A=SIT Technology
E-Government Innovationszentrum

25. August 2020 Studie · Mobile Sicherheit · 14 Min. Lesezeit

Privacy-Preserving Service Composition

Der Volltext liegt derzeit nur in der Fassung „English“ vor. Die Übersetzung folgt.

Archiviert. Dieser Inhalt wird nicht mehr aktualisiert. Er bleibt zur Information verfügbar.

Stand 2020-08-13
Quelle https://technology2.a-sit.at/veroeffentlichungen/privacy-preserving-service-composition/

1. Einleitung

Eine Service-Komposition ist die Kombination mehrerer einzelner Dienste, um einen komplexeren Dienst zu erzielen. Diese einzelnen Services können vom öffentlichen oder privaten Sektor bereitgestellt werden. Die Vorteile der Service-Kompositionen ergeben sich aus der Modularität der Services: Einzelne Services, die in verschiedenen Anwendungsfällen wiederverwendet werden können, wodurch die Komplexität beim Erstellen neuer Prozessabläufe oder beim Anpassen vorhandener Prozesse verringert wird. Darüber hinaus schreiben einige Anwendungsfälle vor, dass mehrere Dienste integriert werden müssen, z. B. wenn die beteiligten Daten von verschiedenen Organisationen verwaltet werden. Unsere Arbeit konzentriert sich auf Service-Orchestrierungen, bei denen eine Middleware, Anforderungen an Sub-Services innerhalb des Prozessflusses koordiniert. Das Weiterleiten der Service-Anfragen über eine Middleware wirft Datenschutzbedenken auf, da diese den Dateninhalt der Anforderungen lesen oder daraus lernen können. Darüber hinaus benötigen in vielen Fällen nicht alle Akteure, die im Prozessfluss eines zusammengesetzten Services involviert sind, Zugriff auf den gesamten Datensatz einer initial übertragenen Service-Anforderung, da sie nur bestimmte Datensätze benötigen.

Schließlich stellt die Ende-zu-Ende-Integrität im Allgemeinen eine Herausforderung bei Service-Kompositionen dar, da Sub-Services innerhalb des Prozessflusses oder die Middleware in Service-Orchestrierungen selbst Nachrichtenteile verändern können. Eine vollständig vertrauenswürdige Middleware kann Datenschutzrichtlinien bewerten und die erforderlichen Attribute nur selektiv an die verschiedenen Dienste weitergeben (vgl. Ma et al. [1]). Dies ermöglicht eine differenzierte Zugriffskontrolle durch die Middleware und verbessert somit die Privatsphäre der Benutzer in Bezug auf Informationen, die den Unterdiensten ausgesetzt sind. Dieser Ansatz erfordert jedoch volles Vertrauen in eine zentrale Partei (d.h. die Middleware), die weiterhin die (sensiblen) Daten der Benutzer lesen kann. Alternative Arbeiten zielen darauf ab, das Risiko einer schädlichen Middleware zu verringern, indem durch Anwendung von Public-Key Kryptografie Ende-zu-Ende-Integrität erreicht wird (vgl. Singaravelu und Pu [2]). Bei diesem

Ansatz verschlüsselt ein Absender Nachrichtenfelder für jeden Service. Nachteile dieses Konzepts sind das erforderliche Wissen über Empfänger und ihre öffentlichen Schlüssel, sowie eine schlechte Effizienz, eine hohe Anzahl erforderlicher Schlüssel und der Nachrichtenaufwand durch mehrere Verschlüsselungen desselben Attributs für verschiedene Services.

