Zentrum für sichere Informationstechnologie – Austria
Secure Information Technology Center – Austria

A-SIT

Seidlgasse 22 / 9, 1030 Wien
Tel.: (+43 1) 503 19 63–0
Fax: (+43 1) 503 19 63–66

Inffeldgasse 16a, 8010 Graz
Tel.: (+43 316) 873-5514
Fax: (+43 316) 873-5520

http://www.a-sit.at
E-Mail: office@a-sit.at

DVR: 1035461
ZVR: 948166612
UID: ATU60778947

POLICY-ENFORCEMENT IN VERTEILTEN UMGEBUNGEN

Version 1.1 vom 06.05.2020

Bernd Prünster – bernd.pruenster@a-sit.at

Abstract/Zusammenfassung: In Szenarien, in denen mehrere Teilnehmer mit unterschiedlichen Eigenschaften auf einer gemeinsamen Datenbasis arbeiten, ergibt sich besonders ohne zentrale Kontrollinstanz der Bedarf, automatisiert Zugriffsrechte und -regeln durchzusetzen. Ziel dieses Projekts ist die Entwicklung von Konzepten zur automatisierten Durchsetzung von Zugriffs-Policies in verteilten Systemen. Hierfür soll keine zentrale Instanz verantwortlich sein; stattdessen sollen Teilnehmerinnen und Teilnehmer autonom dafür sorgen, dass Empfängerinnen und Empfänger von ausgehenden Daten nur explizit erlaubte Operationen auf diesen Daten ausführen können, bzw. soll zumindest detektierbar sein, wenn sich eine Teilnehmerin oder ein Teilnehmer nicht korrekt verhält. Ebenso soll mit möglichst hoher Sicherheit garantiert sein, dass Policies korrekt ausgewertet werden, und eventuelles Fehlverhalten im Rahmen der Policy-Auswertung soll feststellbar und in einem gewissen Maß tolerierbar sein, ohne dass es zu Zugriffsverletzungen kommt.

Im Rahmen dieses Projekts wurde ein Konzept zu dezentralen Durchsetzung von Zugriffs-Policies entwickelt, sodass Daten verschlüsselt ausgelagert werden können und Zugriffskontrollen implementiert werden können, ohne dass der Ersteller, bzw. die Erstellerin dieser Daten in den Zugriffsprozess involviert sein muss. Die Grundlagen hierfür bilden ein Peer-to-Peer-Netzwerk und Secret-Sharing. Darüber hinaus wurde das entwickelte Konzept auch im Hinblick auf aktuelle Möglichkeiten im Bereich Trsuted-Computing positioniert. Das vorgestellte System ist nicht an einen bestimmten Anwendungsfall gebunden und dadurch sowohl in verteilten Systemen mit gemeinsamer Datenbasis, als auch beispielsweise im Kontext traditioneller Cloudspeicherdienste einsetzbar.

© A-SIT, Partner für Sicherheitsfragen

Inhaltsverzeichnis

Inhaltsverzeichnis 2
1. Einleitung 2
2. Hintergrund und Vorarbeiten 2
2.1. Attribute-Based-Encryption 2
2.2. Trusted Computing 3
3. Systemanforderungen 4
3.1. Aktoren 4
3.1.1. Ersteller 4
3.1.2. Datenspeicher 4
3.1.3. Evaluator 4
3.1.4. Empfänger 4
3.2. Geforderte Systemeigenschaften 5
3.2.1. Dezentralisierung 5
3.2.2. Datensicherheit 5
3.2.3. Nachvollziehbarkeit/Accountability 5
4. Asynchroner Datenaustausch 5
4.1. Verschlüsselung von Daten 5
4.2. Entschlüsselung von Daten 7
4.3. Policy-Definition und Attributprovider 9
5. Sicherheitseigenschaften 10
6. Fazit 10
Referenzen 11

1. Einleitung

Innerhalb von klassischen Client-Server-Szenarien lassen sich Zugriffskontrollen konzeptionell relativ einfach umsetzen. Sollen, wie im Rahmen dieses Projekts, Clients in unterschiedliche Klassen, mit unterschiedlichen Berechtigungen bezüglich Datenverarbeitung, eingeteilt werden und entsprechende Zugriffs-Policies durchgesetzt werden, lässt sich dies prinzipiell auf zwei Arten umsetzen: Entweder Zugriffs-Policies werden trivial bei jedem externen Zugriff direkt vom Ersteller, bzw. der Erstellerin von Daten ausgewertet, oder diese Aufgabe wird an Dritte ausgelagert. Im zweiten Fall stellen sich Fragen des Vertrauens in Dritte. Hier bieten sich je nach Vertrauensmodell entweder *Attribute-Based Encryption* (ABE) oder Trusted-Computing-Verfahren an. Im Rahmen dieses Projekt wird ein nicht zentral organisiertes System angestrebt. Gleichzeitig soll asynchroner Datenaustausch umgesetzt werden. Das bedeutet, dass Ersteller und Empfänger von Daten nicht gleichzeitig online sein müssen, also keine direkte Verbindung zwischen beiden Parteien erforderlich sein soll, die Daten jedoch trotzdem durch eine Zugriffs-Policy geschützt werden sollen. Im Hinblick auf diese angestrebten Eigenschaften werden nachfolgend ABE-Konzepte, sowie Umsetzungsmöglichkeiten auf Trusted-Computing-Basis beschrieben. Anschließend wird das angestrebte Szenario detailliert beschrieben. Abschnitt 4 präsentiert das auf Basis dieser Anforderungen erarbeitete System für asynchronen Datenaustausch, dessen Sicherheits-eigenschaften in Abschnitt 5 diskutiert werden. Abschließend werden die Projektergebnisse in Abschnitt 6 zusammengefasst.

