RSA-Kryptosystem

0. Einleitung

RSA ist ein asymmetrisches Verschlüsselungsverfahren, welches im Jahre 1977 von Ronald L. **R**ivest, Adi **S**hamir und Leonard **A**delman entwickelt wurde. Es ist sowohl zum Verschlüsseln als auch zum Signieren geeignet. Das Buch „New Directions in Cryptography" von W. Diffie und M. Hellman legte 1976 den Grundstein zur Verwendung eines Public-Key-Verfahrens. Unter diesem Konzept wurde 1978 erstmals von Rivest, Shamir und Adelman einer der bis heute sichersten Public-Key-Algorithmen vorgestellt. Dazu führte die Erkenntnis, dass die Primfaktorenzerlegung um ein vielfaches aufwendiger ist als die Multiplikation zweier Primzahlen.

Vorteile

- Durch die Nutzung von öffentlichen und privaten Schlüsseln wird das Problem der Schlüsselübertragung behoben
- Es ist nicht möglich, durch Brute-Force-Methoden beide verwendeten Primfaktoren in annehmbarer Zeit zu ermitteln (dauert je nach Größe mehrere Mrd. Jahre)
- Um eine sichere Kommunikation aufzubauen, müssen den Teilnehmern nur die öffentlichen Schlüssel der Gegenstelle bekannt sein
- Durch Erweiterung der Primzahlen um ein paar Bit lässt sich der Algorithmus an die immer steigenden Prozessorleistungen anpassen, und bleibt somit auch zukünftig sicher

Nachteil

Die Sicherheit des RSA-Kryptosystems beruht auf der praktischen Annahme, dass die Faktorisierung immer größerer Primzahlen selbst mit den schnellsten Algorithmen eine ungeheuer lange Zeit braucht und damit praktisch nicht durchführbar ist. Wird jedoch ein Algorithmus gefunden, *p* und *q* schnell aus *n* (bez. als RSA-Modul, aus den zufälligen Primzahlen *p* * *q*) zu folgern, ist der gesamte RSA-Algorithmus geknackt und damit unbrauchbar.

1. Schlüsselerzeugung

Die RSA-Verschlüsselung steht und fällt mit der Umkehrbarkeit der Funktionen. Deshalb setzt man auf Verfahren der Modul-Arithmetik (Division mit Rest). Darin lassen sich gewisse „Einwegfunktionen" finden, die nur mit erheblichem Aufwand umgekehrt werden können.
Es werden zwei zufällig gewählte Primzahlen (bezeichnet als *p* und *q*) ausgewählt und miteinander multipliziert. Der daraus resultierende Wert *n* wird als RSA-Modul bezeichnet.
Es ist wichtig, für *p* und *q* sehr große Zahlen (mit 100 und mehr Dezimalstellen) auszuwählen, um eine Brute Force Attacke unmöglich zu machen.
Nun werden mit unterschiedlichen arithmetischen Funktionen, unter anderem der Euler'sche Satz und später der Euklid'sche Erweiterungsalgorithmus die Schlüssel berechnet.

2. Verschlüsselung

Da nur Zahlen verschlüsselt werden können, muss der Klartext mit dem ASCII-Code zuvor in Zahlenreihen umgewandelt werden. Nun wird der Klartext in 512 Bit-Blöcke aufgeteilt, und mit der Funktion $g = t^c \; MOD \; n$ chiffriert. Der Geheimtext g ist der Rest der Ganzzahlendivision aus dem ursprünglichen Text *t* potenziert mit dem Codierschlüssel (öffentlichen Schlüssel) *c* des Adressaten und *n*.

3. Entschlüsselung

Die Entschlüsselung ist die dementsprechend einfache Umkehrung der Verschlüsselung:
$t = g^d \; MOD \; n$ wobei *d* den privaten Schlüssel des Empfängers darstellt.

Virtual Private Network

0. Einleitung

In erster Linie dient VPN (Virtual Private Network) dazu, Teilnehmer eines privaten Netzwerks (bspw. Heimnetz) an ein anderes privates Netzwerk zu binden (bspw. Firmennetz). Dadurch wird es möglich, sich Zugang zu einem lokalen Netzwerk zu verschaffen, ohne physisch anwesend sein zu müssen. VPN ist ein reines Softwareprodukt, dessen wichtigste Funktion darin besteht, eine verschlüsselte Verbindung über einen sogenannten Tunnel herzustellen. Dies geschieht über die Erstellung einer virtuellen Netzwerkkarte, welche die Datenpakete direkt an die VPN-Software weiterreicht, die diese verschlüsselt und an die VPN-Gegenstelle sendet. Die Anordnung und Anzahl der verwendeten VPN-Knoten spielt dabei keine Rolle, da VPN als eigenständiges logisches Netz arbeitet.