1.1. Wissenschaftlicher Beitrag

In diesem Artikel schlagen wir eine Service-Kompositionsarchitektur vor, die die oben genannten Probleme löst, indem sie die Funktionen der Ciphertext-Policy Attribute-based Encryption (CP-ABE)

[3] in Kombination mit einer kollaborativen Schlüsselverwaltung sowie einer Outsourced Decryption nutzt [4]. Unsere Hauptbeiträge sind:

1. Unsere vorgeschlagene Architektur schützt die Privatsphäre der Benutzer, sowohl gegenüber den verschiedenen Sub-Services, als auch gegenüber der Middleware, indem fortschrittliche kryptografische Mechanismen (vgl. CP-ABE) in unserem definierten Service-Kompositionsprotokoll angewendet werden. Diese Ende-zu-Ende-Verschlüsselung reduziert die Vertrauensanforderungen an die Middleware, da nur verschlüsselte Daten verarbeitet werden.
2. Darüber hinaus wird die Flexibilität bei der Service-Zusammensetzung erhöht. Bei der attributbasierten Verschlüsselung muss der Absender nichts über die Identität (wie z.B. den öffentlichen Schlüssel) aller empfangenden Teilnehmer der Service Komposition wissen. Stattdessen verschlüsselt der Absender Nachrichtenteile für Attribute, während die Middleware Sub-Services kombinieren und ersetzen kann, die über ausreichende Entschlüsselungsrechte verfügen.
3. Die angewandte Kryptographie ermöglicht eine effiziente One-to-Many-Kommunikation, da die Entschlüsselungsanforderungen eines Nachrichtenteils von mehreren Empfängern erfüllt werden kann. Im Gegensatz zu anderen Ansätzen ist die erarbeitete Architektur im Sinne des Rechenaufwands für die Entschlüsselung sowie des Verschlüsselungsaufwandes (z.B. keine doppelte Verschlüsselung desselben Nachrichtenteils für verschiedene Empfänger) effizient.
4. Diese Arbeit unterstützt die Ende-zu-Ende-Integrität und führt zu erhöhter Verantwortlichkeit durch die Integration digitaler Signaturen in das Service Kompositionsprotokoll.
5. Für die Evaluierung unseres vorgeschlagenen Konzepts wurde eine Fallstudie erarbeitet, die einen möglichen Anwendungsfall für ein beispielhaftes E-Health-Szenario beschreibt. In dieser Fallstudie definieren wir die Akteure und zeigen detailliert die Interaktionen sowie die Datenflüsse einschließlich kryptografischer Berechnungen. Darüber hinaus haben wir einen Proof-of-Concept implementiert, der nicht nur die Machbarkeit unseres Konzepts demonstriert, sondern auch dessen Effizienz in realen Szenarien hervorhebt.

2. Angewandte Kryptographie

In diesem Projekt wenden wir eine ausgelagerte Verschlüsselung in Kombination mit einem kollaborativem Schlüsselmanagement für Attribut-basierte Verschlüsselung an.

Sahai und Waters [11] führten die Attributbasierte Verschlüsselung (ABE) ein, die einen Verschlüsselungsmechanismus darstellt, der über herkömmliche Verschlüsselungsvorgänge mit öffentlichen Schlüsseln (wie bei der Public-Key Verschlüsselung) mit einem Schlüssel hinausgeht. Anstatt den öffentlichen Schlüssel eines Empfängers zum Verschlüsseln einer Nachricht zu verwenden, muss ein

Absender nur die Attribute des Empfängers in einem ABE-System kennen. Auf diese Weise kann der Absender eine Nachricht verschlüsseln, die nur von Empfängern entschlüsselt werden kann, die die angegebenen Attribute besitzen.

In CP-ABE (Ciphertext-Policy Attribute-Based Encryption) [3] definiert der Sender die Zugriffsstruktur (Kombination aus Attributen). Eine vertrauenswürdige Partei (Key Authority) weist einem Client eine Reihe von Attributen zu, wenn das Schlüsselmaterial übermittelt wird. Da in CP-ABE nur der Client Zugriffsstrukturen festlegen kann, verwenden wir CP-ABE für unser Konzept.

Darüber hinaus verwenden wir ein kollaboratives Schlüsselverwaltungsprotokoll für die Attributbasierte Verschlüsselung von Ciphertext-Policy (CKM-CP-ABE), das von Lin et al. [4] vorgestellt wurde. In diesem Modell ist der private Schlüssel eines Akteurs in drei Schlüsselteile unterteilt: 1) für die Schlüsselautorität (KA), 2) für einen Neuverschlüsselungsserver (Reencryption Server) und 3) für den Client (CL).

