Searchitect

Endbericht | Call 12 | Projekt ID 2099

Inhalt

1 Einleitung.....5

2 Projektbeschreibung.....5

3 Verlauf der Arbeitspakete.....6

3.1 *Arbeitspaket 1 - Evaluierung Usecases Consumer*.....6

3.2 Arbeitspaket 2 - *Design Architektur*.....6

3.3 Arbeitspaket 3 - *Implementierung Webservice*.....7

3.4 Arbeitspaket 4 - *Implementierung ClientLib*.....7

3.5 *Arbeitspaket 5 - Demo ClientApp*.....8

3.6 *Arbeitspaket 6 - SE Verfahren auswählen und integrieren*.....8

3.7 *Arbeitspaket 7 - Auswahl weiterer SE Verfahren*.....8

3.8 *Arbeitspaket 8 - Integration in Drittsoftware*.....9

3.9 *Arbeitspaket 9 – Hosting*.....9

3.10 *Arbeitspaket 10 – Wartung*.....9

3.11 *Arbeitspaket 11 - Integrationstests & Quantitative Analyse*.....9

3.12 *Arbeitspaket 12 - Dokumentation erstellen*.....10

3.13 *Arbeitspaket 13 – Paper*.....10

3.14 *Arbeitspaket 14 – Öffentlichkeitsarbeit*.....10

3.15 *Arbeitspaket 15 – Projektmanagement*.....11

4 Liste Projektendergebnisse.....11

5 Verwertung der Projektergebnisse in der Praxis.....12

6 Öffentlichkeitsarbeit/ Vernetzung.....13

7 Geplante Aktivitäten nach netidee-Projektende.....13

8 Anregungen für Weiterentwicklungen durch Dritte.....13

1 Einleitung

Das Ziel von Searchitect war es, die Kategorie der Verfahren namens „Searchable Encryption“ (SE - „Durchsuchbare Verschlüsselung) einfach nutzbar zu machen. Dabei handelt sich um kryptographische Verfahren, mit deren Hilfe eine Suche auf verschlüsselten Daten möglich ist. Eine Integration in Webservices bzw. Clouddienstleistungen sollte es ermöglichen gleichzeitig eine online Verarbeitung der Dokumente durchzuführen und die Vertraulichkeit der Dokumente gegenüber dem Dienstanbieter abzusichern. Cloudanbieter sollte es so erleichtert werden, Lösungen nach dem Prinzip „Privacy by Design“ umzusetzen.

2 Projektbeschreibung

Beschreibung der Projektziele / Zielgruppe und inhaltlicher Überblick über das Projektergebnis (max. 5 Seiten)

Searchitect will als Schnittstelle fungieren, die die gesamte Kryptographie implementiert und den Entwicklerinnen in einer einfach zu verwendenden API zur Verfügung stellt. In seiner Gesamtheit ist Searchitect komplett unabhängig von der tatsächlichen Speicherung der Daten, mit der es sich nicht beschäftigt (da es hierfür schon zahlreiche Lösungen gibt). Die Hauptkomponente ist dabei ein Webservice, welches nach einem Microservice Ansatz implementiert wurde.

Userseitig wird davon nur das „Gate“ exponiert. Es authentifiziert die User und leitet Suchanfragen an die jeweiligen Backends im Hintergrund weiter und schickt deren Antworten zurück. Dabei exisitert ein Backend pro verwendetem Searchable Encryption Verfahren. Die Speicherung des verschlüsselten Suchindex wird ebenfalls vom Backend übernommen, und zwar in vom User angelegten „Repositories“.

