Fachhochschule Wedel

Seminararbeit

asymmetrische Kryptographie
Hybride Verschlüsselungsverfahren
& Digitale Signatur

Erstellt von:

Sabrina Elisabeth Carstens

Betreut durch:

Prof. Dr. Michael Anders

Abgabe am:

28. Mai 2017

Inhaltsverzeichnis

I

Asymmetrische Kryptographie

1 Geschichtliches[1]

Im Gegensatz zur 2000 Jahre alten symmetrischen, stellt die asymmetrische Verschlüsselung einen neuen Bereich der Kryptographie dar. Mit den Arbeiten zu einem geheimen Schlüsselstausch legten Whitfield Diffie, Martin Hellman und Raph Merkle Anfang der siebziger Jahre den Grundstein. Auf Basis des von Ralph Merkle 1974 entdeckten Merkles Puzzeles (welches aber erst 1978 publiziert wurde) veröffentlichten Whitfield Diffie und Martin Hellman im Jahr 1976 den Diffie-Hellman-(Merkel)Schlüsselaustausch. Ein Konzept für den sicheren, geheimen Schlüsselaustausch über unsichere Kanäle. 1977 folgte der Durchbruch mittels eines konkreten Verfahrens. Das RSA Verfahren, benannt nach den Entwicklern Ron Rivest, Adi Shamir und Leonard Adleman, eignet sich sowohl für die Verschlüsselung als auch für Signaturverfahren. Popularität erlangte das Verfahren 1991 durch die Veröffentlichung des "Pretty Good Privacay (PGP)"Programm, welches erstmalig eine einfache Anwendung für den privatem Bereich ermöglichte. Der 1978 von Robert J. McEliece entwickelte Algorthimus findet aufgrund der enormen Schlüsselgröße nur selten Anwendung, jedoch ist wäre er auch unter Zuhilfenahme eines Quantencomputers nicht zu knacken. Nur ein Jahr später stand das Rabin-Kryptosystrem zur Verfügung. Basiert es doch auf dem selben Prinzip wie das RSA-Verfahren und bietet theoretisch auch die Option zur Signatur, wird es dennoch kaum verwendet. Gründe hierfür sind die Uneindeutigkeit der Entschlüsselung und die Menge an Angriffsmöglichkeiten. 1984 wurde das Chor-Rivest Verfahren, welches auf dem Rucksack- oder Untersummenproblem beruht, veröffentlicht. Aufgrund der einfachen Lösbarkeit durch einen Greedy-Algorithmus dient es heutzutage lediglich dem Unterrichtswesen.[2] Thaher Elgamal entwickelte 1985 ein auf dem Diffie-Hellman Schlüsselaustausch beruhendes Kryptosystem. Zu dem patentfreien Verschlüsselungsverfahren existiert ein ähnliches Signaturverfahren.[3]

2 Notwendigkeit der asymmetrischen Verschlüsselung

Bei der Verwendung symmetrischer Kryptographie begegnen uns eine Vielzahl von Problematiken:

Zunächst ist es notwendig, dass die Kommunikationspartner ihren Schlüssel austauschen, sodass beide Parteien, Ver- und Entschlüsseln können. Die offensichtliche Notwendigkeit der Geheimhaltung und der sicheren Übertragung kann hierbei jedoch schwer gewährleistet werden. Als Lösungsmöglichkeiten bieten sich hier folgende Optionen:

- **Manueller Schlüsselaustausch bei persönlichem Kontakt**
  → Im Zeitalter gigantischer Datenströme, die weltweit geteilt werden, stellt dies eine unzweckmäßige Lösung dar. Vor allem da eine persönliche Bekanntschaft der Kommunikationspartner meist nicht gegeben ist.[4]

- **Authentifizierungsserver Bsp. Kerberos[5]**
  → Durch Kerberos ist es möglich Benutzer und Rechner zu authentifizieren, jedoch besteht keine Möglichkeit Kerberos gegenüber den Nutzern zu authentifizieren. Es muss System umfassend angewandt werden, jeder Dienst muss kompatibel sein. Zudem wird ein Master-Key zum Chiffrieren der Passwort-Datenbank verwendet. In Kombination mit dem mangelnden Schutz vor Systemmodifikationen stellt Kerberos damit eine eher unsichere Wahl dar.

Des Weiteren muss der Schlüssel nun von beiden Teilnehmern unter höchster Geheimhaltung verwahrt werden. Beide werden dementsprechend zum potenziellen Ziel von Angriffen. Zusätzlich benötigt jedes Kommunikationspaar einen eigenen Schlüssel, somit wächst die Schlüsselanzahl quadratisch zur Anzahl der Empfänger.
Die asymmetrische Kryptographie hält für all dies eine Lösung parat, die ich im folgenden darlegen und erläutern werde.