2. Hintergrund und Vorarbeiten

Dieser Abschnitt fasst die Eigenschaften existierender Technologien im Bezug zu diesem Projekt zusammen und diskutiert relevante Arbeiten, welche ein Teilen als Basis für das im Rahmen dieses Projektes entwickelte System herangezogen wurden. Konkret wird ein kompakter Überblick über Attribute-Based-Encryption gegeben und die Eigenschaften dieses Konzepts den Zielen dieses Projekts gegenübergestellt.
Weiters wird Trusted Computing als Alternative beleuchtet, insbesondere im Hinblick auf Umsetzungsmöglichkeiten, ohne dass vorab definierten Instanzen besondere Rechte zugeteilt werden müssen. Dieser Aspekt ist besonders in dezentral organisierten Systemen relevant.

2.1. Attribute-Based-Encryption

Attribute-Based Encryption (ABE) beschreibt ein Konzept, welches es ermöglicht, Daten mittels Policies kryptografisch zu schützen. Im Gegensatz zu schlichter *Attribute-Based Access Control*

(ABAC), ist nicht eine vertrauenswürdige Instanz für Auswertung und Anwendung von Polices zuständig; stattdessen kommen kryptografische Verfahren zum Einsatz [1] [2]: Im Rahmen einer Setup-Phase werden Systemparameter inklusive Attributen definiert. Attribute werden über passende Verfahren als kryptografische Schlüssel abgebildet, welche im Rahmen von Policies genau wie Attribute verknüpft werden können (typischerweise als Boole'sche Formeln). Die Verknüpfung von Attributen und kryptografischem Material wird von so genannten *Attribute Authorities* durchgeführt. Diese überprüfen, ob Systemteilnehmerinnen und Systemteilnehmer bestimmte Attribute (z.B. Alter, Freigabestufe, …) besitzen, generieren ausgehend davon Schlüsselmaterial und geben dieses an die jeweiligen Teilnehmerinnen und Teilnehmer aus. Folglich wird die Eigenschaft, dass jemand über ein Attribut verfügt, durch den Besitz von diesen Attributen entsprechendem Schlüsselmaterial abgebildet. Eine Konsequenz daraus ist, dass Attribute Authorities vertraut werden muss, insbesondere, da diese in der Lage sind, alle innerhalb des Systems verschlüsselten Daten zu entschlüsseln.
Abseits von diesen zentral organisierten Modellen wurden auch Versuche unternommen, ABE in einem verteilten Kontext umzusetzen. Hervorzuheben ist in diesem Zusammenhang *Multi-Authority-ABE*, dessen Alleinstellungsmerkmal die Möglichkeit ist, beliebig viele *Attribute Authorities* zu unterstützen, ohne, dass diese zentraler Kontrolle unterworfen sind [3]. Zwar gibt es dadurch keine Notwendigkeit mehr, dass alle im System verwendeten Attribute von einer zentrales Instanz verwaltet werden, die zu Grunde liegenden kryptografischen Verfahren erlauben es jedoch nicht, dass ein und dasselbe Attribut von mehreren Attribute Athorities verwaltet wird. Dadurch ergibt sich auch in diesem an sich dezentral organisierten Szenario die Situation, dass einzelne Instanzen die alleinige Kontrolle über bestimmte Attribute haben. In Folge dessen kann es zu einer Matchkonzentration kommen, die mit rein zentral organisieren System vergleichbar ist, wenn beispielsweise ein von einer bestimmten Attribute Authority verwaltetes Attribut intensiv zur Durchsetzung von Zugriffs-Policies genutzt wird.

2.2. Trusted Computing

Als Alternative zu rein auf kryptografischen Garantien basierenden Verfahren bietet sich Trusted Computing an. Bislang war dies jedoch insbesondere in einem dezentral organisierten Kontext nicht sinnvoll umsetzbar. Hauptgrund dafür waren mangelnde Möglichkeiten, dezentral ein Vertrauensverhältnis zwischen Parteien aufzubauen, die nicht unmittelbaren Kontakt zueinander haben. Dank Remote-Attestation-Möglichkeiten im Rahmen von Trusted-Computing-Architekturen, wie Intels *Software Guard Extensions* (SGX) [4], rückt dies jedoch in den Bereich des Möglichen. Allerdings erfordert SGX im Rahmen des Attestation-Prozesses zwingend die Kontaktaufnahme zu einem von Intel betriebenen Service, wodurch sich erneut eine Abhängigkeit zu einer zentralen Instanz ergibt.

Abseits des Desktop-Bereichs wurde im Rahmen eines vorangegangenen A-SIT-Projekts ein Remote-Attestation-Konzept auf Basis von Key-Attestation für Googles Mobilbetriebssystem Android entwickelt, welches vollkommen dezentrale Trusted-Computing-Konzepte ermöglicht [5]. Im Gegensatz zu SGX ist damit zu keinem Zeitpunkt die Kommunikation zu einem vom CPU- oder Gerätehersteller betriebenen Dienst notwendig. Dies wurde auch erfolgreich im Rahmen einer Peer-to-Peer-Netzwerkarchitektur demonstriert, welche die Identitäten aller Teilnehmer und Teilnehmerinnen an den Besitz eines Android-Smartphones koppelt [6]. Auf dieser Basis lässt sich durch den Einsatz von Trusted Computing einerseits Fehlverhalten a priori nahezu ausschließen, andererseits können sich fehlerhaft verhaltende Teilnehmerinnen und Teilnehmer durch die Bindung an ein diskretes Gerät effektiv ausgeschlossen werden. Im Gegensatz zu anderen Lösungen entsteht durch Fehlverhalten und zwingende Kopplung an Hardware (in diesem Fall ein Smartphone) einer sich fehlerhaft verhaltenden Instanz direkt quantifizierbarer Schaden, da ein neues Gerät beschafft werden muss, sofern eine neuerliche Teilnahme am System möglich sein soll.