Die zweite entscheidende Komponente ist die „ClientLib“. Sie verbindet sich mit einem Searchitect-Webservice, um dort einen Suchindex anzulegen bzw. zu aktualisieren und Suchanfragen an diesen zu stellen. Für diese ClientLib exisiteren wieder, analog zu den serverseitigen Backends, einzelne Plugins, die die Logik der einzelnen kryptographischen Algorithmen zur Verfügung stellen. Clientseitig muss noch ein „Clientstate“ gespeichert werden, der geheimes

Schlüsselmaterial (ohne das auf das Repository nicht zugegriffen werden kann) und Teile des Suchindex enthält, ohne die keine Suchanfragen erzeugt werden können.

Für die Verwendung der ClientLib gibt es zwei Möglichkeiten: Bevorzugt wäre es, diese direkt in eine bestehende Applikation zu integrieren. Alternativ kann man auch einen Standalone Client entwickeln, der die Suchfunktionalität anbietet. Im Rahmen von Searchitect wurden beide Varianten demonstrativ implementiert.

3Verlauf der Arbeitspakete

Hinweis: Sofern sich seit dem Zwischenbericht bei diesem Arbeitspaket keine Veränderungen ergeben haben, kann der betreffende Text unverändert aus dem Zwischenbericht übernommen werden.

3.1 Arbeitspaket 1 - Evaluierung Usecases Consumer

Bitte Arbeitspakete gemäß Excel verwenden

Kurzbeschreibung der Haupttätigkeiten

Besondere Erfolge/ Probleme

Gab es große Abweichungen zum Plan? Warum?

Erkenntnisse zur Vorgangsweise

Erstellung einer Onlineumfrage bezüglich möglicher Usecases des Searchitect Frameworks als Vorbereitung für die Integration im AP8. Diese Umfrage ergab u.a., dass UserInnen in erster Linie in ihren Emailclients bzw. in ihrer Dokumentenverwaltung die Suchfunktionen nützen. Des weiteren gaben 82% an, dass sie Technologien zur Speicherung von Daten mit dem Feature Verschlüsselung nutzen würden. Derzeit ist der Ort der Speicherung nahezu gleichmäßig verteilt unter: Cloudservice, USB-Stick und externe Festplatte mit je rund 20% - lediglich der PC/Laptop setzt sich hier mit 35% ab.

3.2 Arbeitspaket 2 - *Design Architektur*

Bitte Arbeitspakete gemäß Excel verwenden

Kurzbeschreibung der Haupttätigkeiten

Besondere Erfolge/ Probleme

Gab es große Abweichungen zum Plan? Warum?

Erkenntnisse zur Vorgangsweise

Festlegung der API des Webservices und der Clientbibliothek sowie der Authentifizierungsmethode.

Entscheidung für Microservice Architektur mit einem RESTful Interface.

Die APIs wurden festgelegt.

Die Authentifizierung erfolgt auf Basis von JWT (JSON Web Token)

Das System is zu organisch gewachsen, eine stärker koordinierte Vorgangsweise am Beginn wäre hilfreich gewesen.

3.3 Arbeitspaket 3 - *Implementierung Webservice*

Bitte Arbeitspakete gemäß Excel verwenden

Kurzbeschreibung der Haupttätigkeiten

Besondere Erfolge/ Probleme

Gab es große Abweichungen zum Plan? Warum?

Erkenntnisse zur Vorgangsweise

Der serverseitige Teil der in AP2 festgelegten Architektur wurde mit Hilfe des Spring Boot Frameworks implementiert.

Der konsequente Einsatz von Test Driven Development ermöglichte eine konsistente Weiterentwicklung

Die frühzeitige Erstellung von Dockerfiles als Vorbereitung für Integrationstests war ebenfalls hilfreich.

Integration von JWT in das Security Framework von Spring Boot war aufwändiger als erwartet. Diese werden zur Authentifizierung von RESTful Anfragen an das Searchable Encryption Webservice verwendet.

3.4 Arbeitspaket 4 - *Implementierung ClientLib*

Der clientseitige Teil der in AP2 festgelegten Architektur wurde mit Hilfe des Spring Frameworks implementiert.