3 Prinzip

Im Gegensatz zur symmetrischen Verschlüsselung arbeitet die asymmetrische Verschlüsselung, auch Public-Key-Verfahren genannt, mit Schlüsselpaaren. Zu Beginn des Verfahrens steht der Empfänger, welcher zwei Schlüssel erzeugt. Den Private-Key zum Decodieren und den Public-Key zum Codieren. Während der Private-Key unter Geheimhaltung beim Empfänger verbleibt, wird der Public-Key für jeden frei zugänglich veröffentlicht. Hierbei ist es wünschenswert die Verbreitung so weit wie möglich auszudehnen, so wird das Risiko eines gefälschten Public-Keys vermindert. Allerdings sollte er sich stets im Bereich der eigenen Kontrolle befinden.
Möchte nun ein Sender S eine codierte Nachricht an Empfänger E senden, so greift S auf den Public-Key von E zu und verschlüsselt damit seinen Klartext. Die dadurch entstandene Chiffre kann bedenkenlos über unsichere Kanäle übermittelt werden. E kann sie anschließend mit seinem Private-Key decodieren und lesen.[6]

4 Vorteile asymmetrischer Kryptographie

Unter der Verwendung des Public-Key-Verfahren entfällt das Schlüsselverteilungsproblem, da der Public-Key für jeden frei zugänglich ist. Des Weiteren ist der Private-Key nur bei einer Person hinterlegt. Das bedeutet: Nur eine Person muss ihn geheim halten. Nur eine Person muss ihn vor potentiellen Angreifern schützen. Nur die Verwahrung, nicht aber die Übermittlung müssen gesichert werden.
Das Arbeiten mit Schlüsselpaaren erlaubt einen simpleren Umgang bei mehreren Kommunikationspartnern. So wächst die Anzahl der Schlüssel(paare) lediglich linear mit der Anzahl teilnehmender Personen. Das Public-Key-Verfahren wird allgemein als sicher betrachtet. Ein Versuche der Dechiffrierung ohne den Private-Key ist sehr zeitintensiv. Ein 193-stelliger Schlüssel konnte nach einem Jahr gehackt werden. Für die üblicherweise 300- stelligen Schlüssel blieb die Arbeit bislang fruchtlos. Ein weiterer Vorteil des Public-Key-Verfahrens ist die Möglichkeit der Authentifikation mittels elektronisch erstellter Unterschriften.[1] Die sogenannte digitalen Signaturen werden in Kapitel drei weiter vertieft.

5 Grundlagen für die Sicherheit

Um die Sicherheit für den geheimen Datenaustausch zu gewährleisten dürfen unter keinen Umständen Rückschlüsse vom Öffentlichen auf den privaten Schlüssel zu ziehen sein. Um dies zu Realisieren bedient sich die asymmetrische Kryptographie der Einwegfunktion mit Falltür.
Die Einwegfunktion stellt dabei eine simple Rechnung dar, deren Invertierung sich allerdings keinem exakt zielführenden Algorithmus unterwirft. Allerdings lässt sich der Rückweg mit einem "Geheimnis", der sogenannten Falltür, abkürzen. Die Falltür ist in diesem Fall der Private-Key. Mit der Entdeckung eines Algorithmus zur schnellen, eindeutigen Umkehrung fände die Sicherheit der asymmetrische Kryptographie ihr Ende.
Angewendet wird die Einwegfunktion mittels des Faktorisierungsproblems oder des Diskreten-Logarithmus-Problems. Das Rucksack- oder Untersummenproblem, auf dem z.B die Merkle-Hellmann- und Chor-Riverst-Kryptosysteme beruhen, stellte sich als lösbar heraus und wird dementsprechend nicht weiter verwendet.[2] Um die Sicherheit zu gewährleisten ist eine ausreichend große Schlüssellänge erforderlich.

5.1 Faktorisierungsproblem

Zwei Zahlen miteinander zu multiplizieren bedeutet keinerlei Aufwand. Allein aus dem Produkt die Faktoren zu bestimmen ist nicht in praktikabler Zeit realisierbar. Um sicherzustellen, dass die Faktoren eindeutig sind multiplizieren wir Primzahlen. Je größer die zu multiplizierenden Zahlen, desto größer der Schlüssel, desto länger dauert die Faktorisieren. Beim RSA-Verfahren beispielsweise wird zu einer Schlüsselgröße von mindestens 1024 Bit geraten. Entstanden durch zwei Primzahlen der Größe 512 Bit, das entspricht einer ca. 155-stelligen Zahl. Die hier beschriebene Einwegfunktion bildet die Grundlage für das RSA-Verfahren.[7]

Faktorisierungspoblem anhand des RSA-Verfahren[8]