Unter Verwendung dieser Schlüsselteile können die Schlüsselautorität und der Reencryption Server einen Chiffretext für den Client neu verschlüsseln und teilweise entschlüsseln, wodurch die rechenintensivsten Berechnungen durchgeführt werden. Schließlich wendet der Client seinen privaten Schlüsselteil an, um den partiell-entschlüsselten Chiffretext zu entschlüsseln, was weniger Rechenaufwand erfordert.

Das kollaborative Schlüsselverwaltungsprotokoll für CP-ABE (CKM-CP-ABE) besteht aus den folgenden Algorithmen:

BaseSetup(1 [k] ,S,) → (params) : Der Setup-Algorithmus nimmt einen Sicherheitsparameter und ein Attribut-Set S als Input. Die Ausgabe dieser Funktion bezeichnet die öffentlichen Parameter, die eine Beschreibung des Attributuniversums U enthalten. **KeyCreation(S) → (IK [KA])** : Der Schlüsselerzeugungsalgorithmus nimmt den geheimen Hauptschlüssel MSK sowie einen Attributsatz S als Eingabe und gibt einen privaten Schlüssel PK aus, der dem gegebenen Attributsatz S entspricht. Der erzeugte private Schlüssel kann zum Entschlüsseln eines Chiffretextes verwendet werden, wenn sein entsprechender Attributsatz die Zugriffsstruktur der Verschlüsselung erfüllen kann. **Encrypt(A, m) → (CT [init])** : Der Algorithmus zur Verschlüsselung einer Nachricht m nimmt eine Zugriffsstruktur A (definiert durch Kombination aus Attributen) als Eingabe. Der Verschlüsselungsalgorithmus gibt einen initial verschlüsselten Text aus.

ReEncrypt(CT [init]) → (CT [re]) : Dieser Algorithmus generiert einen neu verschlüsselten Ciphertext und nimmt einen initial verschlüsselten Ciphertext als Input. **PartialDecrypt(CT [re] , PK [KA,] PK [RS]) → (CT [ulti])** : Dieser Algorithmus nimmt einen neu verschlüsselten Ciphertext, und den privaten Schlüssel der Key Authority und des Reencryption Servers entgegen und gibt einen partiell-entschlüsselten Ciphertext aus. **Decrypt(CT [ulti] , PK [CL]) → (m)** : Der Entschlüsselungsalgorithmus nimmt den partiell-entschlüsselten Ciphertext und den privaten Schlüssel des Client als Input und gibt die Klartextnachricht aus.

3. System Model

In diesem Abschnitt werden die beteiligten Akteure einschließlich ihrer zugehörigen Vertrauensannahmen sowie der Datenfluss eines zusammengesetzten Services in unserer Service-Kompositionsarchitektur beschrieben. Basierend auf diesem Vertrauensmodell leiten wir mehrere Sicherheitsziele ab, die als Motivation für unsere vorgeschlagene Kompositionsarchitektur und unser Protokoll dienen.

3.1. Akteure und Datenaustausch Die Systemarchitektur für unser Service-Kompositionssystem besteht aus den Akteuren, die in den folgenden Absätzen erläutert werden.

3.1.1. Client Der Client stellt ein Gerät dar, das von einem Benutzer verwendet wird, der einen zusammengesetzten Service nutzen möchte. Der Client muss nichts über Akteure wissen, die an einer Komposition teilnehmen, um Anforderung für eine Entschlüsselung der Daten zu definieren, da der Klartext für Attribute verschlüsselt wird. Daher ist es nicht erforderlich, die Empfänger zum Zeitpunkt der Nachrichtengenerierung zu kennen, was eine hochflexible Service Komposition im Backend ermöglicht. Da Smartphones auf der Client-Seite berücksichtigt werden, müssen auch Geräte mit begrenzter Rechenleistung beachtet werden. Eine einmalige initiale Übertragung eines Datensatzes reduziert die Kommunikationsschritte in Service-Kompositionen. Daher müssen Clients in der Lage sein, die Privatsphäre ihrer Daten in einer One-to-Many-Kommunikation zu schützen.