Wir kamen leider zur Erkenntnis, dass deutlich mehr Searchable Encryption Logik in die Clientbibliothek einfließen muss als ursprünglich geplant bzw. erwartet, das heißt die Clientkomponenten sind nicht so schlank wie erhofft, was auch die Portierung auf andere Sprachen klarerweise erschwert.

3.5 *Arbeitspaket 5 - Demo ClientApp*

Zur Demonstration der Möglichkeiten der ClientLib wurde eine Kommandozeilen Client erstellt der mit dem Searchitect Server interagieren kann. Die implementierten Funktionen sind:

- Erstellen eines Users am Gate

- Erstellen eines neuen Repositories auf einem der verfügbaren SE Backends
- Aktualisierung eines bestehenden Repositories
- Durchführung einer Suchoperation

Ein Problem, welches mit den in AP4 geschilderten Problemen zusammenhängt, ist die Notwendigkeit eines „Client State“. Dieser enthält für die Erstellung der Suchtoken relevante Informationen sowie Schlüsselinformationen und muss für die Durchführung vorhanden sein und daher zwischen eventuell vorhandenen mehreren Hosts synchron gehalten werden. Ein Ausweg wäre es, den Clientstate verschlüsselt auf dem Gate bzw. im Backend abzulegen. Aus Zeitgründen (Mehraufwand AP3) konnte diese Lösung aber leider nicht implementiert werden.

Die Erstellung eines mobilen Clients musste aus den gleichen Gründen abgesagt werden.

3.6 *Arbeitspaket 6 - SE Verfahren auswählen und integrieren*

Es konnte gut auf bestehende Arbeiten aus einem Vorprojekt am Kompetenzzentrum für IT-Security aufgebaut werden.

Als erste Verfahren wurde „DynRH2Lev“ aus der Clusion Library ausgewählt und integriert.

Die Speicherung der Datenbank erfolgte dabei zunächst in einem rein Memory-basierten Storage. Später wurde auch eine adaptierte Variante integriert, welche einen persistenten Storage verwendet, der auf dem Open Source Key-Value Store „RocksDB“ von Facebook basiert.

Wir kamen rasch zur Erkenntnis, dass eine starke Fokussierung auf Verfahren, die „Forward Security“ bieten notwendig ist, um ein wirklich sicheres System zu schaffen.

Einige Bugs der DynRH2Lev Implementierung aus der Clusion Library wurden behoben, außerdem wurde die persistente Variante optimiert hinsichtlich der Reduktion der Update- und Suchlaufzeit und des Speicherbedarfs des verschlüsselten Index.

Die Parametrisierung des DynRH2Lev Verfahren war für diesen Anwendungszweck nicht ausreichend dokumentiert, diese musste daher erst ermittelt werden.

3.7 *Arbeitspaket 7 - Auswahl weiterer SE Verfahren*

In Summe wurden drei zeitgemäße Searchable Encryption Verfahren implementiert. Zusätzlich zum schon erwähnten In-Memory DynRH2Lev und der persistierenden Version („DynRH2LevRocks“) haben wir auf das ebenfalls Forward Sercure Verfahren „Sophos“ gesetzt.

Der Aufwand war hier größer, da Sophos bis dato nur in C++ implementiert war, das Verfahren musste daher erst nach Java portiert werden bevor es in das System

integriert werden konnte. Für die Datenpersistierung wurde ebenfalls RocksDB verwendet.

Die von Anfang an geplante Modularität des Systems hat sich hier sehr bewährt.

3.8 *Arbeitspaket 8 - Integration in Drittsoftware*

Für die Testintegration haben wir die Software „Cryptomator“ (https://cryptomator.org/) ausgewählt, welche mittels einer Stacked Filesystem Festplattenverschlüsselung Dokumente in einem beliebigen Cloudspeicherdienst (wie z.B. Dropbox oder Google Drive) verschlüsselt.

