**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

# DEMONSTRATOR IBE ZUR SCHLÜSSELVERWALTUNG

Version 1.0 vom 27.02.2018
Florian Reimair – florian.reimair@iaik.tugraz.at

*Schlüsselverwaltung in Unternehmen ist aufwändig, fehleranfällig und träge. Ein Beispiel dafür ist Email und S/MIME. Zukünftiger Belegschaft kann kein verschlüsseltes Mail geschickt werden, weil noch kein Schlüssel für die Personen existiert, abgelaufene Schlüssel verhindern ebenfalls das Versenden von verschlüsselten Nachrichten.*
*Ziel dieses Projekts ist es, angelehnt an die Technologie der Identity-based Encryption (IBE) ein zentrales Schüsselverwaltungssystem zu erstellen, das die oben genannten Beispiele meistern kann, somit den administrativen Aufwand der Schlüsselverwaltung in Unternehmen vereinfacht und schlussendlich die Hürden von vollständig verschlüsselter Kommunikation nimmt.*

## Inhaltsverzeichnis



## 1. Einleitung

Die Verwendung von verschlüsselter Kommunikation in Unternehmen birgt zahlreiche bekannte und unbekannte Hürden, die es sehr schwierig machen, ein effizientes und zugleich funktionales Arbeiten zu ermöglichen. Schuld daran sind neben dem notwenigen, komplexen Schlüssel- und Zertifikatsmanagement auch die Eigenheiten von Software, die Kryptographie anwenden sollen.

Effektives Schlüssel- und Zertifikatsmanagement, zum Beispiel, beinhaltet meist manuelle Schritte, die eine Person, sei es eine Fachkraft oder die Belegschaft selbst, erledigen muss. Das manuelle Eingreifen in den Managementprozess kann den Prozess deutlich verlangsamen und verhindert damit ein nahtloses und transparentes Einbinden in den beruflichen Alltag. Muss beispielsweise eine IT-Fachkraft eingreifen, kann es zu tagelangen Verzögerungen kommen, in der ein Teil der Belegschaft zu keiner verschlüsselten Kommunikation fähig ist. Muss die Belegschaft selbst

© A-SIT, Partner für Sicherheitsfragen

eingreifen, wird es für die betroffene Person schwierig, alle Vorgaben korrekt zu erfüllen. Beispiele dafür sind Konventionen zu Zertifikatstyp, Gültigkeitsdauer, Schlüssellängen, sowie Information zur Person und zur Organisation im Zertifikat. Weiters bleibt es der Belegschaft selbst überlassen, die jeweilige Software entsprechend zu konfigurieren und sich mit dessen Eigenheiten auseinanderzusetzen.

Ist alles geschafft, müssen die Schlüssel und Zertifikate noch verteilt werden. Gerne wird das von der Software gemacht, die selbst verschlüsseln kann. Schwieriger wird es, wenn unterschiedliche Software verwendet wird und diese nicht kompatibel ausgeführt sind. So wird beispielsweise eine Datensafe-Applikation nicht von einer Email-Applikation Zertifikate erhalten können.

Zu guter Letzt bleibt die Sicherung die Verschlüsselungsschlüssel oder Entsorgung der Signaturschlüssel, wenn deren Gültigkeit endet. Verlorene oder gestohlene Mobilgeräte können, wenn nicht ausreichend geschützt, private Schlüssel kompromittieren. Dies kann, genauso wie unsachgemäß zerstörte private Schlüssel, die sensiblen Daten einer Organisation lesbar machen und damit der Organisation schaden.

Ein solches System so aufzusetzen, zu warten und zu benutzen, dass keine Sicherheitslücken übrigbleiben, ist komplex und kostenintensiv. Oft wird dann auf Public Key Infrastruktur-Systeme (PKI Systeme), die Verschlüsselung ermöglichen würden, ganz verzichtet.

Dieses Projekt hat ein Konzept entwickelt, das in der Lage ist, die Komplexität und Trägheit eines solchen PKI Systems zu verringern. Das Konzept sieht vor (in Anlehnung an die Konzepte und Bausteine von Identity-based Encryption (IBE) Systemen), in einer Organisation zentrale Dienste anzubieten, die sowohl Schlüsselverwaltung als auch Zertifikatsverteilung vollständig autonom übernehmen und ausführen können.