3.1.2. Middleware Die Middleware besteht aus mehreren Komponenten, die die Funktionalität bereitstellen, die zum Koordinieren der Service-Aufrufe und zum Behandeln der kryptografischen Mechanismen erforderlich ist. Da die Koordination von Service-Aufrufen von der Middleware durchgeführt wird, fungiert sie als Service Registry und definiert die Arbeitsabläufe von zusammengesetzten Services. Daher bietet die Middleware auch die Funktionalität zum Registrieren von Services der Service Provider im Kompositionsnetzwerk durch Zusammenarbeit mit einer Schlüsselautorität (Key-Authority). Die Middleware empfängt nur verschlüsselte Nachrichten.

3.1.3. Service Provider Service Provider sind Akteure, die einen oder mehrere Dienste bereitstellen, die für Service-Kompositionen verwendet werden können. Jeder Service Provider muss eine Erstauthentifizierung bzw. Registrierung bei der Key-Authority durchführen, um am Service-Kompositionssystem teilnehmen zu können. Während dieser Authentifizierung kann ein Dienstanbieter eine Reihe gewünschter Attribute beantragen. Bei erfolgreicher Authentifizierung und Validierung von Berechtigungen werden diese Attribute dem jeweiligen Antragsteller zugewiesen. Um die Vertraulichkeit in unserem Systemmodell zu gewährleisten, dürfen Dienstanbieter keine Dateninhalte lesen können, die nicht für sie bestimmt sind. Da ein anfänglicher Datensatz von jedem Service Provider in einer Prozesssequenz geändert werden kann, muss außerdem jeder Akteur in einer Komposition in der Lage sein, zu überprüfen, welche Partei Daten geändert hat. Dies ist notwendig, um den Akteuren Verantwortlichkeiten für die Erbringung eines Services zuweisen zu können.

3.1.4. Key Authority Die Schlüsselautorität (Key Authority, KA) ist eine semi-vertrauenswürdige Komponente und für die Erstellung und Aktualisierung von Schlüsselmaterial, zusammen mit einem Neuverschlüsselungsserver und einem Client verantwortlich. Darüber hinaus ist sie auch für die Registrierung neuer Akteure verantwortlich, die am Service Kompositionssystem teilnehmen möchten. Bei der Registrierung muss jeder Antragsteller ein asymmetrisches Schlüsselpaar generieren. Nach erfolgreicher Registrierung stellt die KA ein Zertifikat für den öffentlichen Schlüssel eines antragstellenden Akteurs aus, das auf einer PKI basiert. Die KA ist berechtigt, Attribute zu erteilen oder zu widerrufen und dem vorhandenen Attribut-Universum (Gesamtheit aller Attribute, die in einem System bestehen) Attribute hinzuzufügen. Darüber hinaus ist die KA für den Widerruf ausgestellter Schlüssel verantwortlich. In den meisten Konzepten für ABE [3], [4], [15] erfordert die KA volles Vertrauen, da sie jeden Entschlüsselungsschlüssel selbst generieren und daher jeden verschlüsselten Text entschlüsseln kann. Um den Schutz der ausgetauschten Daten in unserem Systemmodell zu gewährleisten, darf auch die KA keine Dateninhalte lesen können. Daher darf die KA keine Entschlüsselungsschlüssel selbst generieren können.

Zu diesem Zweck wenden wir die kollaborative Schlüsselverwaltung an, bei der die KA keine Entschlüsselungsschlüssel allein erstellen kann, sondern nur zusammen mit einem Entschlüsselungsserver und einem Client. Auf diese Weise kann die vollständige Privatsphäre der Daten auch gegenüber der KA gewährleistet werden, selbst wenn diese Komponente von einem Angreifer kompromittiert wird. Selbst wenn der Entschlüsselungsserver und die KA zusammenarbeiten, können sie einen verschlüsselten Text nicht vollständig entschlüsseln, da der Client ebenfalls involviert sein muss. Beim Erstellen eines Entschlüsselungsschlüssels erhalten weder die KA noch der Entschlüsselungsserver Zugriff auf den endgültigen Entschlüsselungsschlüssel, sondern tragen nur zu dessen Generierung bei.