Unsere Clientlib wurde in den Desktopclient integriert (die Mobilversion von Cryptomator ist leider nicht als Open Source Software verfügbar).

3.9 *Arbeitspaket 9 – Hosting*

Es wurde ein Server mit schnellem Storage (SSDs) angeschafft, auf dem unsere Serverdienste laufen konnten und umfangreiche Performancetests durchgeführt werden konnten.

Der Server wird in einem Serverrack im Netzwerklabor der FH betrieben und kann auf die dort vorhandene Infrastruktur zurückgreifen. Als Betriebssystem wird die in Österreich entwickelte Open Source Hypervisor Software „ProxMox“ eingesetzt.

3.10 *Arbeitspaket 10 – Wartung*

Um ein friktionsfreies und einfaches Deployment in unserem Rechenzentrum und bei den Endanwendern zu erreichen, wurde konsequent auf Docker gesetzt. Für die einzelnen Komponenten (Gate, Backends) werden Docker Images gebaut, die einfach über docker-compose zum Laufen gebracht werden können.

3.11 *Arbeitspaket 11 - Integrationstests & Quantitative Analyse*

Das System und die implementierten Verfahren werden getestet, um sinnvolle Vergleiche anstellen zu können und die Grenzen des Systems auszuloten.

Es stellte sich heraus, dass die Beschränkung auf künstlich erstellte Testdaten nicht ausreichend ist und für sinnvolle Tests mit echten Daten operiert werden muss.

Es wurden zahlreiche Testdaten mit Hilfe von synthetischen und Real-World Testdaten erstellt und dann visualisiert.

Der verwendete Indexer (der die Suchworte aus den Dokumenten extrahiert) musste an die Struktur der verwendeten Daten angepasst werden (z.B. Datumserkennung, Filterung von Headern).

3.12 *Arbeitspaket 12 - Dokumentation erstellen*

Es wurden Dokumentationen für die unterschiedlichen Benutzergruppen erstellt.

Eine Beschreibung des Systems und der Architektur von Searchitect soll es zukünftigen Entwicklerinnen ermöglichen bzw. erleichtern das Projekt fortzuführen. Interessant wäre z.B. eine Integration weiterer SE Verfahren sowie eine Verbesserung der Usability sowohl Client als auch Serverseitig.

Die Anwenderdoku richtet sich an die Nutzer von Searchitect, die entweder selbst einen Server betreiben möchten oder mit einem Client auf eine vorhandene Instanz zugreifen möchten.

Es wurde darauf geachtet die Dokumentation nicht bis zum Ende aufzuheben sondern diese periodisch zu erstellen, etwa durch ausführliche README Dateien in allen Projekten und eine sinnvolle Codedokumentation.

Mit Hilfe von Swagger wurde weiters eine dynamische API Dokumentation automatisch erstellt.

3.13 *Arbeitspaket 13 – Paper*

Das erste wissenschaftliche Ergebnis des Projektes war die Fertigstellung der Masterarbeit „A Practical View on Dynamic Symmetric Searchable Encryption“ durch Ines Kramer in welcher das Framework und Messergebnisse beschrieben werden.

2018 konnte das Position Papers „Searchitect – A Developer Framework for Hybrid Searchable Encryption“, welches die Architektur von Searchitect beschreibt, auf der „3rd International Conference on Internet of Things, Big Data and Security (IoTBDS)“.

Im darauffolgenden Jahr haben wird das Paper „Search, Find and Resolve: Towards a Taxonomy for Searchable Encryption Schemes“ publiziert, in welches ebenfalls einige Erkenntnisse aus Searchitect eingeflossen sind („4th International Conference on Internet of Things, Big Data and Security (IoTBDS)“.

Eine abschließende Publikation mit allen Ergebnissen „Searchitect – Experimental evaluation of a forward secure dynamic symmetric searchable encryption framework“ wurde erstellt und zur Veröffentlichung auf einer Konferenz eingereicht, die endgültige Veröffentlichung wird daher im nächsten halben Jahr passieren.