Satz von Euler

Der Satz von Euler besagt, wenn $\{m,n\} \in \mathbb{N}$ und $m \perp n$ so gilt

$$m^{\varphi(n)} mod(n) = 1$$

mit $\varphi(n) =$ Anzahl der zu n teilerfremden natürlichen Zahlen, kleiner n (die Anzahl der Zahlen die kleiner n sind und deren größter gemeinsame Teiler ($ggT.$) mit n gleich 1 ist). Mit dem teilerfremden Verhalten von Primzahlen $\{1...(p-1)\} \perp p$ folgt hieraus

$$\varphi(p) = (p-1)$$

für das Produkt zweier Primzahlen gilt analog

$$\varphi(p*q) = (p-1)*(q-1)$$

Euklidischer Algorithmus

Unter Anwendung des euklidischen Algorithmus ermitteln wir den größten gemeinsamen Teiler ($ggT.$) zweier Natürlicher Zahlen a und b. Hierbei wird der Modulo angewandt bis der Rest 0 ergibt . Zudem benötigen wir zwei weitere Variablen x und y
Zu Beginn werden den Variablen x und y die Werte von a bzw. b zugeschrieben. Anschließend folgt die Rechnung $r = x\ mod(y)$

Ist $r \neq 0$

$\Rightarrow\ x = y$

$\Rightarrow\ y = r$

$\Rightarrow$ Modulo wiederholen

Ist $r = 0$

$\Rightarrow\ ggT.(a,b) = y$

Schlüsselerzeugung

Anfangs werden zwei Primzahlen p und q festgelegt. Die Größenordnung sollte für eine sichere Verschlüsselung bei ca. 10^{200} liegen. Außerdem sollten sich die Stellenzahlen von p und q in der Dezimaldarstellung deutlich unterscheiden. Andernfalls sind sie durch gezielte Suchalgorithmen (z.B. mit dem Verfahren der Differenz der Quadrate) relativ leicht zu ermitteln. Im nächsten Schritt wird das RSA-Modul $m = p*q$ berechnet und $\varphi(n)$ nach dem Satz von Euler gebildet.

$$\varphi(n) = (p-1)*(q-1)$$

Nun folgt folgt die Berechnung der eigentlichen Schlüssel:
Dafür wird zunächst der Verschlüsselungsexponent i nach folgenden Kriterien gewählt.

- $1 < i < \varphi(n)$
- $i \perp \varphi(n)$

Um diese Bedingungen zu erfüllen ist die Anwendung des euklidischen Algorithmus zur Bestimmung des Verschlüsselungsexponenten auszuführen. Anschließend bestimmen wir den Entschlüsselungsexponent g, sodass folgendes erfüllt wird:

$$i * g \ mod(\varphi(m)) = 1$$

was äquivalent ist zu:

$$\frac{(g * i) - 1}{\varphi(m)} = \mathbb{N}$$

Der Public-Key setzt sich aus dem Verschlüsselungsexponenten i und dem RSA-Modul m zusammen. Der Private-Key aus dem Entschlüsselungsexponenten g und dem RSA-Modul m
Dabei gilt es folgendes zu beachten:

- Die Primzahlen p und q müssen, wie der Private-Key, geheim gehalten werden, können nun aber auch vernichtet werden.
- Wird dasselbe n für verschiedene Benutzer mehrfach verwendet, kann die Nachricht nach Lemma von Bachet entschlüsselt werden.
- Die $ggT.(i-1;p-1)$ und $ggT.(i-1;q-1)$ sind so klein wie möglich zu wählen. Entspricht das verschlüsselte Zeichen dem Ursprünglichen, wird es als Fixpunkt der Chiffre bezeichnet. Offenkundig erleichtert eine hohe Anzahl von Fixpunkten die unautorisierte Entschlüsselung, was es unter allen Umständen zu verhindern gilt.

Codieren

$$\text{Klartext}^{\text{i}} \ mod(\varphi(m)) = \text{Chiffre}$$

Da nur Zahlen verschlüsselt werden können, wird der zu verschlüsselnde Text mittels ASCII-Code in eine Zahlenfolge konvertiert. Zur Realisierung mittels PC werden diese Zahlen wiederum in einen Binärcode umgewandelt. Dieser wird dann in Blöcken der Reihe nach verschlüsselt.

Decodieren

$$\text{Chiffre}^{\text{g}} \ mod(\varphi(m)) = \text{Klartext}$$

5.2 Diskreter Logarithmus Problem

Wenn p eine beliebige Primzahl ist, ist der endliche Körper M eine Menge von ganzen Zahlen mit endlich vielen ganzzahligen Elementen von 0 bis $(p-1)$. Ergibt sich durch Addition oder Multiplikation der Elemente ein Wert außerhalb der Menge M, wird der Divisionsrest, bei Division durch p, ermittelt. Dieser Rest befindet sich wieder innerhalb des endlichen Körpers M. Dies bezeichnen wir als den Modulo einer Primzahl.
Ebenso kann dabei Potenzrechnung angewendet werden.
Zu einer beliebigen Zahl g, aus M, ist die Potenz x $(x \in M)$ zu ermitteln. Umgesetzt wird dies mittels der Berechnung von $g^{\mathrm{x}}\ mod(p)$. Dabei existieren einerseits Zahlen, welche lediglich einige bestimmte Werte der Menge M annehmen. Andererseits einige wenige Zahlen, die alle Werte von 0 bis M übernehmen können. Diese sind als Primitivwurzel Modulo p deklariert, überdies auch Generator oder Erzeuger genannt. Zu jeder Primzahl existiert mindestens eine Primitivwurzel. Die zahl g, eine Primitivwurzel von p, als Basis der diskreten Exponentialfunktion führt stets zu einem Ergebnis $\in M$. Da die diskrete Exponentialfunktion alle Werte der von M angenommen werden, ist sie mittels des Diskreten Logarithmus invertierbar.
Die diskrete Exponentialfunktion $g^{\mathrm{x}}\ mod(p)$ ist, im Gegensatz zum diskreten Logarithmus auch für große Exponenten effizient zu berechnen. Bis heute existiert kein effizienter Algorithmus zur Berechnung des Exponenten anhand der Basis g, dem Modulo p und dem Ergebnis. In Elliptischen-Kurven-Kryptosystemen werden anstelle der Multiplikation und Potenzierung auf dem endlichen Körper, Punktaddition und Skalarmultiplikation auf elliptischen Kurven angewendet.

