Verschlüsselung

Grundlagen und Anwendungen

Stefan Betz

backspace e.V.

19. April 2012

Inhalt

Warum verschlüsseln?

Gründe seine Daten vor unerwünschtem Zugriff zu schützen

- Schutz bei Diebstahl von Hardware (Notebooks, Datenträger, . . .)
- Schutz bei Überwachung durch VDS, BOFH, Arbeitgeber, . . .
- Schutz bei Strafen, da verschlüsselte Daten eine Auswertung für Ermittlungsbehörden verhindern/erschweren

Varianten

Was man alles verschlüsseln könnte

Dateiverschlüsselung für E-Mails, einzelne Dateien, Backups, ...

Datenträgerverschlüsselung für Datenträger, Container, ...

Transportverschlüsselung für HTTP, IMAP, POP3, SMTP, ...

Kommunikationsverschlüsselung für Instant Messaging, VoIP, ...

Techniken

Wie man verschlüsseln kann

- Symmetrische Verschlüsselung
- Asymmetrische Verschlüsselung
- Hybride Verschlüsselung

Symmetrische Verschlüsselung

Einfachste Technik zur Verschlüsselung

- Einfach, da keine komplexe Verwaltung der Schlüssel erforderlich
- Beliebige Datenmenge als Schlüssel nutzbar
- Gleicher Schlüssel wird für Verschlüsselung und Entschlüsselung verwendet
- Wer dem Schlüssel hat kann die Daten entschlüsseln
- Fehler beim Verteilen von Schlüsseln Kompromittieren das Gesamtsystem
- Komplex bei großen Teilnehmerzahlen

Asymmetrische Verschlüsselung

Aufwändiges Verfahren mit Nachteilen

- Komplexes Verfahren
- Skaliert gut für viele Benutzer
- Aufwändigeres Verwalten von Schlüsseln
- Geringeres Risiko bei der Schlüsselverteilung
- Spezielle Schlüssel erforderlich
- Passwörter können nicht als Schlüssel verwendet werden
- Verschlüsselte Datenmenge steigt proportional zur Anzahl der Empfänger
- Langsames Verfahren

Asymmetrische Verschlüsselung
Bestandteile

Öffentlicher Schlüssel

Dient zur Verschlüsselung und darf/soll/muss überall verteilt werden.

Privater Schlüssel

Dient zur Entschlüsselung und darf unter keinen Umständen verteilt werden.

Passphrase / Mantra

Schützt den privaten Schlüssel bei Verlust. Dies ist Optional, nicht jede asymmetrische Verschlüsselung schützt den privaten Schlüssel!

Hybride Verschlüsselung

Business as usual

- Kombination aus symmetrischer und asymmetrischer Verschlüsselung
- Verschlüsselte Datenmenge enthält pro zusätzlichem Empfänger nur geringen Overhead
- Schneller als asymmetrische Verschlüsselung da Nutzdaten symmetrisch verschlüsselt werden

1. Erzeugen eines zufällig erzeugten symmetrischen Schlüssels
2. Verschlüsseln der Nutzdaten mit diesem symmetrischem Schlüssel
3. Verschlüsseln des symmetrischen Schlüssels mit einem asymmetrischen Verfahren für jeden Empfänger
4. Paketieren des Ergebnisses aus symmetrischer und asymmetrischer Verschlüsselung

Algorithmen
Die Wahl des richtigen Algorithmus

- Nur offene Verfahren sind sicher, was nicht veröffentlicht ist kann auch nicht sicher sein
- Ein unsicheres Passwort macht einen sicheren Algorithmus überflüssig
- Algorithmen sind nur sicher wenn die Software welche diese Implementiert dies ebenfalls ist
- Bei symmetrischer Verschlüsselung mindestens 128 Bit, bei asymmetrischer mindestens 1024 Bit

Signaturen

Wer ist mein gegenüber?

- Stellen sicher das Inhalt nicht verändert wurde
- Stellen sicher das Absender authentisch ist
- Funktioniert nur mit einem Verfahren mit öffentlichen Schlüsseln, da dieser zum überprüfen einer Signatur erforderlich ist

Vertrauensmodelle

Wem gehört der Schlüssel?

- Definieren die Beziehungen von Schlüsseln zueinander
- Definieren wer für die Prüfung der Authentizität zuständig ist
- Definieren was mit kompromittierten Schlüsseln passiert

Dezentrales Vertrauensmodell

Der bekannte vom Neffen der Vater sein Onkel der Bruder...