Um das Konzept besser zu verstehen, wurden Demonstratoren erstellt, die sowohl eine Dateiverschlüsselungsapplikation für den Desktop-PC, als auch Email-Signatur und Verschlüsselung und eine Kryptographie-Applikation für den Browser zeigen. Die Demonstratoren bedienen sich des Cryptographic Service Interoperability Layer (CrySIL) (ehemals Skytrust), da dessen Architektur weitestgehend kompatibel zur in diesem Projekt erarbeiteten Architektur ist und dessen Erweiterbarkeit eine effiziente Projektabwicklung ermöglicht. Die Implementierung hat sich dabei auf die Vereinfachung der Prozesse für die beteiligten Personen konzentriert und andere Bereiche, wie zum Beispiel Implementierungssicherheit, Applikationsqualität und dgl. außer Acht gelassen.

Die Demonstratoren haben die Versprechungen des Konzepts bestätigt. In der Tat kann eine Organisation von dem in diesem Projekt erarbeiteten Konzept in vielerlei Hinsicht profitieren. Beispielsweise verschwindet der Zeitverlust verursacht durch die Notwendigkeit manueller Interaktion. Eigenheiten von Software müssen nur einmal gelöst werden. Die vorgabegerechte Handhabung von Zertifikaten und privaten Schlüsseln wird durch die automatisierte Verarbeitung konsistent und präzise. Und zu guter Letzt wird die Verteilung von Zertifikaten ebenfalls automatisiert und unabhängig von der eingesetzten Software auf den PCs der Belegschaft.

Offen bleiben Herausforderungen, die zum jetzigen Entwicklungsstand des Konzepts einen transparenten Einsatz verhindern. Wieder scheitert es an den Eigenheiten von Software, die Prozesse der PKI übernommen haben, um Organisationen Aufwände abzunehmen. Leider sind diese Lösungen meist nicht miteinander kompatibel. Daraus behindern sie die vollständig transparente Anwendung des in diesem Projekt erarbeiteten Konzepts.

Alles in allem zeigt dieses Projekt Wege zu Fortschritten zur vollständig verschlüsselter Kommunikation in Unternehmen auf. Verbleibende Hindernisse können mit weiterem Aufwand aus dem Weg geräumt werden. Damit verbucht dieses Projekt Teilerfolge in einem, so scheint es, zukunftsträchtigen Thema.

# 2. Architektur

Die Architektur des in diesem Projekt erarbeiteten Konzepts lehnt sich stark an die Methoden und Konzepte der Identity-based Encryption (IBE) und an die Konzepte von CrySIL, dem Cryptographic Service Interoperability Layer, an, die IBE in die Welt von PKI und X.509 überführen [1] bzw. ein invertiertes Vertrauensmodell vorschlagen [2]. Wie bei IBE gibt es auch hier eine zentrale Komponente, die, sofern erwünscht, sämtliche verschlüsselten Inhalte lesbar machen kann. Die Belegschaft gibt einen Namen an, der die Zielperson widerspiegelt. Daraufhin wird von der zentralen Instanz ein Zertifikat ausgeliefert. Sollte das Zertifikat noch nicht existieren, wird eines erzeugt.

Die Komponenten des Konzepts teilen sich grob in zwei Gruppen. Die Organisationsinfrastruktur bildet zunächst den zentralen Sammelpunkt für Zertifikate, Schlüssel und Personalmanagement in der IT der Organisation (Benutzernamen, Passwort, Email-Adressen, …). Diesen zentralen Sammelpunkt gibt es logisch nur ein einziges Mal (Load-Balancing und dgl. sind natürlich möglich). Dem gegenüber steht die Belegschaft. Mitarbeiter und Mitarbeiterinnen bedienen Geräte und Applikationen, die sensible Daten verarbeiten und nach Schutz in der Kommunikation verlangen. Diese Gruppe gibt es öfters, genauer, für jeden Mitarbeiter/jede Mitarbeiterin ein Mal.

Abbildung 1 illustriert die Bausteine und Verbindungen zwischen den Bausteinen, die das Konzept der CrySIL-enabled Organisation ausmachen. In der Belegschaft finden sich zunächst Mitarbeiter und Mitarbeiterinnen. Diese schaffen schützenswerte Daten und sollen/wollen diese nur per geschützter (verschlüsselter) Kommunikation weitergeben.