Insbesondere auf Grund der tatsächlich dezentralen Umsetzbarkeit wird im Rahmen dieses Projekts ein Konzept auf dieser Trusted-Computing-Basis vorgeschlagen, dass nicht rein auf kryptografischen Methoden zur Evaluierung von Policies setzt. Stattdessen kann jede Instanz für sich selbst entscheiden, welchen anderen Instanzen zur Durchsetzung von Policies vertraut wird. Demonstriert wird dieses Verfahren im Rahmen von asynchronem Datenaustausch (siehe nächster Abschnitt). Im Speziellen die umrissenen Trusted-Computing-Verfahren auf Android-Basis bieten

hier den Vorteil, dass keine komplexen Protokolle für die Attestierung eingesetzt werden müssen, sondern lediglich Zertifikatsketten, welche entsprechend ausgewertet werden, zum Einsatz kommen.

3. Systemanforderungen

Die grundlegende Anforderung, die vom vorgestellten Konzept erfüllt werden soll, ist eine Auslagerung von Daten an Dritte und der spätere Abruf der Daten von Empfängern, bzw. Empfängerinnen. Dabei soll es ebenso wie in traditionellen Cloud-Storage-Szenarien nicht notwendig sein, dass Datenersteller und –empfänger gleichzeitig online sind. Um Daten trotzdem mittels Zugriffs-Policies schützen zu können, ist eine weitere Partei notwendig, welche diese Policies evaluiert und sicherstellt, dass nur berechtigte Empfänger unter wohldefinierten Umständen Zugriff erhalten. Nachfolgend werden diese Aktoren definiert.

3.1. Aktoren

Ausgehend von der Grundforderung nach asynchronem, sicherem Datenaustausch ergeben sich die im Rahmen dieses Abschnitts definierten Aktoren. Die Sicht auf das System wurde bewusst allgemein gehalten.

3.1.1. Ersteller

Die Instanz des *Erstellers* verfügt über Daten, welche mittels einer von ihr frei definierbaren Zugriffs-Policy geschützt werden sollen. Jede Ersteller-Instanz wird über ein Public-Private-Keypair identifiziert.

3.1.2. Datenspeicher

Datenspeicher können benutzt werden, um Daten auszulagern, damit diese auch abgerufen werden können, wenn die Ersteller-Instanz offline ist. Um derart ausgelagerte Daten zu schützen, definieren *Ersteller* von Daten, Zugriffs-Policies, welche deklarieren, wer unter welchen Umständen auf ausgelagerte Daten zugreifen kann. Abgesehen von Availability werden keine unmittelbaren Anforderungen an Datenspeicher gestellt. Insbesondere soll kein umfassendes Vertrauensverhältnis notwendig sein, welches über das bloße zur Verfügung stellen von Speicherplatz hinausgeht. Daher gibt es auch keine speziellen Anforderungen an Identifizierbarkeit von Datenspeicher-Instanzen und es kann z.B. auf bestehende Cloudspeicherdienste zurückgegriffen werden.

3.1.3. Evaluator

Evaluatoren werden ebenso wie *Ersteller* an Hand eines Public-Private-Keypair identifiziert. *Ersteller* wählen aus ihnen bekannten Teilnehmerinnen und Teilnehmern des Systems diejenigen Instanzen als Evaluatoren aus, denen sie dahingehend vertrauen, Zugriffs-Policies korrekt auszuwerten. Zwar wird Trusted Computing auf Basis von Android-Key-Attestation als Basis des Vertrauensmodells vorgeschlagen, tatsächlich steht es Erstellern jedoch frei, beliebige Systemteilnehmerinnen und Systemteilnehmer als Evaluatoren auszuwählen, sofern diesen ausreichend vertraut wird.

3.1.4. Empfänger

Als *Empfänger* werden jene Systemteilnehmerinnen und Systemteilnehmer bezeichnet, welche auf Daten zugreifen möchten. Diese rufen Daten von *Datenspeichern* ab und kontaktieren die im Datencontainer definierten *Evaluatoren*, um Zugriff zu erhalten. Empfänger werden ebenfalls über ein Public-Private-Keypair identifiziert.

Nachfolgend werden die geforderten Eigenschaften des im Rahmen dieses Projekts entwickelten Systems definiert. Details zur Policy-Definition werden in Abschnitt 4.3 diskutiert.

3.2. Geforderte Systemeigenschaften

Wichtig sind im Rahmen des Systemmodells drei Eigenschaften: Allen voran soll es keine zentrale Instanz geben, welche etwaige Sonderbefugnisse hat oder besonderes Vertrauen genießt. Infolgedessen soll sich als zweite Kerneigenschaft ein hoher Grad an Flexibilität, insbesondere für die Ersteller von Daten, ergeben. Zusätzlich soll ein hohes Sicherheitsniveau erreicht werden. Ausgehend davon wurden die nachfolgend definierten Anforderungen an das System abgeleitet.

3.2.1. Dezentralisierung

Zu keinem Zeitpunkt soll die Funktionalität des Systems an eine zentrale Instanz gebunden sein.
Auf Grund dieser Anforderung fallen klassische Attribute-Based-Encryption-Systeme als mögliche Umsetzungsbasis weg.

3.2.2. Datensicherheit

Ausgetauschte Daten müssen in jedem Fall vor unerlaubtem Zugriff geschützt werden. Dies soll mit einer Kombination aus Zugriffs-Policies und Verschlüsselung erreicht werden. Ausgehend von Eigenschaft 3.2.1 sollen hierfür weder vordefinierte Vertrauensmodelle, noch Hierarchien (wie beispielsweise im PKI-Kontext) vorausgesetzt werden.