Diffie-Hellmann Schlüsselaustausch

Zu Beginn einigen sich die Kommunikationspartner auf eine beliebige Primzahl p und eine natürliche Zahl g. Wobei gelten muss. $g < p$. Ein sicher Transport dieser Werte ist nicht notwendig.

	Partner A	Partner B
1.	wählt eine natürliche Zahl $x < p$	wählt eine natürliche Zahl $y < p$
2.	berechnet $a = g^{\mathrm{x}}\ mod(n)$	berechnet $b = g^{\mathrm{y}}\ mod(n)$
3.	sendet a an Partner B	sendet b an Partner A
4.	berechnet $k_1 = b^{\mathrm{x}}\ mod(n)$	berechnet $k_2 = a^{\mathrm{y}}\ mod(n)$

Nun gilt $k = k_1 = k_2$
Ein Angreifer der die Kommunikation überwacht hat ist nicht in der Lage k zu berechnen. a bzw. b repräsentieren den Public-Key, x bzw. y den Private-Key.[**spring**, 9]

6 Nachteile asymmetrischer Kryptographie

- Ist die Nachricht an mehrere Empfänger zu versenden entsteht eine erhöhter Aufwand, da die Nachricht für jeden Empfänger individuell verschlüsselt werden muss.
- Die Sicherheit ist nur gewährleistet solange kein eindeutig zielführender Algorithmus zur schnellen Invertierung der Einwegfunktion existiert. Der Quantencomputer hätte den selben Effekt, auch ohne einen entsprechenden Umkehr-Algorithmus.
- Man-In-The-Middle-Attack
- Benötigt große Schlüssellänge[4]
- Hohe Rechenzeit
- Die verwendeten komplexen mathematischen Verfahren bieten mehr Ansätze für die Kryptoanalyse
- Authentifizierungsproblem[1]

Man-In-The-Middle-Attack
Durch den "Man In The Middle" kommt es zu einem Verteilungsproblem der Publik-Keys. Der Angreifer klinkt sich zwischen die Kommunikationspartner, indem er die übermittelten Public-Keys abfängt und durch seinen Eigenen austauscht. Auf diese Weise ist es dem Hacker möglich die Datenübertragung unbemerkt zu überwachen und zu verändern. Gelöst wird diese Problematik durch die Authentisierung mittels Zertifikaten.

Der Schlüsselaustausch kann nur durch die gegenseitige Authentisierung der Kommunikationspartner vor dem Man-In-The-Middel-Attack geschützt werden.

II

Hybride Verschlüsselungsverfahren

In der hybriden Kryptographie vereinen sich die Vorteile der symmetrischen und asymmetrischen Verschlüsselung. Mittels symmetrischer Verfahren wird die eigentliche Datei codiert. Der dafür gültige symmetrische Schlüssel wird asymmetrisch codiert. Somit bieten die hybriden Verfahren eine schnellere Durchführung mit dem Sicherheitsniveau und dem Schlüsselaustausch des Public-Key-Verfahrens. Mit dieser Verfahrenskombination wird das Schlüsselverteilungsproblem der symmetrischen Kryptographie umgangen, während ihre Geschwindigkeit, auch bei großen Datenmengen, beibehalten wird. Genutzt wird dieses Prinzip bei Netzwerkprotokolle wie SSL/TLS, dem Verschlüsseln von E-Mails oder dem Erstellen digitaler Signaturen.