3.2.1. Orchestration Engine Die Orchestration Engine (OE) koordiniert die Reihenfolge der Service-Aufrufe. Informationen zu verfügbaren Services werden in einer Service-Registry gesammelt. Die OE bestimmt den Prozessfluss eines zusammengesetzten Services. Da diese Middleware alle Daten empfängt, die

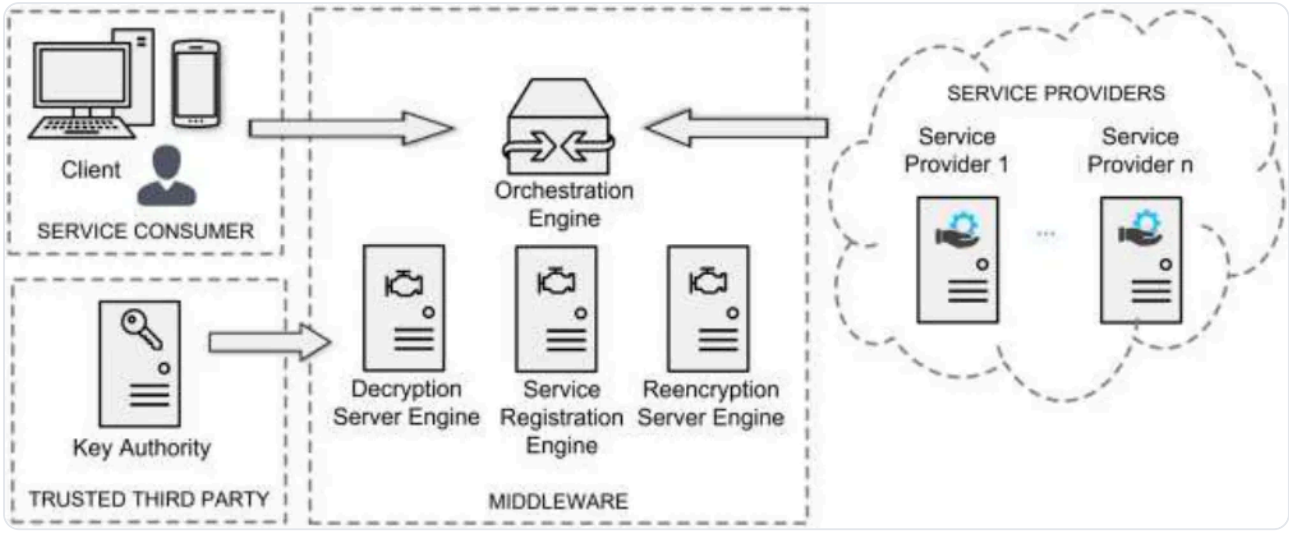


Abbildung 1: Systemarchitektur 3.2. Middleware Komponenten Die Middleware besteht aus den folgenden vier Komponenten.

von Akteuren eines zusammengesetzten Services übertragen werden, müssen diese Daten verschlüsselt werden.

3.2.2. Reencryption Server Engine Die RSE (Reencryption Server Engine) verschlüsselt den Ciphertext mithilfe eines Wiederverschlüsselungs-Algorithmus, der für die Entschlüsselung benötigt wird. Außerdem überträgt die RSE ihren Teil des privaten Schlüssels eines Benutzers an die Decryption Server-Engine, um eine teilweise Entschlüsselung zu ermöglichen. Da der RSE nur Ciphertext empfängt und weitergibt, ohne den Klartext zwischenzeitlich zu lesen, ist die Vertraulichkeit sensibler Daten gewährleistet.

3.2.3. Decryption Server Engine Die Decryption Server-Engine (DSE) bietet die Funktionalität zum teilweisen Entschlüsseln von Ciphertexten. Für die teilweise Entschlüsselung benötigt die DSE zwei Teile des privaten Schlüssels eines Benutzers: einen Teil der Key-Authority und einen anderen Teil der Reencryption Server-Engine. Es führt die rechenintensiven Operationen des Entschlüsselungsalgorithmus aus und gibt einen partiell-entschlüsselten Ciphertext aus. Dieser partiell-entschlüsselte Ciphertext erfordert zur Entschlüsselung auf Client-Seite weniger Rechenaufwand. Die DSE empfängt nur verschlüsselte Informationen und kann einen Ciphertext nicht vollständig alleine entschlüsseln.

