Asymmetrische Kryptographie in Java

Norman Vetter

Seminar „Sichere verteilte Anwendungen mit Java“

Universität Potsdam

Wintersemester 2012/13

Inhaltsverzeichnis

1 Einleitung

So sieht eine Section aus!

2 Theoretische Grundlagen

2.1 Ziele

Ein kryptographisches System dient zum verschlüsseln und entschlüsseln von Texten und anderen Daten, um deren Inhalt vor Dritten geheim zu halten. Genauer gibt ein Kryptosystem an wie ein Klartext in einen von Dritten nicht lesbaren Kryptotext umgewandelt werden kann. Und wie dieser Kryptotext später wieder in einen lesbaren Klartext transformiert wird. Anders als die Steganografie zielt die Kryptographie darauf ab lediglich den Inhalt einer Nachricht zu verschlüsseln, nicht aber deren Existenz zu verbergen. Die asymmetrische Kryptographie ist eines dieser kryptographischen Systeme.

2.2 Asymmetrische Kryptographie

Die asymmetrische Kryptographie wurde Mitte 1970 von Ralph Merkle sowie von Diffie und Hellmann entwickelt. Sie beruht auf der Idee zur Kommunikation zwischen 2 Instanzen ein Schlüsselpaar zu verwenden. Dieses Schlüsselpaar besteht aus dem privaten und dem öffentlichen Schlüssel. Zur Kommunikation muss im Vorhinein ein gegenseitiger Austausch des öffentlichen Schlüssels erfolgt sein, denn die zu schickende Nachricht ist vom Sender mit dem öffentlichen Schlüssel des Empfängers in einen Kryptotext um zu wandeln. Nach Empfang entschlüsselt der Empfänger nun die Nachricht mit seinem eigenen privaten Schüssel. Im Falle einer Antwort verschlüsselt der ehemalige Empfänger (nun Sender) seine Nachricht wieder mit dem öffentlichen Schlüssels seines Gegenüber. Welcher die Entschlüsselung erneut mit dem eigenen privaten Schlüssel durchführen muss. Somit ist während der Kommunikation kein erneuter Austausch eines sicheren Schlüssels notwendig und auch erneute Kommunikationen können mit den bereits vorhandenen Schlüsseln durchgeführt werden. Wichtig ist hierbei jedoch, dass die Sicherheit des privaten Schlüssels und die Authentizität des öffentlichen Schlüssels gewährleistet ist. Mehr dazu in den folgenden Kapiteln.

2.3 Grundlage asymmetrischer Systeme

Im obigen Szenario haben wir gesehen wie ein grober Kommunikationsablauf zwischen Zwei Parteien aussieht. Dieses wird in folgender Grafik (nach [Eck13]) veranschaulicht:

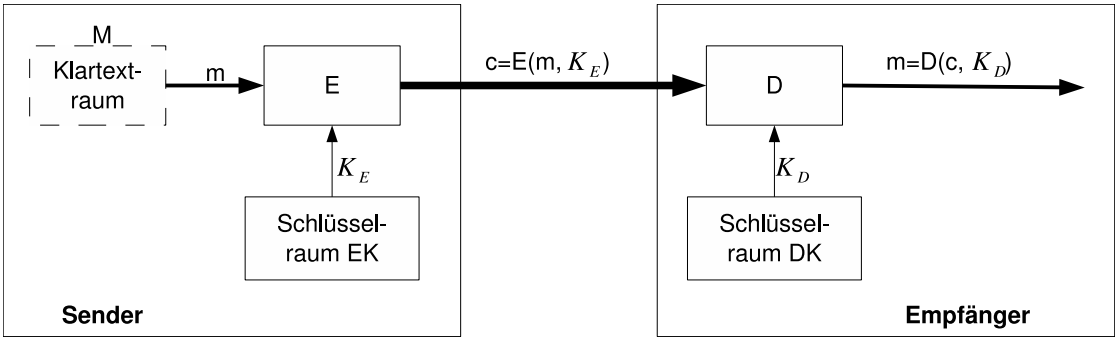

Abbildung 1: Komponenten eines Kryptosystems

1. Tupel
   (M, C, EK, DK, E, D)
2. 2 endliche Alphabete
   (A_1, A_2)
3. Menge von Klartexten
   $(M \subseteq A_1^* \backslash \emptyset)$
4. Klartext
   $(m \in M)$
5. Menge von Kryptotexten
   $(C \subseteq A_2^* \backslash \emptyset)$
6. Kryptotext
   $(c \in C)$
7. Verschlüsselungsschlüsselraum
   $(EK \backslash \emptyset)$
8. Entschlüsselungsschlüsselraum
   $(DK/\emptyset$
   $mit\ f : EK \rightarrow DK$
   $und\ f(K_E) = K_D)$
9. Verschlüsselungsverfahren
   $(E :\ MxEK \rightarrow C)$
10. Entschlüsselungsverfahren
    $(D :\ CxDK \rightarrow M)$
11. Es gilt:
    $\forall m \in M : D(E(m, K_E), K_D) = m$

Die Eigenschaft (11.) der obigen Legende zeigt uns das Zusammenspiel unserer einzelnen Komponenten. Wird ein Klartext m mit einem Schlüssel $K_E \in EK$ durch ein kryptographisches Verfahren E in einen Kryptotext umgewandelt. So ist gegeben, dass dieser Kryptotext unter Verwendung des zu K_E gehörigen Schlüssels $K_D \in DK$, und des kryptographischen Verfahrens D wieder in den ursprünglichen Klartext m umgewandelt werden kann. Damit dies gegeben ist, muss unser asymmetrisches Kryptosystem einige wichtige Punkte erfüllen. Es muss:

- eine effiziente Möglichkeit zur Erzeugung von Schlüsselpaaren (K_E, K_D) geben.
- garantiert sein, dass der Private (K_D) nicht effizient aus dem öffentlichen Schlüssel (K_E) gebildet werden kann.
- möglich sein effizient zu Ver- und Entschlüsseln.

Um dies zu erreichen nutzen wir Einwegfunktionen.

Einwegfunktionen sind besondere Funktionen $(f : X \to Y)$ bei denen das Urbild (X) nicht unter vertretbarem zeitlichem Aufwand aus dem Bild (Y) zu berechnen ist. Mathematische Probleme, welche derartige Funktionen bilden, sind unter Anderem das Faktorisierungsproblem und der diskrete Algorithmus. Es ist nicht bewiesen, dass die Umkehrung nicht möglich ist, jedoch übersteigt die Komplexität der Berechnung unser heutiges Vermögen diese in einer vertretbaren Zeitspanne zu lösen.

Bei der Verwendung von normalen Einwegfunktionen in einem Kryptosystem wäre zwar die Sicherheit der Daten garantiert, jedoch stellt die Umkehrung selbst für autorisierte Personen ein unüberwindbares Hindernis dar. Wir benötigen zum Entschlüsseln unserer Daten eine Hintertür - die so genannte Falltür. Einwegfunktionen mit Falltür bieten eine identische Sicherheit wie normale Einwegfunktionen, und die Option verschlüsselte Daten unter Kenntnis eines Geheimnisses (bei und der private oder öffentliche Schlüssel) zu entschlüsseln. Beispiele für mathematische Probleme mit Falltür sind die h-te Potenz modulo(n) und der zusammengesetzte modulo(n) ([Eck13]).

2.4 RSA

2.5 Verschlüsselung und Endschlüsselung

2.6 Schlüsselmanagement

Bei der asymmetrischen Kryptographie ergeben sich aufgrund ihrer besonderen Grundidee der zwei Schlüssel auch ganz eigene Möglichkeiten und Pflichten zur Erzeugung, Aufbewahrung und möglichen Wiederherstellung dieser.