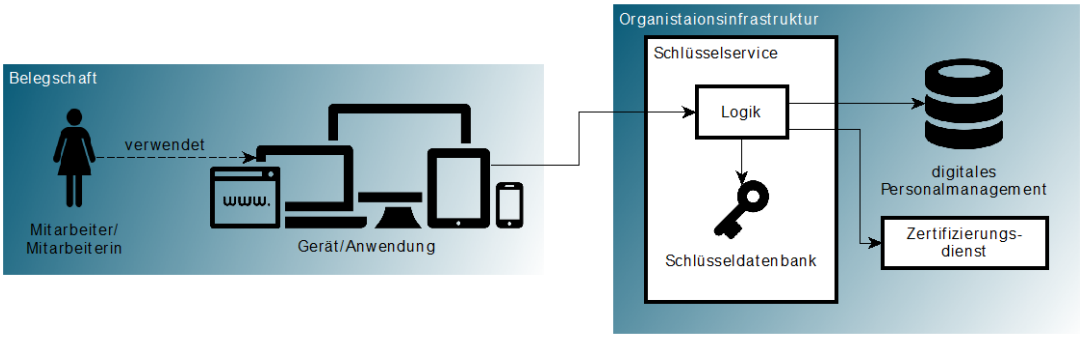


*Abbildung 1 Architektur des CrySIL-enabled Organisation Konzepts*

Um die geschaffenen Daten zu persistieren, verwenden die Mitarbeiter und Mitarbeiterinnen Anwendungen, die auf einer Vielzahl von verschiedenen Geräten benutzt werden. Diese Geräte und Anwendungen stellen somit die Werkzeuge der Mitarbeiter und Mitarbeiterinnen dar, auf denen die Mitarbeiter und Mitarbeiterinnen (sensible) Inhalte in speicherbare Form bringen. Die Anwendung ist damit auch verantwortlich dafür, dass die sensiblen Inhalte entsprechend geschützt werden. Entweder wird der Inhalt durch Verschlüsselung verschleiert oder durch elektronische Signatur vor (unautorisierten) Änderungen geschützt. Die Applikation übernimmt dabei die Kommunikation mit der Organisationsinfrastruktur und leitet den jeweiligen Mitarbeiter/die jeweilige Mitarbeiterin durch die notwendigen Prozesse.

Das zentrale Schlüsselservice nimmt Anfragen und Aufträge der Geräte/Applikationen entgegen. Bevor diese beantwortet oder ausgeführt werden, wird das digitale Personalmanagement kontaktiert, um entweder die Existenz einer Zielperson zu bestätigen oder die Korrektheit entgegen genommener Authentifizierungsdaten zu überprüfen. Nach erfolgreicher Überprüfung wird nun die Schlüsseldatenbank nach dem passenden Zertifikat/Schlüssel gefragt. Existieren diese nicht, werden sie mit Hilfe des Zertifizierungsdienstes erstellt und in der Schlüsseldatenbank abgelegt. Dort stehen die Zertifikate/Schlüssel der Belegschaft zur Verfügung.

Das digitale Personalmanagement, welches in einer für das Konzept interessanten Organisation, sehr wahrscheinlich bereits existiert, wird als Informationsquelle für den Personalstand verwendet. Der Schlüsseldienst kann dort erfragen, ob denn eine gesuchte Person in der Organisation existiert,

oder aber die Korrektheit der Zugangsdaten, also die Authentizität, eines Mitarbeiters oder einer Mitarbeiterin überprüfen.

Zu guter Letzt benötigt das Konzept einen Zertifizierungsdienst. Dieser Dienst nimmt zumindest Certificate Signing Requests (CSRs) an und kann diese autonom abarbeiten. Wenn der Dienst auch über einen Schlüsselgenerator verfügt, können Schlüssel und Zertifikate vollständig automatisch und bei Bedarf erstellt werden.

Alles in allem verwendet das Konzept viele bekannte und erprobte Technologien. Dennoch kann durch die hier präsentierte Kombination dieser Technologien gemischt mit etwas Invention ein für die Belegschaft deutlich einfacheres PKI-artiges System erstellt werden.

## 2.1. Ablauf: Verschlüsseln für Zielperson

Dieses Kapitel diskutiert eine mögliche Verwendung des Konzepts, geht dabei aber detailliert auf das Zusammenspiel zwischen den Bausteinen sowie auf den gesamten Prozess ein. Abbildung 2 stellt die Abläufe und Interaktionen, unter Zuhilfenahme von Elementen aus der Business Process Modeling Language (BPMN), dar.

Der hier als Beispiel herangezogene Anwendungsfall betrachtet den Prozess, den eine der Organisation zugehörigen Person folgen muss, um einer anderen, der selben Organisation zugehörigen Person, Daten in verschlüsselter Form übermitteln will. Die Beschreibung setzt dabei eine geeignete Applikation voraus, die die Interaktion mit dem Benutzer übernimmt. Diese Applikation weiß mit den zentralen Komponenten des Konzepts umzugehen. Weiters wird nicht weiter auf den Transportweg eingegangen, da dieser das Konzept und damit den Prozess nur unwesentlich beeinflusst.