1. Anwendung

Es folgen einige Anwendungsszenarien, für die VPN unter anderem Verwendung findet:

- Netzwerkzugriff für Außendienstarbeiter (Client-to-LAN)
- Verbindung mehrerer Unternehmensstandorte (LAN-to-LAN)
- Umgehung von Zensur
    - Bspw. bei Standort in China oder Nordkorea
- LAN-Spiele über Internet

2. Funktionsweise | Aufbau einer VPN-Verbindung

1. Internetverbindung über normalen ISP wird hergestellt
2. VPN-Client sendet Verbindungsanfrage an VPN-Server
3. Authentisierung beim VPN-Server
4. IPsec-Tunnel (sicherer VPN-Datentunnel) wird geöffnet

Die Standardkonfiguration sieht vor, dass eine direkte Internetverbindung vom Client ins Internet nicht mehr möglich ist, sobald der Tunnel geöffnet wurde. D.h. der Zugriff auf das Internet erfolgt über die Anbindung des am anderen Ende des Tunnels befindlichen Netzwerks. Ist dies nicht der Fall, besteht ein sogenannter Split-Tunnel und die Sicherheit des Clients/der Daten ist nicht gewährleistet (da der Client über das Internet angesprochen werden kann).
Gleichzeitig bekommt der Client eine private IP-Adresse aus dem Netzwerk (Intranet) zugewiesen. Dies ermöglicht es, das firmenseitige Ende des Tunnels mit einer Firewall zu sichern. Nun ist es dem Client möglich, sich von überall auf der Welt, mit (fast) jedem Verbindungsmedium (bspw. auch Satellit) mit dem Firmennetzwerk zu verbinden.

3. Netztopologien

Es bieten sich mehrere Optionen, eine virtuelle Netzwerkverbindung zu arrangieren. Die Auswahl der Methode hängt vom Verwendungszweck der Anwendung ab.

Point to Point

Es wird eine direkte Verbindung zweier „Points“ (Netzwerkknoten) über sogenannte TUN-Schnittstellen hergestellt. Dadurch können diese zwei Punkte entweder direkt miteinander kommunizieren oder zur Anbindung von Netzwerken mit Hilfe von Routing verwendet werden. Verbinden sich mehrere Clients mit einem VPN-Server, so besteht immer eine P2P-Verbindung zwischen Client und Server. Es entsteht eine Stern-Topologie, welche im Fachjargon als Multipoint bezeichnet wird. TUN und TAP Schnittstellen sind virtuelle Netzwerkkarten, welche die empfangenen Daten an die VPN-Software weiterleiten. TUN arbeitet auf OSI-3 und kommuniziert somit mit IP, während TAP ein Ethernetgerät auf Layer 2 simuliert.

Ethernet

Die TAP-Schnittstelle erlaubt es der Clientsoftware, sich direkt in ein Netzwerk einzubinden. Der VPN-Server fungiert hierbei als Switch, welcher die Clients an ein Netzwerk anbindet. Sind mehrere Clients mit einem Server verbunden, entsteht eine Baumstruktur mit dem VPN-Server als Verteiler.

4. Netzwerkanbindung

Routing

Um mehrere IP-Netze verbinden zu können, wird Routing eingesetzt. Das bedeutet, dass die VPN-Knoten auf IP-Ebene arbeiten. Durch Routing ist es möglich, ein VPN einzurichten, welches es ermöglicht, dass zwei private Netzwerke (bspw. 192.168.1.0/24 und 192.168.178.0/24) sich gegenseitig erreichen können. Erfolgt dies über eine Point-to-Point-Verbindung, wird diese durch ein eigenständiges Netz dargestellt (bspw. 192.168.200.0/25). Es ist auch möglich, eine TAP-Schnittstelle für Routing zu verwenden. Dies kommt in der Praxis jedoch eher selten vor.
Die Einrichtung von Routing über P2P lässt sich relativ unkompliziert durchführen und stellt eine effektive Form der VPN Datenübertragung dar. Allerdings unterstützt diese Methode keine Broadcasts zwischen den Netzen, wodurch sich einige Dienste nicht ohne Weiteres nutzen lassen.