3.2.3. Nachvollziehbarkeit/Accountability

Insbesondere diejenigen Instanzen, welche für die Auswertung von Zugriffs-Policies verantwortlich sind, müssen sich korrekt verhalten. Eventuelles Fehlverhalten soll nachvollzogen und – soweit möglich – abgefangen werden können.

Ausgehend von diesen Anforderungen wurde ein Konzept zum asynchronen Datenaustausch ausgearbeitet. Dieses wird nachfolgend präsentiert.

4. Asynchroner Datenaustausch

Das im Rahmen dieses Projekts entwickelte System wurde als Peer-to-Peer-Netzwerk konzipiert. Grundlage für diese Entscheidung war einerseits der Dezentralisierungsanspruch, andererseits jedoch die schlichte technische Notwendigkeit, dass im Rahmen von Policy-Auswertungen direkt Daten zwischen Empfänger und Evaluator ausgetauscht werden müssen. Nachdem die kritischen Aktoren des Systems durch ein Public-Private-Keypair identifiziert werden, bieten sich entsprechende vorangegangene Arbeiten aus früheren A-SIT-Projekten zum Thema sichere Peer-to-Peer-Netze als Basis für die Umsetzung an [7]. In Kombination mit Remote-Attestation ergibt sich die Möglichkeit, Evaluatoren auf Trusted-Computing-Basis umzusetzen, wodurch korrekte Auswertung von Policies ermöglicht wird, ohne dass Evaluatoren zuvor im Rahmen von Enrolment-Prozeduren in das System eingebracht werden müssen. Um die Sicherheitsmarge weiter zu erhöhen, besteht die Möglichkeit, Policies von mehreren Evaluatoren simultan auswerten zu lassen, sodass das sich Ergebnis auf Basis eines Mehrheitsentscheids ergibt. Details hierzu werden nachfolgend beschreiben.

4.1. Verschlüsselung von Daten

Das Grundkonzept für sicheren, asynchronen Datenaustausch auf Basis von Zugriffs-Policies, ist die kryptografische Bindung zwischen Ersteller, Policy und Daten, sowie Secret-Sharing und die Möglichkeit Trusted Computing im Kontext von Policy-Auswertung einzusetzen. Ziel dieses Ansatzes ist, dass Daten und Policy nicht voneinander getrennt werden können, sowie dass die Urheberschaft der Daten zweifelsfrei nachgewiesen werden kann. Darüber hinaus soll für Empfänger nachvollziehbar sein, welche Instanz eine Policy ausgewertet hat, und – soweit möglich – soll auch Fehlverhalten detektierbar sein. Abbildung 1 skizziert die Erstellung eines Datencontainers inklusive Secret-Sharing. Dabei wird folgende Grundidee verfolgt: Daten werden verschlüsselt; der

verwendete Schlüssel wird auf Evaluatoren aufgeteilt, welche die Schlüssel-Shares nur Empfängern preisgeben, welche die an die Daten gekoppelte Policy erfüllen. Durch den Einsatz eines m-aus-n Secret-Sharing-Schemas können Mehrheitsentscheide umgesetzt werden. Insbesondere wird dadurch jedoch verhindert, dass Evaluatoren selbst Zugriff auf Daten erlangen können, auch wenn auf Trusted Computing verzichtet wird.