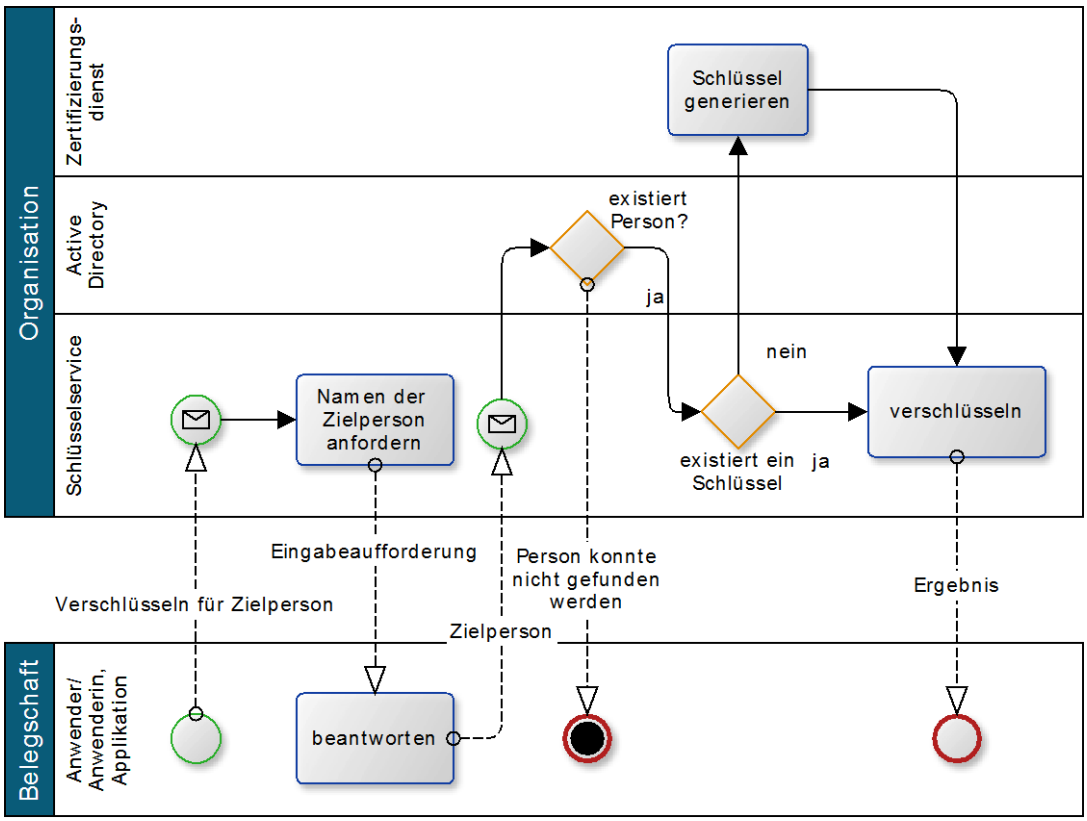


*Abbildung 2 Ablauf: Verschlüsselung für einen Zielperson*

Der Prozess beginnt damit, dass eine Person eine Applikation bedient, die Daten für eine Zielperson verschlüsseln will. Die Applikation fordert beim Schlüsselservice der Organisation die Verschlüsselungsoperation an. Der Schlüsselservice antwortet mit der Frage nach der Zielperson. Die den Prozess initiierende Person beantwortet diese Frage mit zum Beispiel einem Benutzernamen.

Der Schlüsselservice überprüft jetzt, ob eine Person zum erhaltenen Benutzernamen in der Organisation existiert. Existiert so eine Person nicht, wird dieses der initiierenden Person in Form eines Abbruchs des Prozesses mitgeteilt. Existiert die Zielperson, überprüft der Schlüsselservice, ob denn bereits ein zugehöriges und gültiges Zertifikat/Schlüsselpaar existiert. Existiert so eines nicht, wird ein neues Schüsselpaar generiert und der organisationeigene Zertifizierungsdienst beauftragt, am Erstellen eines zugehörigen Zertifikats mitzuwirken. Das Schlüsselpaar und das Zertifikat werden vom Schlüsselservice für zukünftige Anfragen aufbewahrt. Nun kann der Schüsselservice den ursprünglichen Auftrag des initiierenden Benutzers beantworten, indem er mit dem ev. neu erstellen Schlüsselpaar die Verschlüsselungsoperation durchführt und das Ergebnis an die Applikation übergibt.