Bridging

Das Betriebssystem stellt Funktionen zur Verbindung der realen Netzwerkkarte mit der virtuellen TAP-Schnittstelle zur Verfügung (Windows: *bridge connections*; Linux: *bridge-utils*). Diese Bridge fungiert als Switch auf OSI-Schicht 2 und übergibt die Netzwerkpakete mit Zielrechner sowie Broadcasts an die TAP-Schnittstelle, welcher sie an den VPN-Counterpart weiterleitet. Dies ermöglicht den Clients, sich so zu verhalten, als wären sie im selben Netzwerk.
Dadurch, dass auch Broadcasts jeden verbundenen Empfänger erreichen, nehmen diese einen großen Teil der Bandbreite in Anspruch, was die VPN-Verbindung verlangsamen kann. Auch ist zu berücksichtigen, nicht mehr als einen DHCP-Server im Netzwerk arbeiten zu lassen, oder ggf. die Adressbereiche zu differenzieren.
Bei der Verwendung einer TAP-Schnittstelle mit Verzicht auf Bridges erhält man ein Netzwerk, welches getrennt von den eigentlichen Netzwerkverbindungen des Servers und der Clients arbeitet und trotzdem die Vorteile des Broadcastings mit sich bringt.

5. Open VPN

OpenVPN ist eine der am weitesten verbreiteten VPN-Lösungen. Lizenziert unter GNU GPL und damit Quelloffen. Durch dessen Plattformunabhängigkeit ist OpenVPN für die breite Masse der netzwerkfähigen Geräte verwendbar.
Es befinden sich 3 mögliche Authentifizierungsvarianten in dessen Repertoire:

Pre-shared Key:

- Statischer Schlüssel, dient zum ver- und entschlüsseln
- Darf nicht verloren gehen oder kompromittiert werden

Benutzer/Passwort:

- Statischer Key ist einem User zugewiesen
- anfällig für Man-In-The-Middle-Attacken1

Zertifikatsbasiert:

- Authentifizierung über das TLS-Protokoll mit privaten und öffentlichen Schlüsselpaaren
- Server und Clients müssen ein gültiges, von einer bekannten Zertifizierungsstelle ausgestelltes Zertifikat besitzen.

[1] Angreifer befindet sich logisch zwischen Sender und Empfänger; kann Daten einsehen und manipulieren.

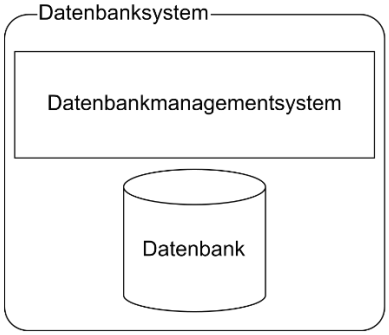
Datenbanken

0. Einleitung

Eine Datenbank, auch Datenbanksystem (DBS) genannt, ist ein System zur elektronischen Datenverwaltung. Die wesentliche Aufgabe eines DBS ist es, große Datenmengen effizient, widerspruchsfrei und dauerhaft zu speichern und benötigt Teilmengen in unterschiedlichen, bedarfsgerechten Darstellungsformaten für Benutzer und Anwendungsprogramme.

1. Komponenten eines Datenbanksystems

Das Datenbanksystem ist das ausgeführte Datenbankmanagementsystem zusammen mit den zu verwaltenden Daten der Datenbank. Ein Datenbanksystem gewährleistet die dauerhafte Speicherung sowie die Konsistenz der Nutzdaten. Es bietet für die benutzenden Datenbankanwendungen mit dem Datenbankmanagementsystem Schnittstellen zu Abfrage, Auswertung, Veränderung und Verwaltung dieser Daten.