3.2.4. Service Registration Engine Diese Service Registration Engine (SRE) ist erforderlich, um einen Registrierungsmechanismus für Service Provider bereitzustellen, die sich für die Teilnahme am Service Kompositionssystem bewerben. Die Key-Authority informiert die SRE nach erfolgreicher Registrierung eines Service Providers über die genehmigten Attribute und übermittelt das ausgestellte Zertifikat für das Signaturschlüsselmaterial. Die SRE validiert das Zertifikat und fügt die entsprechenden Dienste der Service Registry hinzu.

4. Privacy-Preserving Composition-Protocol

In diesem Abschnitt wird ein neuartiges Protokoll beschrieben, das Ende-zu-Ende-Verschlüsselung und Integrität bieten. In diesem Protokoll sendet ein Benutzer zunächst eine Service-Anfrage an die OE, die dies Service-Anfragen nacheinander an mehrere Service Provider (SP1, ..., SPn) weiterleitet, die die empfangenen Daten verarbeiten. Dieser Prozess wird fortgesetzt, bis die Service Komposition endgültig abgeschlossen ist und (optional) eine Antwort für den Benutzer generiert wird. Abbildung 2 beschreibt die Schritte des Protokolls als Iteration von einem Sender i zu einem Empfänger i + 1, der zum nächsten Sender wird und die Iteration erneut startet.

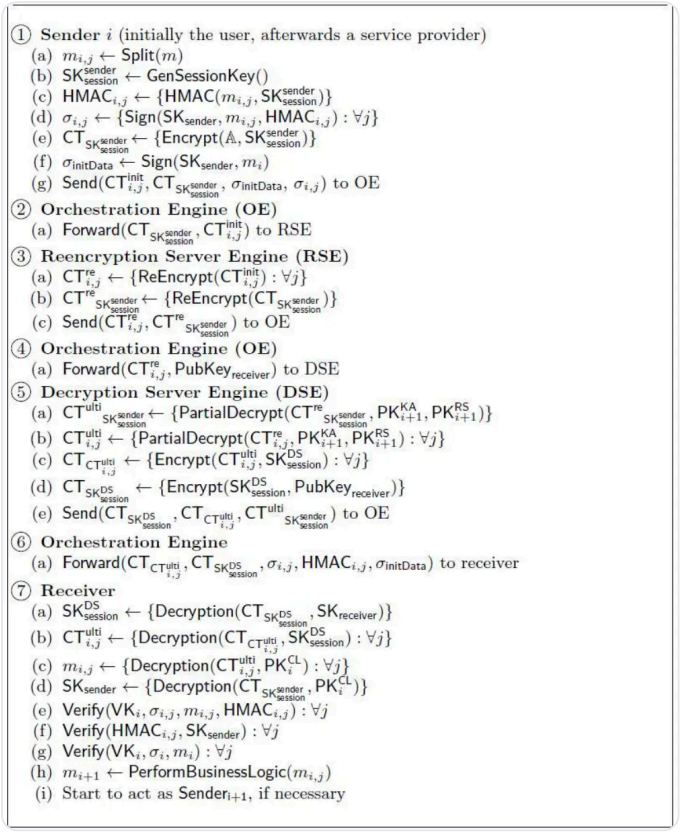


Abbildung 2: Privacy-Preserving Composition Protocol

5. Evaluation

In diesem Abschnitt wird eine Fallstudie für einen Telemedizin-Dienst dargestellt, eine Proof-of-Concept-Implementierung vorgestellt und verschiedene Aspekte erörtert.

5.1. Implementierung Wir haben den eingeführten Anwendungsfall mit Java-basierten Webdiensten sowie einer Android-App implementiert. Diese Akteure basieren auf dem von Lin et al. eingeführten kollaborativen Schlüsselverwaltungsprotokoll für CP-ABE. [4]. Wir haben dieses kryptografische Schema mithilfe der IAIK Java Cryptography Extension (IAIK-JCE) sowie der IAIK ECCelerate-Bibliothek für elliptische Kurven implementiert, die kryptografische Funktionen für bilineare Paarungen bietet. Für unsere Bewertung wählen wir einen Sicherheitsparameter von lamda = 256 Bit, der eine langfristige Sicherheit gemäß der Empfehlung von NIST bietet [18].