Schlüsselverwaltung

Analog zur asymmetrischen Kryptographie erzeugt zunächst jeder Empfänger sein Schlüsselpaar, anschließend folgt der Austausch des Public-Key. Durch das Verwenden von Schlüsselpaaren muss lediglich das Authentifikationsproblem gelöst werden.Die Anforderung der sichern Übertragung entfällt.
Der Session-Key wird für jede Verschlüsselung neu generiert. Gelangt dieser in unautorisierte Hände gefährdet das lediglich einen Sitzungsschlüssel.[10]

1 Kryptographische Protokolle - SSL/TLS

Kryptographische Protokolle werden in den Bereichen Kommunikations- und Netzwerktechnik breit gefächert eingesetzt. Durch das Zusammenfassen verschiedener kryptographischer Aspekte werden Probleme der Verschlüsselung bei der Kommunikation von Daten eliminiert. Zumeist werden Verfahren zur Schlüsselerzeugung, zum Schlüsselaustausch und Datenverschlüsselung kombiniert. Vielfach werden auch Integritätskontrolle und Authentifikation eingeflochten.
SSL dient der Authentifikation und Verschlüsselung von Internetverbindungen. Definiert wird das kryptographische Protokoll Secure Socket Layer zwischen einem Anwendungs- und einem Transportprotokoll. SSL kann zum Absichern von TCP-Diensten, wie zB. HTTPS, POPS, IMAPS, FTPS, SMTPS etc. genutzt werden.
Dabei ist die wohl häufigste Anwendung der gesicherten Abruf vertraulicher Daten über HTTP und deren Übermittlung an den HTTPS Server. Die Funktion umfasst die Authentisierung des kontaktierten Servers mittels Zertifikaten und der Verschlüsselung der Kommunikation zwischen Server und Client. SSL gilt als Standardprotokoll, bzw. als Standarderweiterung für Anwendungsprotokolle ohne Verschlüsselte Verbindung. Ursprünglich entwickelt wurde SSL während der neunziger Jahre von Netscape.

Die Internet Engineering Task Force (IETF) übernahm ab Version 3.0. Mittels einiger Änderungen wurde ein zu SSL zum inkompatiblen Transport Layer Security (TLS), der SSL Version 3.1. Beide Begriffe werden heutzutage äquivalent verwendet.

2 Anwendungsbeispiel HTTPS

Für HTTP stellt SSL eine optionale Sicherheitskomponente für die Verarbeitung vertraulicher Daten dar. Daran beteiligt sind: Der Server mit seinem persönlichen Schlüsselpaar, eine Zertifizierungsstelle und dem Client. Überdies verfügt jede SSL-geschützte Homepage über eine eigene IP-Adresse.

2.1 Prinzip

Die Funktion von SSL setzt sich aus folgenden Teilfunktionen zusammen:

- **Zertifizierung**
  Die Zugehörigkeit des angegebenen Public-Keys wird anhand von Zertifikaten, ausgestellt durch eine Zertifizierungsstelle, authentisiert. Ohne diese Zertifizierung ist die Sicherheit von SSL hinfällig.

- **Authentifizierung des Servers**
  Anhand von Zertifikaten wird die Identität des Servers auf ihre Echtheit geprüft.

- **Validierung**
  Das übermittelnde Zertifikat wird über die Wurzelinstanz der CA auf Vertrauenswürdigkeit überprüft.

- **Verschlüsselte Übertragung**

2.2 Zertifizierung

Zu Beginn des Verfahrens der Public-Key-Zertifikate lässt sich der Server-Betreiber durch eine Zertifizierungsstelle oder Certification Authority (CA) ein Zertifikat ausstellen. Nach Überprüfung der Identität des Antragstellers wird das erstellte Zertifikat von der CA mit dem eigenen Private-Key signiert. Somit ist die Echtheit der Daten bestätigt. Auch für CA ist eine Zertifizierung über einen weiteren CA Instanz vonnöten. Diese Zertifizierungsabfolge endet bei einer sogenannten Wurzelinstanz, die sich selbst zertifiziert. Bezeichnet wird dieses Hierarchie-System auch als Public-Key-Infrastructure (PKI). Beispiele für Wurzelzertifikate sind die Deutsche Telekom, Root, CA 2 oder die PCA-1-Verwaltung vom BSI für Behörden und öffentliche Institutionen.
SSL verwendet eine Public-Key-Infrastructure nach X.509v3, dabei wird die Identität an den Public-Key geknüpft, zudem ist ein Ablaufdatum im Zertifikat enthalten. Zertifikate können beispielsweise auf einem Webserver hinterlegt werden, sodass es dem Client zur Verfügung steht. Mittels des Zertifikats authentisiert sich der Server gegenüber dem Client, durch die Prüfung des Zertifikats wird die Vertrauenswürdigkeit des Servers in Erfahrung gebracht.
Die Zertifikate können in drei Kategorien unterteilt werden. Diese unterscheiden sich hinsichtlich des Validierungsaufwandes und versichern verschiedene Echtheitsstufen.

- Domain-Validated-Zertifikat (DV-SSL)
- Organisation-Validation-Zertifikat (OV-SSL)
- Extended-Validation-Zertifikat (EV-SSL)

DV- und EV-Zertifikate stellen dabei den meist genutzten Anteil. Den EV-Zertifikaten wird eine höhere Vertrauenswürdigkeit zugeschrieben, zugleich treten infolge des enormen Prüfaufwand hohe Kosten auf. Gegensätzlich dazu sind DV-Zertifikate kostengünstig, mitunter sogar kostenfrei.