Alles in allem können, bis auf die Beantwortung der Frage nach der Zielperson, alle Schritte automatisiert durchgeführt werden. Durch die Frage nach der Zielperson muss die Applikation keine Daten, wie Zertifikate, Benutzernamen, etc., speichern. Letztendlich ist garantiert, dass die Zielperson ein gültiges, den Vorgaben entsprechendes Zertifikat und ein zugehöriges, ebenfalls den Vorgaben entsprechendes Schüsselpaar besitzt.

## 2.2. Ablauf: Zielperson entschlüsselt

Dieses Kapitel diskutiert eine andere mögliche Verwendung des Konzepts, die speziell mit der vorher behandelte Verwendungsmöglichkeit zusammenspielt. Dabei geht es aber detailliert auf das Zusammenspiel zwischen den Bausteinen sowie auf den gesamten Prozess ein. Abbildung 3 stellt die Abläufe und Interaktionen, unter Zuhilfenahme von Elementen aus der Business Process Modeling Language (BPMN), dar.

Der hier als Beispiel herangezogene Anwendungsfall betrachtet den Prozess, den eine einer Organisation zugehörigen Person folgen muss, um verschleierte Daten zu entschlüsseln, die von einer anderen, der selben Organisation zugehörigen Person, übermittelt wurden. Die Beschreibung setzt dabei eine geeignete Applikation voraus, die die Interaktion mit dem Benutzer übernimmt. Diese Applikation weiß mit den zentralen Komponenten des Konzepts umzugehen. Weiters wird nicht weiter auf den Transportweg eingegangen, da dieser das Konzept und damit den Prozess nur unwesentlich beeinflusst.

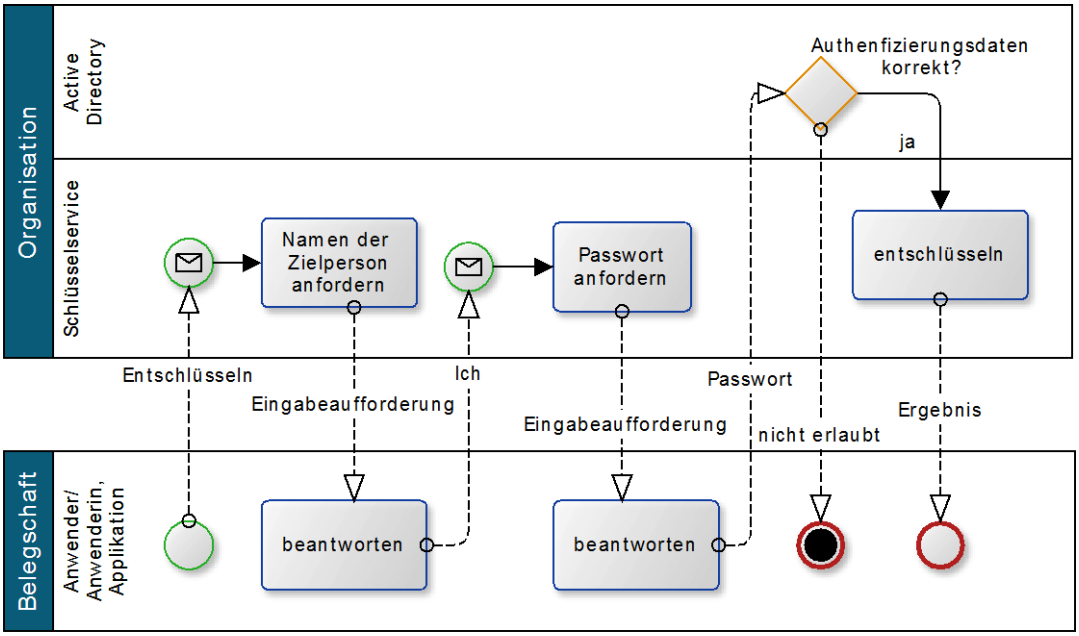


*Abbildung 3 Ablauf: Entschlüsseln von Daten durch die Zielperson*

Wieder startet der Prozess mit einer Person, die eine Applikation bedient, die ein eben eingelangtes, für die bedienende Person verschlüsseltes Datenpaket entschlüsseln will. Die Applikation initiiert ein Entschlüsselungsverfahren, indem der organisationseigene Schlüsselservice den Auftrag bekommt, das Datenpaket zu entschlüsseln. Der Schlüsselservice fragt nun wieder nach der Zielperson,