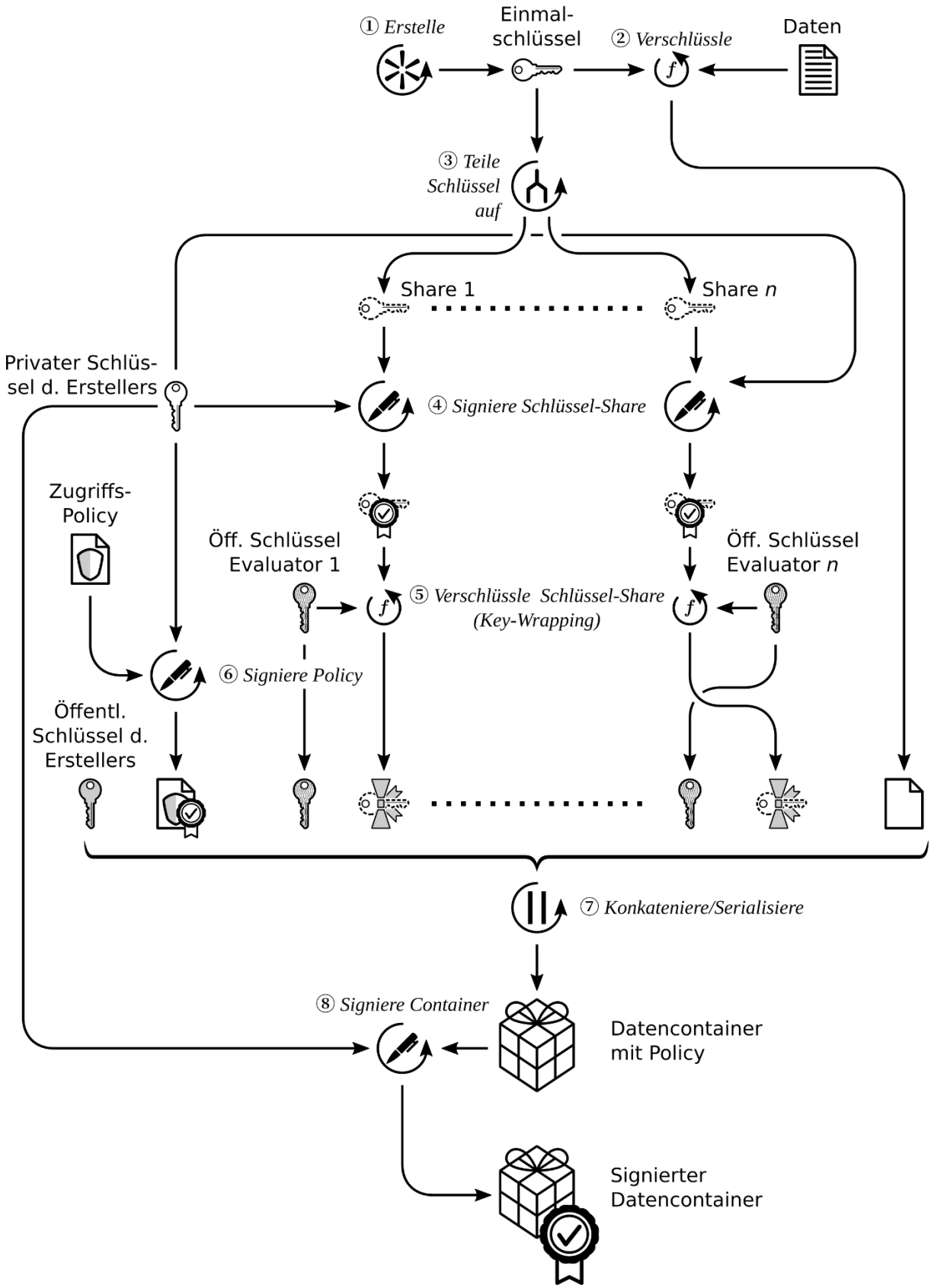

Abbildung 1: Erstellen eines verschlüsselten Datencontainers mit Policy

Im Detail wurde ausgehend von zu schützenden Daten und einer Zugriffs-Policy, welche beschreibt, unter welchen Umständen auf diese Daten zugegriffen werden darf, nachfolgend beschriebener Prozess zum Erstellen eines Datencontainers definiert.

⓪ Erstellen einer Nonce

① Erstellen eines Einmalschlüssels zur Verschlüsselung der zu schützenden Daten.

② Verschlüsselung der Daten mittels erstelltem Einmalschlüssel. Durch den Einsatz von Verschlüsselung können die Daten ab diesem Zeitpunkt öffentlich abgelegt werden.

③ Aufteilen des Einmalschlüssels mittels *m*-aus-*n*-Secret-Sharing in *n* Shares.

④ Verknüpfen der Nonce mit allen Shares, sowie signieren dieser Tupel. Dadurch wird eine kryptografische Bindung zwischen jedem Share und dessen Ersteller erzeugt. Zusätzlich können Empfänger nach Erhalt entschlüsselter Shares deren Integrität prüfen.

⑤ Verschlüsseln eines jeden zuvor signierten Shares unter dem öffentlichen Schlüssel jeweils eines Evaluators. Folglich können nur die in diesem Schritt ausgewählten Evaluatoren später vom Empfänger zur Policy-Auswertung, bzw. Freigabe der Shares herangezogen werden. Zumindest *m* aus *n* Evaluatoren müssen zum Zeitpunkt der Policy-Auswertung erreichbar sein.

⑥ Signieren von Nonce und Zugriffs-Policy unter dem privaten Schlüssel des Erstellers, um eine kryptografische Bindung zwischen Policy und Schlüssel-Shares (und dadurch auch zwischen Ersteller und verschlüsselten Daten) herzustellen.

⑦ Konkatenierung und Serialisierung von

- Öffentlichem Schlüssel des Erstellers
- Signierter Policy
- Öffentlichen Schlüsseln der Evaluatoren
- Verschlüsselten, signierten Shares
- Verschlüsselten Daten

⑧ Signieren des resultierenden Containers unter dem privaten Schlüssel des Erstellers

Die Eigenschaften dieses Datencontainers werden im nachfolgenden Abschnitt im Rahmen des Entschlüsselungsvorgangs diskutiert. In jedem Fall sind jedoch alle Komponenten kryptografisch miteinander verbunden, wodurch beispielsweise das Austauschen der Policy verhindert wird.

4.2. Entschlüsselung von Daten

Wie das im Rahmen dieses Projekts entwickelte Konzept seine Sicherheitsgarantien erreicht, zeigt sich beim Entschlüsselungsvorgang. Das zugehörige Sequenzdiagramm in Abbildung 2 veranschaulicht die nachfolgend beschriebene Entschlüsselung eines Datencontainers. Der Prozess selbst gestaltet sich wie folgt:

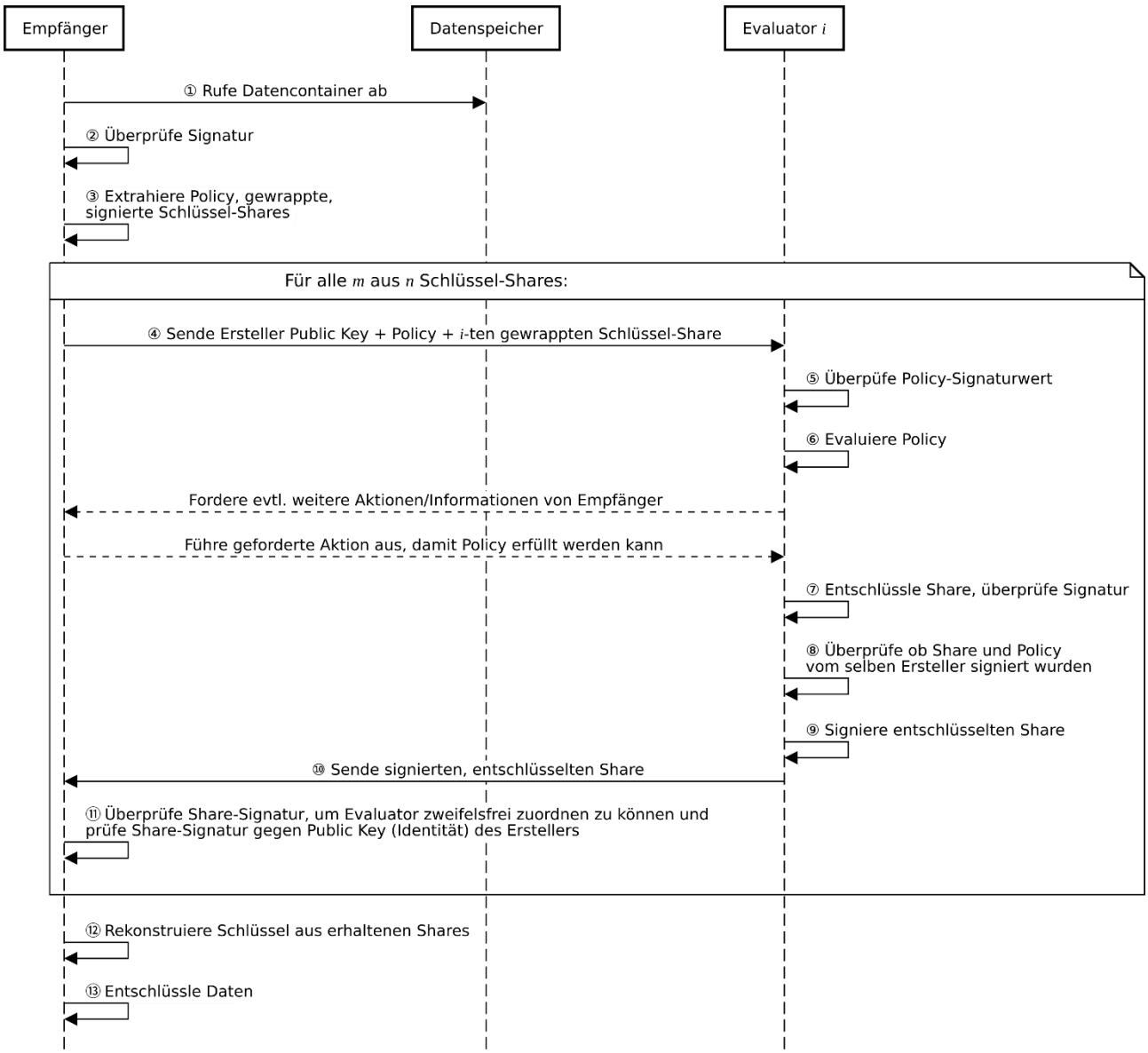

Abbildung 2: Entschlüsselungsvorgang

① Abrufen des Datencontainers vom Datenspeicher

② Überprüfen der Containersignatur. Da der öffentliche Schlüssel des Erstellers Teil des Containers ist, ist dies offline möglich, ohne den Ersteller kontaktieren zu müssen, wodurch die Urheberschaft eines Containers überprüft werden kann. Durch diese in sich abgeschlossene Datenstruktur ist eine grundlegende Voraussetzung für asynchrone Datenübertragung erfüllt.

③ Im nächsten Schritt werden die enthaltene Policy, sowie die Schlüssel-Shares und die zugehörigen öffentlichen Schlüssel der Evaluatoren extrahiert. Auf dieser Basis werden für *m* aus *n* Evaluatoren die folgenden Schritte durchgeführt:

④ Der Ersteller-Public-Key, die Policy, sowie der für die Evaluator-Instanz gewrappte Schlüssel-Share werden an die Evaluator-Instanz übermittelt.

⑤ Diese validiert die Signatur der Policy auf Basis des empfangenen Ersteller-Public-Keys.

⑥ Die Policy wird evaluiert. Eventuell notwendige Interaktionen mit der Empfänger-Instanz werden eingeleitet, um zu ermitteln, ob der Empfänger, bzw. die Empfängerin die vorliegende Policy erfüllt. Da der Empfänger zur Erfüllung der Policy diese im Wesentlichen kennen muss,

wurde auf eine Verschlüsselung von Policies verzichtet. Tatsächlich können jedoch auch nur für Evaluatoren zugänglich Informationen in Policies abgelegt werden.

⑦ Wenn ja, wird der erhaltene Schlüssel-Share entschlüsselt.

⑧ Nachfolgend wird die Signatur des entschlüsselten Shares validiert und überprüft, ob Policy und Schlüssel-Share (auf Basis von Nonce und Signaturen) zusammengehören.

⑨ Ist dies der Fall, signiert der Evaluator den entschlüsselten Share. Hierfür ist vom Evaluator der zu seiner Identität passende private Schlüssel einzusetzen. Sollte sich ein Evaluator wider Erwarten unehrlich verhalten und nicht den korrekten Share an den Empfänger übertragen wollen, kann durch diese Signatur auch nachträglich festgestellt werden, welcher Evaluator sich unehrlich verhalten hat.

⑩ Der signierte Share wird anschließend an den Empfänger übertragen.

⑪ Der empfangene Share wird gegen die im Container hinterlegte zugehörige Signatur geprüft, um sicherzustellen, dass sich ein Evaluator auch tatsächlich den korrekten Share liefert.

⑫ Der Empfänger rekonstruiert den Einmalschlüssel, der zum Entschlüsseln der im Container enthaltenen Daten notwendig ist.

⑬ Abschließend können die Daten vom Empfänger entschlüsselt werden.

