dem Merkletree nochmals überarbeitet. Geplant ist eine Mischform aus einer Hashchain verbundenen mit kleinen Bäumen mit fester Größe.

Ausblick über die geplanten Tätigkeiten und Ergebnisse des nächsten Berichtszeitraums

Die Hashverifikation bildet im Moment eine Verifikationsstruktur über alle Events im gesamten System. Dies bietet hohe Sicherheit durch die große Menge an einzelnen Hashwerten, jedoch ist die Verifikation von einzelnen Dokumenten, welches ein häufiger Anwendungsfall ist, aufwendig und langsam. Deshalb soll eine unabhängige Verifikation einzelner Dokumente erstellt werden. Weiter sollen Signaturen als Ankerpunkte in die Hashstruktur integriert werden. Zuletzt sollen Lasttests erstellt werden, um die Performance zu evaluieren und verschiedene Verfahren zu vergleichen.