welche die initiierende Person mit sich selbst beantwortet. Im Unterschied zum vorher beschriebenen Verschlüsselungsverfahren fragt der Schlüsselservice jetzt aber Zugangsdaten ab, die für die Autorisierung des Entschlüsselungsauftrags verwendet werden. Die initiierende Person gibt also ihr/sein Passwort an und der Schlüsselservice überprüft Namen und Passwort der Person mit Hilfe des digitalen Personalmanagements (zum Beispiel Microsofts Active Directory). Stimmen die Authentifizierungsdaten, wird der ursprüngliche Auftrag ausgeführt und das Ergebnis an die initiierende Applikation und damit an die initiierende Person geliefert. Stimmen die Daten nicht, wird der Prozess abgebrochen.

Wieder können, bis auf die Beantwortung der Fragen nach Zielperson und Authentifizierungsdaten, alle Schritte automatisiert durchgeführt werden. Der Schlüsselservice übernimmt das Erstellen des Ziel-Schüssels/-Zertifikats und die zugehörige Person kann den Schlüssel einfach und ohne tieferes Wissen und Aufwand verwenden. Wieder bleibt der Schlüssel beim Schlüsselservice, damit kann die Person ihren/seinen Schlüssel von verschiedenen Geräte verwenden, ohne dabei aufwändig Schüsselsynchronisation zu betreiben oder die Vertraulichkeit des Schlüssels zu kompromittieren. Letztendlich ist garantiert, dass die Zielperson ein gültiges, den Vorgaben entsprechendes Zertifikat und ein zugehöriges, ebenfalls den Vorgaben entsprechendes Schüsselpaar erhält.

## 2.3. Schlüsselservice

Nachdem das Konzept, d. h. die Architektur und zwei Ablauf-Beispiele, vorgestellt wurden, werden nun die zwei Hauptkomponenten genauer diskutiert. Zuerst wird der organisationsinterne Schlüsselservice behandelt.

Der Schlüsselservice bedient sich des Cryptographic Service Interoperability Layer (CrySIL) Konzepts [3], dessen Flexibilität eine saubere Trennung zwischen den Zuständigkeiten und dessen Erweiterbarkeit zu schnellen Ergebnissen führt. Abbildung 4 illustriert den Schlüsselservice, dessen Innenleben und dessen externe Abhängigkeiten am Beispiel einer Active Directory-Anbindung. Der CrySIL Knoten, ein Zusammenschluss aus verschiedenen Modulen der CrySIL-Welt, bildet somit das Herz der Komponente. Erreichbar ist dieser Knoten aus dem Intranet und bedient sich vorhandener Infrastruktur wie eines digitalen Personalmanagements (z.B.: Microsofts Active Directory) und einer Zertifizierungsstelle.

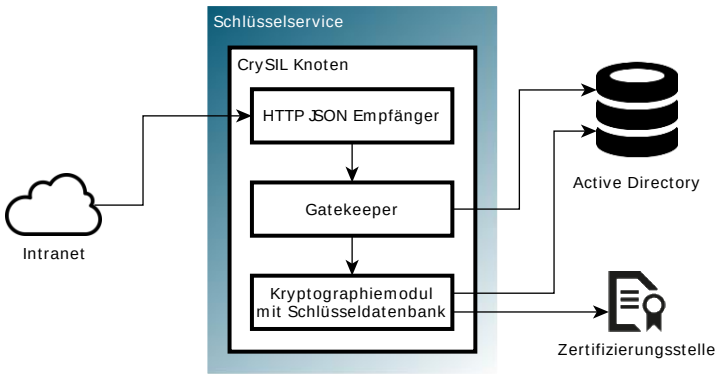


*Abbildung 4 Der Schlüsselservice im Detail*

Anfragen aus dem Intranet werden zunächst vom HTTP JSON Empfänger-Modul entgegengenommen. Dieses Modul überprüft die strukturelle Korrektheit eines Auftrags und leitet dieses an das Gatekeeper-Modul weiter. Das Gatekeeper-Modul entscheidet, ob und wenn ja, welche Authentifizierungsdaten notwendig sind, um den Auftrag zu autorisieren und damit dem Kryptographie-Modul weiterzugeben. Welche Authentifizierungsdaten notwendig sind, ist im Gatekeeper-Modul selbst konfiguriert, die Überprüfung erfolgt hier allerdings über das existierende Service (in diesem Beispiel Microsofts Active Directory). Nachdem das Gatekeeper-Modul alle zur Autorisierung notwendigen Daten gesammelt und geprüft hat reicht es, den ursprünglichen Auftrag an das Kryptographie-Modul weiter. Das Kryptographie-Modul bedient sich einer Zertifizierungsstelle um Schüssel, wenn notwendig, on-the-fly zu generieren bzw. diese in ein Zertifikat zu packen. Die Schlüsseldaten selbst bleiben in der Schlüsseldatenbank des Kryptographie-Moduls.