5.1.1. Nachrichtenformat Um die Nachrichten zu codieren, haben wir JSON Web Tokens (JWT) ausgewählt. Der Absender (d. H. Benutzer oder Dienstanbieter) strukturiert zuerst die Daten als JSON-Dokumente und fügt dann eine Signatur hinzu, die mit seinem privaten Signaturschlüssel generiert wurde. Wenn ein Akteur die Nutzdaten eines JWT ändert oder einen neuen Nachrichtenteil hinzufügt, muss er das Token erneut signieren, um eine Nachverfolgung zu ermöglichen, wer Daten bearbeitet hat. Schließlich werden alle JWTs mit einer gewünschten Zugriffsstruktur A verschlüsselt. Die Verwendung von JWT ermöglicht eine leichtgewichtige Kommunikation, die für unseren Anwendungsfall ein geeignetes Datenformat bietet.

5.1.2. Test-Setup Die Algorithmen für das kollaborative Schlüsselverwaltungsprotokoll werden von drei Webdiensten ausgeführt, die auf einem Lenovo Thinkpad T460s mit Intel ausgeführt werden i5-6200U Dual-Core-Prozessor und 12 GB RAM. Da wir auch clientseitig Smartphones in Betracht ziehen möchten, verwenden wir ein HTC U11 mit einem Qualcomm Snapdragon 835 Quad-Core-Prozessor mit jeweils 2,45 GHz und 4 GB RAM.

5.1.3. Performance des Use-Cases Das Smartphone startet den Prozess, indem es zunächst die beiden generierten JWT mit unterschiedlichen symmetrischen 256-Bit-AES-Sitzungsschlüsseln verschlüsselt. Anschließend wurden diese Schlüssel mit verschiedenen Zugriffsstrukturen verschlüsselt. Ein Web Service repräsentiert die Orchestrierungs-Middleware und übernimmt die Koordination der Service-Aufrufe und führt den Reencryption-Algorithmus durch. Der zweite Web Service führt den Algorithmus zur partiellen Entschlüsselung aus, der normalerweise von einem leistungsstarken Entschlüsselungsserver ausgeführt wird. Um zuverlässige Ergebnisse zu erzielen, haben wir den Test 50 Mal durchgeführt und den Median dieser Leistungswerte verwendet. Die Testergebnisse zeigen, dass unsere Auswahl an kryptografischen Mechanismen eine praktisch effiziente Ausführungszeit für Service Kompositionen bietet. Insbesondere die erforderliche Ausführungszeit für die Entschlüsselung auf der Client-Seite zeigt einen perfekt geeigneten Ansatz für Geräte mit begrenzter Rechenleistung wie Smartphones. Die Ausführungszeit der partiellen Entschlüsselung kann weiter verkürzt werden, indem anstelle des Lenovo Thinkpad T460s leistungsstarke Hardware für den Entschlüsselungsserver verwendet wird. In diesem Leistungstest haben wir die Leistung gut überprüfter Algorithmen wie die Generierung von HMACs und Sitzungsschlüsseln oder die Ausführungszeit der Überprüfung der asymmetrischen Signatur nach dem neuesten Stand der Technik nicht untersucht, da wir uns auf die Leistung von CKM-CP-ABE konzentriert haben.

5.1.4. Generelle Performance von CKM-CP-ABE Nach Ambrosin et al. [17] umfasst ein vernünftiges Szenario in ABE etwa 30 Attribute, was zu hochkomplexen Zugriffsstrukturen führen kann. Ziegler et al. [14] führten eine umfassende Bewertung von CP-ABE mit ausgelagerten Entschlüsselung durch. Sie untersuchen die Ausführungszeiten für eine unterschiedliche Anzahl von Attributen sowie die Auswirkungen unterschiedlicher Sicherheitsstufen mit Sicherheitsparametern von Lambda = 80 Bit bis Lambda = 512 Bit. Ihre Ergebnisse zeigen, dass die Anwendung von CKM-CP-ABE auf ein Service-Orchestrierungssystem eine praktische Effizienz ermöglicht.