Die Erzeugung asymmetrischen Schlüsselmaterials kann der Nutzer entweder selbst übernehmen oder einer zentralen Instanz (z.B. CA) überlassen. Sollte der Nutzer den Schlüssel selbst generieren hat dies zum Vorteil das er die volle Kontrolle über den Erstellungsprozess hat. Jedoch obliegt ihm somit auch die Verantwortung den Prozess und die fertigen Schlüssel durch nötige Einstellungen im System und nötige Vorsicht zu sichern. Besonderes Augenmerk liegt dabei auf dem privaten Schlüssel, welcher unter keinen Umständen öffentlich werden darf. Da die Sicherheit der verschlüsselten Daten dann nicht mehr gewährleistet ist. Im falle der Erzeugung durch eine Zentrale Instanz gibt der Nutzer diese Verantwortung Größtenteils aus der Hand, sollte sich

jedoch bewusst sein das somit auch keine Kontrolle seinerseits besteht. Des weiteren ist es nun notwendig den privaten Schlüssel sicher dem Nutzer zur Verfügung zu stellen, was unter Umständen kritisch werden kann. Ist der Schlüssel erstmal erzeugt muss dieser zwingend sicher aufbewahrt werden.

Die Schlüsselspeicherung muss gewährleisten das der private Schlüssel keinem Unbefugten zur Verfügung steht. Um ein Fehlverhalten des Nutzers aus zu schließen wäre es also ratsam nicht einmal diesem den Zugang zu dem privaten Schlüssel zu gestatten. Diesen Weg gehen Token (oder SmartCard). Token sind Chipkarten oder USBsticks bei denen der Schlüssel auf einem internen Chip sicher verwahrt wird. Der Zugriff ist ausschließlich über Sicherheitssoftware in Emailprogrammen (PGP) oder anderen (OpenSSH) möglich, welche den Schlüssel selbstständig auslesen und ihn nicht zwischenspeichern. Außerdem ist es natürlich möglich den Schlüssel selbst sicher zu verwahren.

Die Verbreitung des öffentlich Schlüssels ist über beliebige Wege möglich. Zum einen besteht die Möglichkeit den öffentlich Schlüssel auf einem KeyServer zu lagern, oder ihn per E-Mail zu verschicken. Es gibt noch wesentlich mehr Möglichkeiten welche jetzt hier nicht alle gennant werden sollen. Zu berücksichtigen ist jedoch das die Authentizität eines öffentlich Schlüssels gegeben sein muss. Nicht immer lässt sich garantieren das der Schlüssel den man als öffentlich Schlüssel von Person A gefunden oder erhalten hat auch wirklich von dieser ist. Zu diesem Zweck werden Schlüssel von Gesprächspartnern, welche sich bereits gegenseitig überprüft haben, Unterschrieben. Es wird also eine mit dem privaten Schlüssel des Signierenden verschlüsselte Signatur an den öffentlichen Schlüssel an gehangen. Aufgrund dessen unsere Operationen umkehrbar sind. Lässt sich nun diese Signatur mit dm öffentlichen Schlüssel des Unterzeichners öffnen und überprüfen. Da die Signatur jedoch nicht geändert werden kann ist sie zudem vor Fälschung sicher. Dieses Verfahren wird ebenso bei Certifikaten in anderen Situationen genutzt.

2.7 Hybride Kryptographie

3 Kryptographie mit Java

3.1 Java Cryptography Architecture

3.2 Zufallszahlen

3.3 Schlüsselgenerierung

3.4 Schlüsselspeicherung

3.5 Schlüsseleinigung

3.6 Ver- und Entschlüsselung

4 Zusammenfassung

Zum Schluss bitte eine Zusammenfassung!

Literatur

[Eck13] Claudia Eckert. *IT-Sicherheit : Konzepte - Verfahren - Protokolle*. Oldenburg, 2013.