2. Datenbankmanagementsystem - Anforderungen

1. **Persistenz**
   Daten sollen dauerhaft gespeichert werden und zu einem späteren Zeitpunkt wieder aufrufbar sein.
2. **Anlegen von Datenschemata**
   Daten haben je nach Kontext unterschiedliche Bedeutungen. Ein Schema (z.B.: eine Tabelle) stellt den Zusammenhang zwischen Daten und Kontext her.
3. **Einfügen, Ändern und Löschen von Daten**
   Möglichkeit Daten in das Datenschema einzutragen, zu ändern oder auch wieder zu löschen.
4. **Lesen von Daten**
   Es muss möglich sein Daten aus der Datenbank wieder aufzufinden.
5. **Integrität und redundanzfreie Datenhaltung**
   Die Möglichkeit, dass ein Datum, welches an mehreren Stellen benutzt wird, nur an einer Stelle hinterlegt ist, aber es trotzdem nur einmal geändert werden muss, damit es an allen Stellen geändert wird.
6. **Koordination der parallelen Nutzung**
   Sicherstellung das Integrität der Datenbank bei parallelen Zugriffen nicht verloren geht. Jeder Nutzer muss den Eindruck haben, dass ihm die Datenbank alleine gehört.
7. **Rechteverwaltung**
   Unterschiedliche Benutzer der Datenbank sollen unterschiedliche Berechtigungen haben.
8. **Datensicherung**
   Das DBMS ermöglicht eine Datensicherung des aktuellen Datenbestandes herzustellen und diesen auch wieder in das System zurückzuspielen.
9. **Katalog**
   Möglichkeit die Struktur des gesamten Systems bestimmten Benutzern zugänglich zu machen. (Datenschema, Nutzerrechte, usw.)

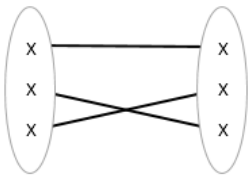
Das Datenbankmanagementsystem besteht aus 3 Schichten:

1. **Externe Schicht**
   Jede Benutzergruppe sieht den Ausschnitt der Datenbank, der für sie von Bedeutung ist. Die Daten werden so dargestellt, wie es für die Benutzer wünschenswert oder leicht anschaulich ist.
2. **Logische Schicht**
   In der Datenbank sind alle wichtigen Daten zusammengefasst. Um die Datenbank zu erstellen zu können, ist eine Gesamtschau der Daten notwendig. Alle Daten müssen zunächst auf logischer Ebene in Form von Informationseinheiten und deren Beziehungen untereinander beschrieben werden, aber unabhängig von EDV-Gesichtspunkten. Diese Beschreibung der Gesamtheit der Unternehmensdaten nennen wir logische Gesamtschicht.
3. **Physische Schicht**
   Die Daten müssen auf den Speicher so organisiert werden, dass die Zugriffsanforderungen der verschiedenen Benutzer möglichst effizient erfüllt werden können.

3. Beziehungen

Beziehungen geben an wie einzelne Entitäten zueinander „abhängig“ sind (zueinander in Beziehung stehen)

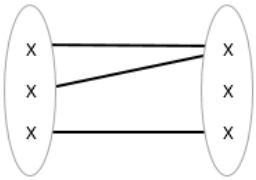
1:1 Beziehung



„Ein Angestellter besitzt eine Keycard.“

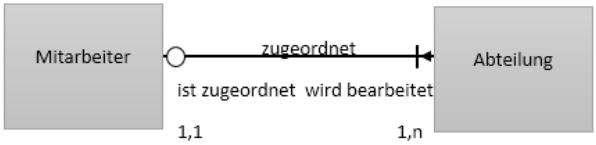
„Eine Keycard gehört einem Angestellten“

N:1 Beziehung

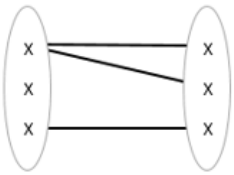


„Ein Mitarbeiter ist **einer** Abteilung zugeordnet.“

„Eine Abteilung ist **einen oder mehrere** Mitarbeiter“

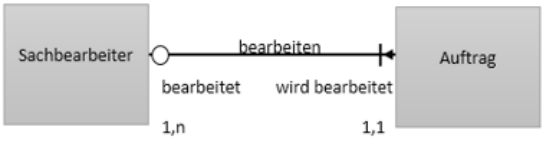


1:N Beziehung

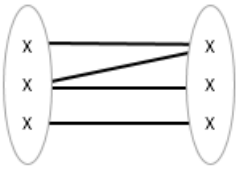


„Ein Sachbearbeiter bearbeitet **einen oder mehrere** Aufträge.“

„Ein oder mehrere Aufträge werden **einem oder keinem** Sachbearbeiter bearbeitet.“

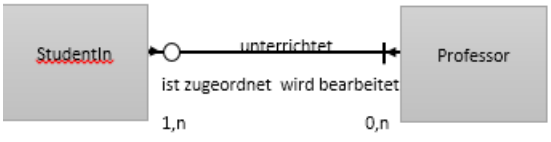


M:N Beziehung



„Ein Student wird unterrichtet von **einem oder mehreren** Professoren“

„Ein Professor unterrichtet **mehrere** Professoren“



SQL – Structured Query Language

0. Einleitung

SQL ist eine Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanken sowie zum Bearbeiten (Einfügen, Verändern oder Löschen) und Abfragen von darauf basierenden Datenbeständen. Vorteil an SQL ist, dass diese Sprache einfach zu lernen ist, da sie sich semantisch an die englische Umgangssprache anlehnt.