3.14 *Arbeitspaket 14 – Öffentlichkeitsarbeit*

Folgende Tätigkeiten wurden durchgeführt um Searchitect bekannt zu machen:

- *Für einen Artikel zum Thema „Speichersysteme der Zukunft“ der Zeitschrift Heureka 7/2017 (Wissenschaftsbeilage des Falter) wurde Mathias Tausig interviewt.*

- *Über den Netidee Blog haben wir laufend über den aktuellen Stand des Projektes berichtet*
- Eine eigenen Webseite wurde erstellt und unter der angeschafften Domain searchitect.eu publiziert
- An den Open House Tagen an der FH Campus Wien wurde das Projekt mit Tisch und Poster den Besucherinnen vorgestellt
- Bei zwei Privacy fokusierten Veranstaltungen in Wien wurde das Projekt vor Fachpublikum vorgestellt: Auf der „Privacyweek“ 2018 sowie auf dem „Easterhegg“ im Frühjahr 2019. Bei beiden Veranstaltungen ergaben sich in Folge einige Gespräche mit an der Technologie Interessierten, die aber leider nicht in eine unmittelbare Kooperation mündeten

3.15 *Arbeitspaket 15 – Projektmanagement*

Die für die Verwaltung des Projektes notwendigen Tätigkeiten wurden durchgeführt:

- Anstellung von Ines Kramer
- Abrufung der Förderraten
- Laufende Pflege des Projektcontrollings
- Abgabe des Zwischenberichtes
- Verfassen von FH-internen Berichten

4 Liste Projektendergebnisse

Kurzbeschreibung der erreichten Projektendergebnisse jeweils mit Open Source Lizenz und Webadresse (netidee Vorgaben beachten!)

1	**Projektzwischenbericht**	CC-BY-SA 3.0 AT	https://www.netidee.at/searchitect
2	**Projektendbericht**	CC-BY-SA 3.0 AT	https://www.netidee.at/searchitect
3	Entwickler-DOKUMENTATION des Projektergebnisses für andere Entwickler ("Dritte"), die das Projektergebnis nach Projektende nutzen/weiterentwickeln wollen	CC-BY-SA 3.0 AT	https://www.searchitect.eu/documentation
4	Anwender-DOKUMENTATION des Projektergebnis für Anwender, die das Projektergebnis nach	CC-BY-SA	https://www.searchitect.eu/

	Projektende nutzen wollen	3.0 AT	documentation
5	Veröffentlichungsfähiger Einseiter * Kurzfassung WAS | FÜR WEN | WIE * Liste Projektergebnisse -> also diese Liste, ggf. kompromiert * mit Angabe Open Source Lizenz/Webadresse * wo finden Dritte die Projektergebnisse (inkl. Nutzerdokumentation Anwender bzw. Entwickler) * mögliche Weiterentwicklungen/ weitere Einsatz-/ Nutzungsmöglichkeiten	CC-BY-SA 3.0 AT	https://www.searchitect.eu/about
6	Dokumentation Externkommunikation zur Erreichung Sichtbarkeit /Nachhaltigkeit (eigenes Arbeitspaket vorsehen!) * Welche Maßnahmen wurden in welchem Umfang gesetzt * Jeweils Bewertung Aufwand / Nutzen * Lessons Learned / Empfehlungen für andere Projekte	CC-BY-SA 3.0 AT	Siehe Endbericht
6	**Searchitect Framework** **Searchitect Gate** **Searchitect Client Library** **SE Plugins für Gate und Client Library** **CLI Client**	GNU GPL v3	gitlab.com/Searchitect
7	**„Cryptomate“** **Cryptomator mit demonstrativer Searchitect Integration**	GNU AGPL v3	gitlab.com/Searchitect
8	**Masterarbeit von Ines Kramer**	CC-BY	https://pub.fh-campuswien.ac.at/obvfcwhsacc/content/titleinfo/2844442
9	**Position Paper „Searchitect - A Developer Framework for Hybrid Searchable Encryption“**	CC-BY-NC-ND	http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0006789402910298
10	**Paper „Search, Find and Resolve: Towards a Taxonomy for Searchable Encryption Schemes„**	CC-BY-	https://www.scitepress.org/PublicationsDetail.aspx?