Alles in allem zeigt der in diesem Projekt behandelte Anwendungsfall der CrySIL-Enabled Organisation, dass Schüsselmanagement nicht mit großem Aufwand verbunden sein muss. Auch das CrySIL Konzept hat sich einmal mehr bewährt.

## 2.4. Applikation

Aufgrund der Umsetzung mit Hilfe des CrySIL Frameworks ist es notwendig, dass auch die Client-seite des Konzepts der CrySIL-Enabled Organisation CrySIL spricht. Aus diesem Grund ist stellt auch in der Applikation ein CrySIL-Knoten einen wichtigen Baustein bereit. CrySILs vorhandenen Implementierungen verstecken sich allerdings hinter gut bekannten APIs für kryptographische Provider und damit muss die Applikation selbst nicht auf das CrySIL-Framework ausgelegt sein. Abbildung 5 illustriert die Applikation, ihre Bausteine und ihre Umgebung.

Wann immer die Programmlogik kryptografische Dienste benötigt, wendet sie sich an ihre Krypto-graphie-API hinter der sich allerdings eine Implementierung für das CrySIL-Framework versteckt. Damit geht ein Verschlüsselungsauftrag direkt an das Kryptographie-API-Modul des CrySIL-Knotens. Dieses Modul extrahiert alle notwendigen Daten und erstellt einen CrySIL-konformen Auftrag. Dieser Auftrag wird durch das Authentifizierungsmodul zum HTTP JSON Sender-Modul weitergeleitet, welches diesen durch das Intranet an den Schlüsselservice sendet. Hier verstecken sich noch keine Finessen und das CrySIL Konzept kann hier keine Vorteile ausspielen.

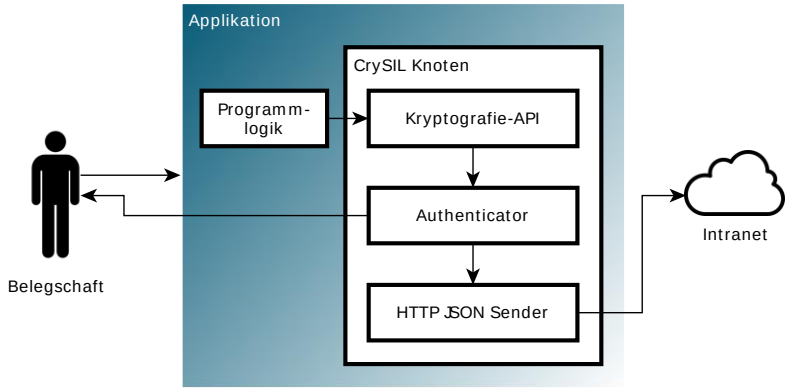


*Abbildung 5 Die Applikation im Detail*

Wenn der Schlüsselservice aber nach Authentifizierungsdaten fragt, also nach einem Namen und, im Falle eines Entschlüsselungsauftrags, auch nach Geheimnissen, leistet das Authentifizierungs-modul seinen Betrag zum komfortablen Prozess des CrySIL-Enabled Organisation Prozesses. Eine Anfrage nach der Zielperson kommt also aus dem Intranet (vom Schlüsselservice) beim HTTP JSON Sender an. Dieser überprüft die strukturelle Korrektheit der Kommunikation und gibt nach erfolgreicher Prüfung die Anfrage an das Authentifizierungsmodul weiter. Dieses interagiert jetzt direkt mit dem Benutzer und fragt diesen nach den benötigten Daten. Die Applikation muss für diese Interaktion mit dem Benutzer keine Sorge tragen. Nach Erhalt der benötigten Daten sendet das Authentifizierungsmodul diese über den HTTP JSON Sender wieder übers Intranet zum Schüssel-service.

Mit diesem Konzept wird es möglich, existierende Applikationen (wie zum Beispiel Emailprogramme, Betriebssystem-eigene Verschlüsselungsprogramme und dgl.) nachträglich in die CrySIL-Enabled Organisation einzupflegen und dessen Vorteile nutzbar zu machen. Weiters brauchen sich so Entwickler solcher Anwendung nicht um Kryptographie kümmern, sondern diese lediglich verwenden.