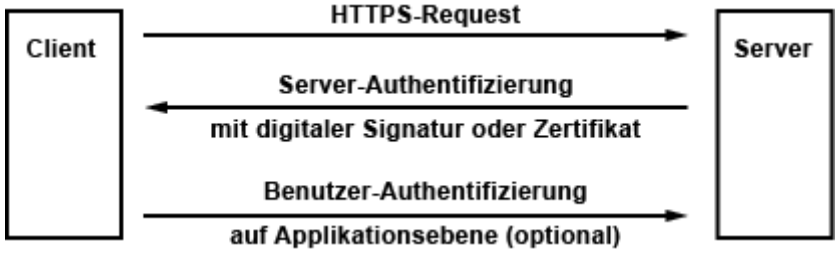
2.3 Validierung des Zertifikats

Durch die Prüfung eines Zertifikats kann die Identität, ohne den Austausch von Authentifizierungsinformation (zB. Schlüssel) bestätigt werden.

1. Der Client bzw. Browser prüft die Vertrauenswürdigkeit des CA. Ist für den entsprechenden CA ein Wurzelzertifikat im Browser hinterlegt, gilt er als vertrauenswürdig.
2. Der Browser kontaktiert die angegebene Validierungs- bzw. Zertifizierungsstelle. Diese prüft die Gültigkeit des Zertifikats und übermittelt die Antwort an den Browser.

Handelt sich um ein bereits bekanntes Zertifikat, ist die Prüfung nicht zwingend erforderlich. Die Validierung anderer Protokolle funktioniert grundsätzlich auf die selbe Weise.

2.4 Ablauf der Authentifizierung



Zu Beginn der Authentifizierung sendet der Client eine HTTPS-Request an den Server. Diesen Vorgang bearbeitet SSL mittels asymmetrischer Kryptographie. Der Server beantwortet die Anfrage des Client durch das Übermittel des Public-Keys und eines Zertifikats. Die anschließende Prüfung der übermittelten Daten, auf ihre Vertraulichkeit, bestätigt die Authentizität des Webservers gegenüber dem Client. Anschließend generiert der Browser einen symmetrischen Schlüssel, den Session Key, codiert ihn mittels des Public-Key des Servers und überträgt ihn an den Server. Nach der Entschlüsselung der Daten,durch den Private-Key, wird der Session Key für die symmetrische Kryptographie der anschließend aufzubauenden Verbindung verwendet. SSL versichert uns so eine abhör- und fälschungssichere Verbindung. In wie weit dieses Versprechen erfüllt wird folgt im anschließenden Abschnitt.
Über die Dauer der Verbindung werden immer neue Schlüsselvereinbarung getroffen, sodass eine eventuelle Störung der Verbindung zeitlich stark begrenzt ist.
Die SSL Spezifikation bietet zusätzlich die Option der beidseitigen Authentifizierung mittels Signatur. Lediglich bei einem SSL-VPN die Authentifizierung des Client gefordert.

3 Sicherheit von SSL/TLS

Seit dem NSA Skandal 2013 ist SSL offiziell als unsicher deklariert. Unter Einsatz einer Authentifikation, deren Sicherheit fragwürdig ist, kann die Güte der Verschlüsselung allenfalls als zweitklassig betrachtet werden.
Die Problematik basiert dabei auf dem hierarchischen Vertrauenskonzept, der Public-Key-Infrastructure (PKI). Wenn sich dem Datenhunger der Geheimdienste kein noch so milliardenschweres Unternehmen entgegenstellen kann, bestehen begründete Zweifel an der Vertrauenswürdigkeit der Zertifizierungsstellen. Letztendlich bewegen wir uns trotz Authentifikation und Verschlüsselung unsicher.[11]

Sicherheit von HTTP

In einer kürzlich veröffentlichten Studie wird berichtet: Sicherheitssoftware wie Antiviren-Programme und Firmen-Proxies überwachen die gesicherte HTTPS-Verbindung um mögliche Schadsoftware zu überführen. Was eigentlich schützen soll verursacht nun beträchtliche Probleme bei der Sicherheit. 10 % der untersuchten Verbindungen wurden als unterbrochen und damit nicht vertrauenswürdig diagnostiziert.[12]

III

Digitale Signatur

Zur Lösung des Authentifizierungsproblems und Wahrung der Integrität wir das Verfahren der digitalen Signatur angewandt. So wird gewährleistet, dass die übertragen Daten tatsächlich vom Empfänger stammen und niemand die Daten lesen oder verändern konnte. Die digitale Signatur, eine verschlüsselte Information, die an ein Dokument angehängt wird, ist gleichbedeutend mit einer analogen Unterschrift.
Die digitale Signatur funktioniert grundlegend wie die Umkehrung der asymmetrischen Kryptographie. Der Private-Key wird hier nicht zum Entschlüsseln, sondern zum Hinzufügen der Signatur verwendet. Unter Einsatz des Public-Keys kann der Empfänger nun die Unversehrtheit und den Ursprung der Daten überprüfen.
Den folgenden Anforderungen sind digitale Signaturen gerecht zu werden:

- **Nicht wiederverwendbar**
  darf nicht ”einfach” auf ein anderes Dokument übertragbar sein
- **Unveränderbar**
  unrechtmäßige Änderungen müssen erkennbar sein, sowohl bei der Signatur als auch im Dokument
- **Eindeutig**
  eine digitale Signatur ist eindeutig zu ihrem Ursprung zurückzuverfolgen
- **Universell**
  jedem steht die Überprüfung zur Verfügung

Des Weiteren ist es notwendig die Signatur permanent zu prüfen.