Im Prinzip sind bei der Definition von Policies keine Einschränkungen vorgesehen. Allerdings stellt sich beispielsweise im Rahmen von Attributbasierten Policies die Frage nach Attributprovidern, welche auch Evaluatoren zugänglich sind. Details hierzu werden im nachfolgenden Abschnitt diskutiert.

4.3. Policy-Definition und Attributprovider

Ebenso wie die Wahl nach vertrauenswürdigen Evaluatoren liegt auch die Wahl nach geeigneten Attributprovidern im Rahmen einer Policy-Definition im Ermessen der Ersteller-Instanzen. Unabhängig davon stellt sich die grundsätzliche Frage nach der Umsetzbarkeit von Zugriffs-Policies abseits zentral definierter Attributprovider. Da innerhalb eines Systems das in diesem Bereich vollkommen offen konzeptioniert ist, per Definition keine vollständige Abhandlung dieser Thematik möglich ist, werden in diesem Abschnitt exemplarisch einige Umsetzungsmöglichkeiten beschrieben.

Grundsätzlich können allgemeine Eigenschaften der Betriebsumgebung (wie beispielsweise Datum und Uhrzeit) mit hinreichender Genauigkeit zur Policy-Definition herangezogen werden. Im Zweifel kommt ein Mehrheitsentscheid zum Tragen (siehe Abschnitt 4.1). Dadurch lassen sich beispielsweise ablaufende Daten umsetzen, welche nach einem bestimmten Zeitpunkt nicht mehr abrufbar, bzw. entschlüsselbar sind. Im Prinzip sind jedoch immer externe Attributprovider notwendig, um attributbasierte Zugriffskontrolle umzusetzen. Dabei kann jedoch auf beliebige bestehende Dienste, wie z.B. *OpenID Connect*[1] zugegriffen werden, wodurch sich semantisch ähnliche Szenarien wie im Rahmen von Multi Authority ABE (siehe Abschnitt 2.1) umsetzen lassen.
Ein entscheidender Mehrwert gegenüber ABE-Konzepten kann sich jedoch auch in diesem Kontext durch den Einsatz von Trusted Computing ergeben: Wenn laut Policy ebenfalls der Besitz eines aktuellen Android-Smartphones vorausgesetzt wird, welches umfassende Remote-Attestation-Features unterstützt, können Attribute auf Basis von Sensordaten, wie z.B. geografische Position abgefragt werden. Darüber hinaus kann über die Keystore- und Biometrie-APIs von Android vorausgesetzt werden, dass ein Nutzer, bzw. eine Nutzerin physisch mittels Fingerabdruck, Gesichts- oder Iris-Scan zur Interaktion gezwungen wird, wodurch sicher gestellt werden kann, dass die Empfänger-Instanz tatsächlich von einem Menschen (und nicht etwa einem Bot) bedient wird.

[1] https://openid.net/connect/

Generell erlaubt das vorgestellte System die Definition beliebig komplexer Policies, ohne dass dies nennenswerte Auswirkungen auf die Performance hat. Der Daten- und Rechenoverhead ist insgesamt gering, wodurch dich auch hier ein Vorteil gegenüber ABE ergibt.
Ein offenes Problem ist in diesem Zusammenhang die Möglichkeit durch ausreichend komplexe Policies, Denial-of-Service-Angriffe auf Evaluatoren auszuführen. Im Rahmen von Kryptowährungen, wie z.B. *Ethereum*[2] wurde dieses Problem gelöst, indem jeder Rechenschritt, der von Dritten ausgeführt werden soll – in diesem Zusammenhang im Rahmen von Smart Contracts – Kosten für den Auftraggeber verursacht. Im Kontext dieses Projekts wurde diese Problemstellung nicht als Teil des Projektumfangs betrachtet.

5. Sicherheitseigenschaften

Durch die wichtige Rolle, die Evaluatoren einnehmen, muss sichergestellt werden können, dass diese sich korrekt verhalten. Wie bereits in einem vorangegangen A-SIT-Projekt erarbeitet wurde, lässt sich dies auf Basis aktueller Android-Geräte mittels Remote-Attestation erreichen. Da dieser Ansatz jedoch ein Restrisiko birgt (Geräte könnten über Exploits kompromittiert worden sein), wurden zusätzliche Sicherheitsmaßnahmen ergriffen.
Sollte sich ein Evaluator unehrlich verhalten, kann dies nicht nur detektiert werden, sondern durch die Bindung an dessen Public-Private-Keypair auch an andere Systemteilnehmer kommuniziert werden. Sinnvoll ist dies jedoch nur, wenn nicht beliebig Identitäten generiert werden können. Als mögliche Umsetzung dieser Restriktion bietet sich die zusätzliche Bindung von Evaluator-Identität an etwas nicht „billig“ oder ohne Aufwand austauschbares. Unter dem verwendeten Trusted-Computing-Konzept bietet sich die Bindung an ein Smartphone über dessen IEMI, bzw. MEID an. Dies kann insofern erzwungen werden, als dass die Identitäten von Evaluatoren nicht nur auf einem Public-Private-Keypair basieren, sondern andere Systemteilnehmer Evaluator-Identitäten nur akzeptieren, wenn diese ebenfalls von einer Geräte-ID abgeleitet sind. Konkret wird das Identifier-Format für ein sicheres Peer-to-Peer-Netzwerkdesign aus einem vorangegangen A-SIT-Projekt auf Basis von Android-Remote-Attestation vorgeschlagen, das diese Bedingung erfüllt.
Da eine umfassende Beschreibung des Verfahrens, sowie des Identifier-Formats, dessen Eigenschaften und die daraus folgenden Konsequenzen den Rahmen dieses Berichts sprengen würden, wird an dieser Stelle auf ein vorangegangenes A-SIT-Projekt zum Thema Remote-Attestation [5], sowie auf die aus einem anderen A-SIT-Projekt zum Thema Sicherheitsanalyse aktueller Peer-to-Peer-Netze hervorgegangene wissenschaftliche Publikation [6] verwiesen. Im Kontext dieses Projekts und des hier diskutierten Konzepts ist es lediglich relevant, *dass* es möglich ist, die Integrität aktueller Android-Smartphones aus der Distanz zu überprüfen und Identifikatoren an Geräte zu binden.
Durch den Einsatz von Trusted Computing ergibt sich nicht nur für Ersteller die Sicherheit, dass Policies korrekt evaluiert werden, sondern auch Empfänger können (z.B. im Falle einer Zugriffsverweigerung) sicher gehen, dass die Auswertung einer Zugriffs-Policy korrekt durchgeführt wurde.
Selbst wenn auch Trusted Computing gänzlich verzichtet wird, kann durch den Einsatz von Secret Sharing immer noch verhindert werden, dass eine einzelne Evaluator-Instanz Zugriff auf verschlüsselte Daten erhält, da diese lediglich auf einen einzelnen Share zugreifen kann. So lange nicht ausreichend viele Evaluatoren kollaborieren, hält diese Annahme. Ohne Trusted Computing gibt es jedoch keine Garantien, dass Policies korrekt ausgewertet werden, und es obliegt Erstellern, ihrer Meinung nach vertrauenswürdige Evaluatoren auszuwählen. Hier schafft Secret Sharing teilweise Abhilfe, da schlicht Mehrheitsentscheide bezüglich Gewährung von Zugriff umgesetzt werden können. Bei hinreichend hoher Anzahl von notwendigen Evaluatoren verliert der Einfluss einzelner Instanzen (und damit deren Fehlverhalten) an Bedeutung.