# 3. Demonstratoren

Im Laufe des Projekts sind Demonstratoren entstanden, die es erlauben, die Eigenschaften des Konzepts klarer und anschaulicher darzustellen. Es wurden drei Demonstratoren erstellt, die jeweils

stellvertretend für Applikationsgruppen stehen, nämlich eine Webapplikation die auf jedem Gerät ohne Installation verwendet werden kann, eine CMS Verschlüsselungsapplikation, die die Gruppe der zu installierenden Applikationen vertritt und schlussendlich Thunderbird Mail (bzw. ein Security Module Plugin dafür), die die Gruppe der Email-Applikationen darstellt.

Leider konnte der Thunderbird Mail Demonstrator nicht zur vollen Zufriedenheit umgesetzt werden. Thunderbird Mail wie auch andere Email-Applikationen versuchen seit langem, die Verwendung von Kryptographie in der schriftlichen Kommunikation einfacher zu gestalten und haben begonnen, dem Benutzer Dinge wie Schlüsselmanagement und dgl. abzunehmen bzw. zu erleichtern. Thunderbird Mail fragt ein konfiguriertes Security Module (wie Software Implementierungen, Smartcards oder auch die Anbindung an das CrySIL-Enabled Org-Konzept) zum Beispiel lediglich bei Applikationsstart nach den vorhandenen Zertifikaten und Schlüsselpaaren. Für Signaturen ist das kein Problem, sobald aber eine verschlüsselte Nachricht gesendet werden soll wird lediglich der anfangs abgeholte Datenstand nach einem passenden Zertifikat durchsucht. Die bedienende Person wird dadurch nicht explizit nach einem Empfänger gefragt, und damit kann dieser auch nach keinem Namen suchen lassen. Damit ist es im Thunderbird-Anwendungsfall auch nicht möglich, Schlüssel on-the-fly zu generieren.

Die Demonstratoren stehen am Internetauftritt von CrySIL[1] bereit; die Webapplikation[2], die Verschüsselungsapplikation[3] und das Security Module Plugin für Thunderbird[4]. Um die CrySIL-Enabled Organisation zu simulieren, muss lediglich der entsprechende Schlüsselservice ausgewählt werden[5]. Die Simulation verzichtet dabei auf sowohl auf ein ActiveDirectory, als auch auf eine CA. Beides ist in Software nachgebaut. Die ActiveDirectory Simulation lässt jeden Suchbegriff zu und erwartet sich ein gleichlautendes Passwort. Der Source-Code zu den Demonstratoren finden sich in den Git-Repositories.
Nichts desto trotz zeigten die Demonstratoren auf, dass Schlüsselmanagement in Organisationen für die Belegschaft nicht komplex und aufwändig sein muss und dass vollständige Verschlüsselung durchaus ein erreichbares Ziel ist.

## 4. Zusammenfassung

Dieses Projekt hat sich zum Ziel gesetzt, Schlüsselmanagement in Organisationen unter die Lupe zu nehmen, Unzulänglichkeiten, Probleme und Verbesserungspotentiale zu identifizieren und diese auch zu adressieren. Das dabei entstandene Konzept verwendet existierende Technologien soweit möglich, hat noch offene Herausforderungen vor sich, zeigt aber deutliche Erfolge, die sich unter anderem in reduziertem zeitlichen und organisatorischen Aufwand bzgl. Schlüsselmanagement und Zertifikatsverwaltung für die Belegschaft einer Organisation widerspiegeln.

## Literaturverzeichnis


[1] F. Reimair und J. Feichtner, „Attribute-based Encryption goes X.509," in *Proceedings of the International Conference on e-Business Engineering (ICEBE)*, Peking, 2015.

[2] F. Reimair, B. Prünster und P. Teufl, „In Certificates We Trust - Revisited," in *Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, Helsinki, 2015.

[3] F. Reimair, P. Teufl und T. Zefferer, „CrySIL: Bringing Crypto to the Modern User," in *Lecture Notes in Business Information Processing*, Bd. 246, Springer, 2016, pp. 70-90.


[1] https://crysil.iaik.tugraz.at
[2] https://crysil.iaik.tugraz.at/browser-apps/example/demo-simplest.html
[3] https://crysil.iaik.tugraz.at/crysil-desktop-app/crysil-desktop-app.jar
[4] https://crysil.iaik.tugraz.at/thunderbird/linux.tar.gz
[5] „CrySIL Node for IBE emulation", https://crysil.iaik.tugraz.at/tomcat/webservice-ibe/json