1 Einweg-Hash-Funktion

Wird die digitale Signatur blockweise erstellt, so besitzt sie mindestens die selbe Größe der Nachricht. Dies bedeutet einen enormen Aufwand an Rechenleistung, sowohl für das Signieren als auch für das Prüfen. Zur Minderung des Aufwands wird deshalb das Signaturverfahren nur auf eine aus der Nachricht gebildete Prüfsumme angewendet. Das Verfahren zur Erstellung der Prüfsumme wird unter dem Namen Hash-Funktion geführt. Ihr Ergebnis ist als Hash(-Wert) benannt. Es gibt eine Vielzahl von Hash-Funktionen, die unterschiedlichen Anforderungen gerecht werden.

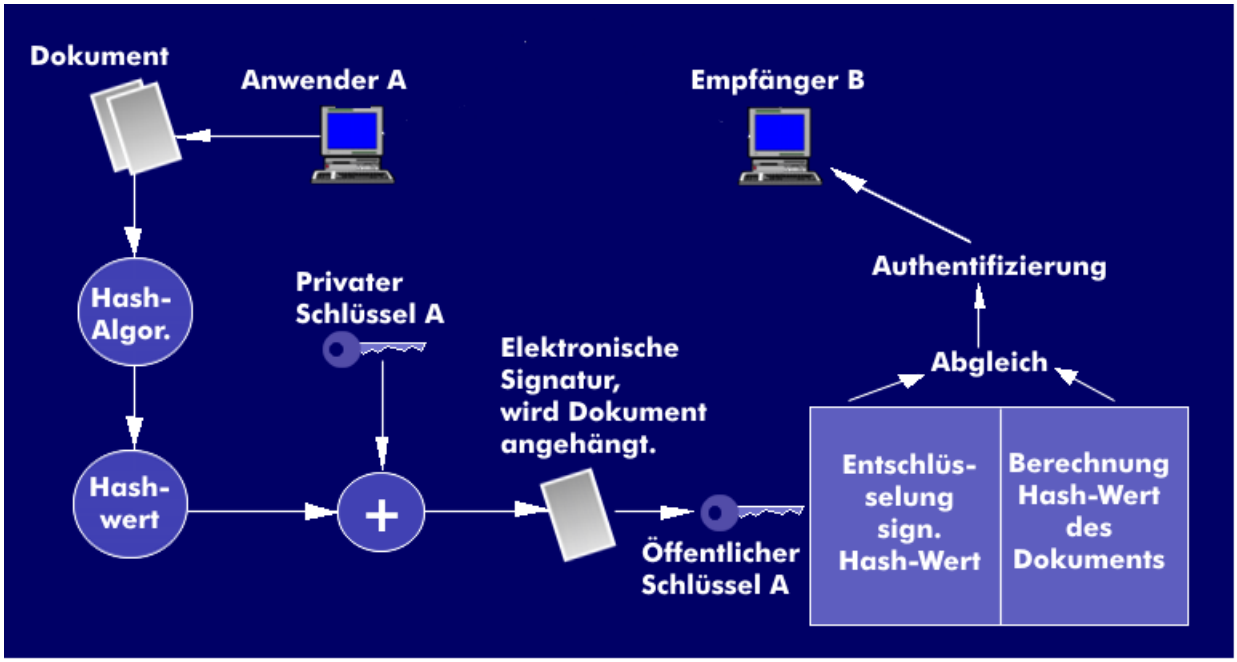
Anforderungen an kryptographische Hash-Funktionen

- **Eindeutig**
  eine bestimmte Zeichenkette erzeugt immer den gleichen Hash.

- **Irreversibel**
  Vom Hash ist keine Rückführung auf die originale Zeichenfolge realisierbar.