		NC-ND	ID=wbyLvk5eUkU=&t=1
11	**Präsentation Privacy Week 2018**	CC-BY-SA	https://2018.privacyweek.at/pw18/talk/MADNEA/
12	**Präsentation Easterhegg 2019**	CC-BY-SA	https://conference.c3w.at/eh19/talk/9ZTMTZ/

5 Verwertung der Projektergebnisse in der Praxis

Angaben zur Verwertung der Projektergebnisse in der Praxis

Eine konkrete Verwendung in einem praktischen Umfeld konnte während der Projektlaufzeit leider nicht erreicht werden.

Unsere Befürchtung ist, dass der Stand der Verfahren leider noch nicht so weit ist wie wir ursprünglich erhofft hatten, was einen Praxiseinsatz erschwert. Insbesondere die Update Möglichkeit des Suchindex ist noch verbesserungsbedürftig. So müsste z.B. eine Logik, die zwischen veränderten und nicht veränderten Dokumenten unterscheidet, Client-seitig implementiert werden.

6 Öffentlichkeitsarbeit/ Vernetzung

Beschreibung der im Rahmen Ihres netidee-Projektes bereits erfolgten bzw. noch geplanten Öffentlichkeitsarbeit oder Vernetzung

Unsere öffentlichen Vorträge in Wien auf der Privacyweek 2018 und dem Easterhegg 2019 waren durchaus erfolgreich. Die Zahl der Zuhörenden war mit ca. 25 und ca. 50 für ein derart spezielles Thema durchaus zufriedenstellend.

Wir konnten im Anschluss daran auch mit jeweils 2-3 Personen, die die Technologie „Searchable Encryption“ bis dato noch nicht kannten, interessante Gespräche führen über einen möglichen Einbau unserer Technologie in ihre Produkte. Es blieb aber leider bis dato bei diesen losen Kontakten und kam noch nicht zu einer konkreteren Kooperation.

7 Geplante Aktivitäten nach netidee-Projektende

Sind weiterführende Aktivitäten nach dem netidee-Projektende geplant?

Das schon erwähnte dritte Paper ist fertiggestellt und aktuell in einem Peer Review Prozess einer Konferenz. Dessen Publikation & Präsentation wird auf jedenfall zeitnah erfolgen.

Darüberhinaus sind weitere Aktivitäten im Rahmen von Studierendenarbeiten angedacht aber noch nicht konkret geplant, da eine Beteiligung weiterer Studierender vor dem offiziellen Projektabschluss uns nicht sinnvoll erschien. Im kommenden Studienjahr ist aber eine Vergabe von Wahlfachprojekten oder Abschlussarbeiten die das System weiterentwickeln, insbesondere im Usability Bereich, durchaus denkbar und angedacht.

8 Anregungen für Weiterentwicklungen durch Dritte

Welche Nutzungs- und Weiterentwicklungsmöglichkeiten für Dritte ergeben sich durch Ihr netidee-Projekt bzw. empfehlen Sie?

Die folgenden Punkte wären sinnvolle Verbesserungen, die in einem nächsten Schritt angegangen werden könnten:

- Eine Userverwaltung für das Gate (eingeschränkte Registrierungsmöglichkeiten, Verwaltung durch einen Administratoraccount, Änderungsmöglichkeiten an den Accounts)
- Ein Webinterface bzw. Dashboard mit dem ein Searchitect Server komfortabl administriert und überwacht werden kann.
- Sichere Abspeicherung des Client-States, bevorzugt Server-seitig