6. Fazit

Das im Rahmen dieses Projekts entwickelte Konzept erlaubt es, Daten durch Zugriffs-Policies in verteilten Umgebungen, ohne dass eine zentrale Instanz involviert ist, zu schützen. Da lediglich

[2] https://www.ethereum.org/

Secret-Sharing und konzeptionell simple kryptografische Primitive zum Einsatz kommen, welche nur den Besitz eines Public-Private-Keypair voraussetzen, kann das im Rahmen dieses Projekts entwickelte Konzept verhältnismäßig einfach in bestehende Systeme integriert werden. Unter anderem um dies zu erleichtern, wurde kein vorgegebenes Vertrauensmodell definiert, oder Trusted Computing zwingend vorgeschrieben. Auch wenn im Rahmen dieses Projekts klar der Dezentralisierungsaspekt und damit eine Integration in Peer-to-Peer-Architekturen verfolgt wurde, schließt das einen Einsatz im Unternehmensumfeld oder klassischen Client-Server-Umgebungen nicht aus.
In jedem Fall lässt sich dieses Konzept auch auf verteilte Systeme im eigentlichen Sinn anwenden, in deren Rahmen mehrere Teilnehmer auf einer gemeinsamen Datenbasis arbeiten, um feingranulare Zugriffberechtigungen umzusetzen. Der Datenoverhead des im Rahmen dieses Projekts entwickelten Systems bewegt sich im Bereich weniger Kilobyte, selbst wenn komplexe Policies und eine Vielzahl von Evaluatoren zur Policy-Auswertung definiert werden.

Ein Vorteil gegenüber ABE ist die Möglichkeit komplexe Policies, z.B. durch den Einsatz von XACML oder ähnlichen ABAC-Frameworks, umzusetzen, ohne dass dies mit hohem Rechenaufwand und komplexen kryptografischen Operationen einhergeht. Drüber hinaus ergibt sich durch den Einsatz von Secret-Sharing die zusätzliche Sicherheit, dass einzelne Evaluatoren (im Gegensatz zu Attribute Authorities im ABE-Kontext) nicht die Möglichkeit haben, Daten (unberechtigt) zu entschlüsseln.

Referenzen

[1] J. Bethencourt, A. Sahai und B. Waters, „Ciphertext-Policy Attribute-Based Encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, IEEE, 2007, pp. 321-334.

[2] V. Goyal, O. Pandey, A. Sahai und B. Waters, „Attribute-Based Encryption for Fine-grained Access Control of Encrypted Data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ACM, 2006, pp. 89-98.

[3] M. Chase, „Multi-authority Attribute Based Encryption," in *Theory of Cryptography*, Springer, 2007, pp. 515-534.

[4] Intel, „Enhanced Security for Applications and Data In-use," 2019. [Online]. Available: https://software.intel.com/sites/default/files/managed/6a/85/intel-sgx-product-brief-2019.pdf. [Zugriff am 11 06 2019].

[5] B. Prünster, G. Palfinger und C. Kollmann, „Fides - Unleashing the Full Potential of Remote Attestation," in *Proceedings of the 16th International Security and Cryptography (SECRYPT)*, Prag, Tschechien, SciTePress - Science and Technology Publications,, 2019.

[6] B. Prünster, E. Fasllija und D. Mocher, „Master of Puppets: Trusting Silicon in the Fight for Practical Security in Fully Decentralised Peer-to-Peer Networks," in *Proceedings of the 16th International Security and Cryptography (SECRYPT)*, SciTePress - Science and Technology Publications, 2019.

[7] B. Prünster, „Sichere Peer-to-Peer-Netze auf Basis von Selbstzertifizierung," 07 05 2018. [Online]. Available: https://technology.a-sit.at/sichere-peer-to-peer-netze-auf-basis-von-selbstzertifizierung/. [Zugriff am 11 07 2018].