1. Relationale Datenbank

Eine relationale Datenbank dient zur elektronischen Datenverwaltung in Computersystemen und beruht auf einem tabellenbasierten relationalen Datenbankmodell.

Beispiel: Datenbanktabelle ***Student***

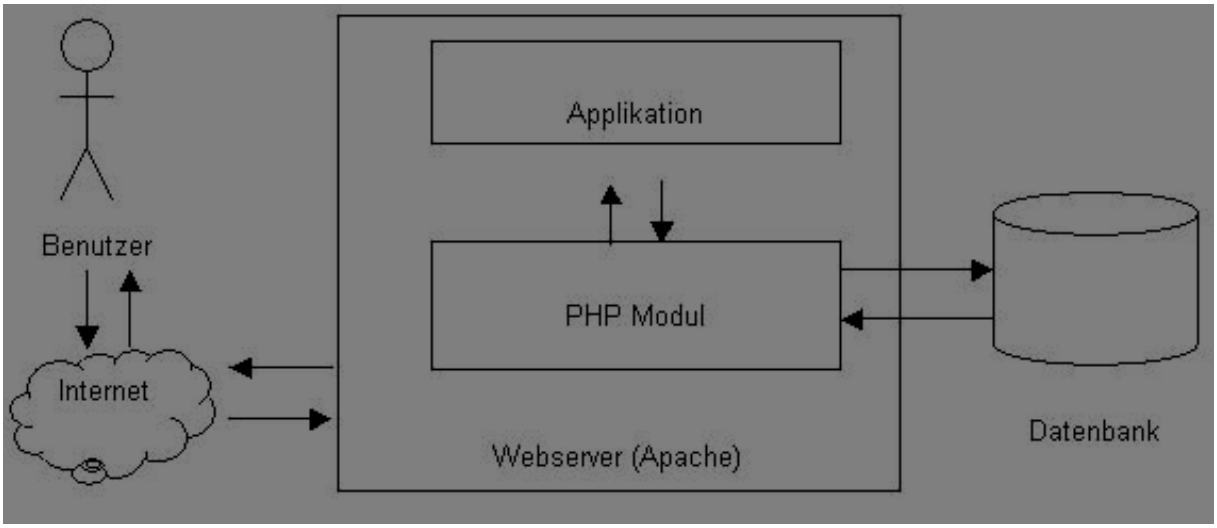
Attribute

ID	Name	Vorname	Studienfach
0	Höfer	Sebastian	Informatik
1	Oepen	Dominik	Informatik

Primärschlüssel Attributwerte

2. SQL - Injections

SQL Injections werden für das Einschleusen von SQL Befehlen in Webapplikationen zum Beispiel PHP, ASP, JSP usw. benötigt. Allerdings können große Sicherheitslücken entstehen, wenn die Nutzereingaben ungeprüft angenommen und in die Datenbank eingetragen werden.



3. Die wichtigsten Befehle:

```
SELECT [ALL | DISTINCT] {spalten | *}
FROM tabelle [alias] [tabelle [alias]] ...
WHERE {bedingung | unterabfrage}]
GROUP BY spalten [HAVING {bedingung | unterabfrage}]]
ORDER BY spalten [ASC | DESC]...];
```

SELECT Wähle die Werte aus der/den Spalte(n) **[mehrfache Datensätze nur einmal]**...
FROM ... **aus** der Tabelle bzw. den Tabellen ...
WHERE ... **wobei** die Bedingung(en) erfüllt sein soll(en) ...
GROUP BY ... und **gruppiere** die Ausgabe von allen Zeilen mit gleichem Attributwert zu einer einzigen...
HAVING ... **wobei darin** folgende zusätzliche Bedingung(en) gelten müssen/muss ...
ORDER BY
[ASC/DESC]
... und sortiere nach den Spalten **[auf- bzw. absteigend]**.