6. Conclusion

In diesem Projekt haben wir eine Architektur und ein Protokoll für eine effiziente, flexible und privatsphäre-schützende Service Komposition erarbeitet. Das vorgeschlagene Konzept ermöglicht eine Ende-zu-Ende-Verschlüsselung und fein abgestimmte Zugriffskontrollen in einer One-to-Many-Kommunikation, indem eine attribut-basierende Verschlüsselung mit einer ausgelagerten Entschlüsselung und einem kollaborativen Schlüsselverwaltungsprotokoll kombiniert wird. Eine solche attributbasierte Verschlüsselung verbessert auch die Flexibilität, da Absender die endgültigen Empfänger nicht kennen müssen, wodurch Middlewares die Services innerhalb des Prozessflusses bei Bedarf ändern oder ersetzen können. Während eine sichere One-to-Many-Kommunikation den Übertragungsaufwand verringert, ist für Clients (z. B. Smartphones) nur ein geringer Rechenaufwand erforderlich, da durch ausgelagerte Entschlüsselung große Teile des Entschlüsselungsaufwands an leistungsstarke Server ausgelagert werden können. Darüber hinaus bietet unser Protokoll eine Ende-zu-Ende-Integrität für ausgetauschte Nachrichten, wodurch Änderungen an den Nachrichtenteilen nachverfolgt werden können und somit Verantwortlichkeit gewährleistet ist.

Referenzen

[1] [1. Z. Ma, J. Manglery, C. Wagner, and T. Bleier: Enhance Data Privacy in Service] Compositions through a Privacy Proxy. In: Sixth International Conference on Availability, Reliability and Security. pp. 615-{620. https://doi.org/10.1109/ARES.2011.94 (2011)

[2] L. Singaravelu and C. Pu: Fine-Grain, End-to-End Security for Web Service Compositions. In: IEEE International Conference on Services Computing (SCC 2007). pp. 212-{219. https://doi.org/10.1109/SCC.2007.61 (2007)

[3] J. Bethencourt, A. Sahai, and B. Waters: Ciphertext-Policy Attribute-Based Encryption. In: IEEE Symposium on Security and Privacy (SP '07). pp. 321-{334. https://doi.org/10.1109/SP.2007.11, (2007)

[4] G. Lin, H. Hong, and Z. Sun: A Collaborative Key Management Protocol in Ciphertext Policy Attribute-Based Encryption for Cloud Data Sharing. In: IEEE Access 5 (2017), pp. 9464|9475. https://doi.org/10.1109/ACCESS.2017.2707126 (2017)

[11] Amit Sahai and Brent Waters: Fuzzy Identity-Based Encryption. In: Advances in Cryptology { EUROCRYPT 2005, Springer Berlin Heidelberg, pp. 457|473. (2005)

[14] Dominik Ziegler, Josef Sabongui, and Gerald Pal_nger: Fine- Grained Access Control in Industrial Internet of Things - Evaluating Outsourced Attribute-Based Encryption. In: ICT Systems Security and Privacy Protection - 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, https://doi.org/10.1007/978-3-030-22312-0 7 (2019)

[15] N. Oualha and K. T. Nguyen: Lightweight Attribute-Based Encryption for the Internet of Things. In: 2016 25th International Conference on Computer Communication and Networks (ICCCN). pp. 1|6. https://doi.org/10.1109/ICCCN.2016.7568538 (2016)

[17] M. Ambrosin, A. Anzanpour, M. Conti, T. Dargahi, S. R. Moosavi, A. M. Rahmani, and P. Liljeberg: On the Feasibility of Attribute-Based Encryption on Internet of Things Devices. In: IEEE Micro 36, pp. 25-{35. https://doi.org/10.1109/MM.2016.101 (Nov 2016)

[18] National Institute of Standards & Technology: Recommendation for Key Management, Part 1: General (Rev 4). SP 800-57, (2016)