- **Kollisionsresistent**
  Unterschiedliche Zeichenfolgen führen immer zu unterschiedlichen Hashs. Da die Menge an Hashs begrenzt ist, müssen wir uns mit Funktionen begnügen, welche zumindest eine hohe Kollisionsresistenz aufweisen. Die Verfahren MD5 und SHA-1 zählen nicht dazu.[13]

2 Prinzip der Digitale Signatur



Zunächst ermittelt der Sender den Hash der zu übertragenden Daten und verschlüsselt diesen mit seinem Private-Key. Das Ergebnis ist die digitale Signatur, welche dem Dokument angehängt wird. Der Empfänger validiert die Signatur. Dafür entschlüsselt er die Signatur mittels des Public-Keys vom Empfänger und vergleicht den ”Klartext” mit dem eigens aus der Datei berechneten Hash. Stimmen beide überein hat eine sichere Übertragung stattgefunden.[14]

3 Signaturverfahren

Neben dem bekanntesten, dem RSA Signaturverfahren auf Basis des Faktorisierungsproblems, existieren noch eine Reihe von Discrete Logarithm Signature Systems (DLSS). Beispielsweise ElGamal oder DSA. Das Angebot übersteigt hier die in der Praxis genutzten Verfahren. Im Vergleich zur Authentifizierung mit RSA zeigen sich folgende Nachteile:

- Zum Erstellen der Signatur wird einen Zufallszahl aus einem Generator benötigt, die ausreichende Qualität der Zufallsgeneratoren ist dabei nicht selbstverständlich.
- Das Prüfen der Signatur beansprucht ca. zehnmal mehr Zeit. Durch eine Implementierung mittels elliptischer Kurven ist das Verfahren wiederum schneller als RSA. Auch RSA kann mit elliptischen Kurven implementiert werden, dies wird allerdings kaum eingesetzt.

3.1 RSA als Authentifikation

Der Sender verwendet seinen Private-Key $\{g;m\}$ um damit die Signatur s zu erzeugen $s = n^{\mathrm{i}}\ mod(m)$. Nun wird die Nachricht n gemeinsam mit der Signatur s versendet. Der Empfänger wiederum prüft, mittels dem Public-Keys $\{i;m\}$ des Empfängers, die Signatur. Ist $s^{\mathrm{i}}\ mod(m) = n$, wurde die Echtheit der Nachricht, sowie Identität des Senders eindeutig bestimmt. Zusätzlich ist es möglich die Signatur mit dem Public-Key des Empfängers zu chiffrieren. Somit ist die Verifikation einzig dem Empfänger vorbehalten.

3.2 DSA - Digital Signature Algorithm

Von der NIST (National Institute of Standards and Technology) beauftragt, entwickelte die NSA den Digital Signature Algorithm. Dieser wurde 1991 veröffentlicht und 1994 standardisiert. Die Notwendigkeit für den Auftrag entstand durch das bis 2000 gültige Patentrecht auf die RSA-Authentifikation. In diesem Verfahren bietet sich lediglich die Option die Signatur zu erstellen und zu verifizieren, allerdings existieren Implementierungen die eine RSA-Verschlüsselung unterstützen. DSA ist anfällig gegenüber schlechten Zufallszahlengeneratoren. Wird der selbe Zufallswert mehrfach verwendet ist eine Berechnung des Schlüssels einfach möglich. Obwohl DSA ein verbreiteter Standard ist gilt es dennoch als unsicher. Zum einen besteht großes Misstrauen gegenüber dem Urheber zum anderen wird das unsichere SHA-1 als Hash-Funktion verwendet.[15]

Oh bedauernswert, du, der zu lenken ersucht, endet im Betrug.
Fürchtest das Chaos- den Kontrollverlust.
Zu meinem Verdruss
Ist es nicht das Fortschrittliche schreiten, das aller Menschen Wege leitet?
Nur der Freiheit schwingen lassen uns geistige Höhen erklimmen.
Nur der unverstellte blick verspricht wahres Glück
Oder fürchtest du gar dadurch du wirst zum Überfluss ?

Literatur

[1] URL: https://www.philipphauer.de/info/info/asymmetrische-verschluesselung/.

[2] URL: http://numbersandshapes.net/2012/04/the-chor-rivest-cryptosystem/.

[3] URL: http://krypto.mufuku.de/2006-10-19/symmetrisch-asymmetrisch/+.

[4] URL: http://www.kryptowissen.de/asymmetrische-verschluesselung.html%3E.

[5] URL: http://ddi.cs.uni-potsdam.de/Lehre/e-commerce/elBez2-5/page11.html.

[6] URL: %3Chttps://www.elektronik-kompendium.de/sites/net/1910111.htm.

[7] URL: https://www.elektronik-kompendium.de/sites/net/1910121.htm.

[8] URL: https://www.zum.de/Faecher/Inf/RP/infschul/kr_rsa.html#funktion.

[9] URL: http://altlasten.lutz.donnerhacke.de/mitarb/lutz/security/cryptfaq/alg_tech.html#A7.

[10] URL: http://www.elektronik-kompendium.de/sites/net/1910141.htm.

[11] URL: http://www.elektronik-kompendium.de/sites/net/0908071.htm.

[12] URL: https://www.heise.de/security/meldung/Sicherheitsforscher-an-AV-Hersteller-Finger-weg-von-HTTPS-3620159.html.

[13] URL: http://www.elektronik-kompendium.de/sites/net/1910131.htm.

[14] URL: http://www.itwissen.info/Digitale-Signatur-digital-signature-DSig.html.

[15] URL: http://www.itwissen.info/DSA-digital-signature-algorithm-DSA-Algorithmus.html.