1. Alice und Bob signieren ihre Schlüssel
2. Alice und Carol signieren ihre Schlüssel
3. Bob vertraut Alice und damit auch Carol

Dezentrales Vertrauensmodell

Eigenschaften, Vorteile, Nachteile

- Jeder Teilnehmer signiert wenn möglich andere Teilnehmer
- Signaturen der Teilnehmer werden auf sog. Keyserver geladen und somit verteilt
- Teilnehmer holen sich regelmäßig die neuesten Signaturen der Server
- Keyserver speichern im Idealfall alle öffentlichen Schlüssel aller Teilnehmer
- Keyserver tauschen die Informationen untereinander aus (Redundanz)
- Was der Keyserver einmal kennt bekommt man dort nie mehr weg (!!!)
- Sicherheit des Gesamtsystems hängt vom Sicherheitsbewusstsein der Teilnehmer ab
- Ermittlungsbehörden können das System nicht angreifen da dezentral
- Höherer Aufwand, da sich jeder Teilnehmer mit dem Konzept beschäftigen muss

Zentrales Vertrauensmodell

Vertrauen durch Geld / Punkte

1. Teilnehmer erstellt ein eigenes Zertifikat und dazu einen CSR (Certificate Signing Request)
2. Öffentlicher Schlüssel und CSR werden an die sog. CA (Certificate Authority) geschickt
3. CA überprüft im Idealfall ob alles OK ist und signiert dann den öffentlichen Schlüssel
4. Teilnehmer kann nun den Schlüssel verwenden
5. Clients mit passendem Zertifikat der CA erkennen eigenes Zertifikat als gültig

Zentrales Vertrauensmodell

Eigenschaften, Vorteile, Nachteile

- Clients haben in der Regel eine vordefinierte Liste an CAs die vertrauenswürdig sein sollten
- Zentrale Stelle (CA) kümmert sich um Signaturen und Überprüfung der Authentizität
- CAs sind meistens gewinnorientierte Unternehmen, das widerspricht dem Sicherheitsgedanken
- CAs neigen dazu Dienstleistungen für Ermittlungsbehörden und Regierungen anzubieten
- Teilnehmer bekommt von dem ganzen zum großen Teil nichts mit, sehr transparent

Unsicherheitsfaktoren

Probleme der Technik

Wer A sagt, muss auch B sagen:

- Wird ein (privater) Schlüssel / Passwort veröffentlicht ist der dadurch verschlüsselte Inhalt öffentlich einsehbar und gilt als Unverschlüsselt.
- Wird ein Algorithmus geknackt müssen damit Verschlüsselte Inhalte neu verschlüsselt werden.
- Wird ein Bug in einer Software zur Verschlüsselung gefunden müssen ggf. neue Schlüssel erzeugt werden (Google: SSH, Debian).
- Das Gesamtsystem steht und fällt mit dem IQ des Anwenders.

Dateiverschlüsselung

Dateiverschlüsselung

Verschlüsseln von Dateien, Archiven und E-Mail

- Wird in der Regel zum Austausch von wichtigen Informationen über unsichere Kommunikationskanäle (E-Mail, FTP, USB-Stick, . . .) verwendet.
- E-Mail ist dank MIME (RFC 2045) nichts anderes als ein Dateiaustausch via SMTP und zählt daher zur Variante der Dateiverschlüsselungen.
- Häufig ist es erforderlich Inhalt an mehrere Empfänger zu senden, daher nur hybride Verschlüsselungstechniken sinnvoll.

Techniken

Gängige Möglichkeiten der Dateiverschlüsselung

- GPG / PGP / OpenPGP
- S/MIME

GPG
Fakten über GPG / PGP / OpenPGP

Vorteile:

- Dezentrales Vertrauensmodell (Web of Trust)
- Guter Support in OpenSource Clients
- Häufig können Dateimanager über das Kontextmenü Dateien damit verschlüsseln

Nachteile:

- Schlechter / Kein Support in ClosedSource Clients, meistens nur über Plugins
- Verschiedene Umsetzungen im Umlauf, teils inkompatibel oder veraltet (z.B. PGP/Inline)

S/MIME
Fakten über S/MIME

Vorteile:

- Guter Support in Mainstream Mailclients (Outlook, Thunderbird, Evolution, . . .)

Nachteile:

- Zentrales Vertrauensmodell durch CA / Firma
- Unzureichender Support für Dateiverschlüsselung

S/MIME vs. GPG in Firmen

Businessfakten für sichere Kommunikation

- Kommunikation mit externen gestaltet sich schwierig da diese in der Regel weder S/MIME noch GPG haben.
- Anbindung externer über sog. Encryption Gateways mit Weboberfläche ist umständlich und führt in der Regel zu wenig Akzeptanz.
- S/MIME bzw. GPG können nicht sinnvoll gleichzeitig verwendet werden, man muss sich für ein Verfahren entscheiden.
- Schulungen führen dazu das die Akzeptanz der eigenen Mitarbeiter wächst.

Datenträgerverschlüsselung

Datenträgerverschlüsselung

Verschlüsseln von Datenträgern

- Wird in der Regel verwendet um unerwünschten Zugriff auf Datenträger effektiv zu verhindern
- Nur komplett verschlüsselte Systeme sind sichere Systeme, da temporäre Dateien gefährlich sind bzw. sein könnten
- Verschlüsselung schützt nur wenn der Datenträger nicht gemounted ist, Angreifer über andere Lücken werden dadurch nicht aufgehalten
- FireWire und andere externe Schnittstellen mit DMA (Direct Memory Access) Kompromittieren das System und müssen/sollten abgeschaltet werden

Datenträgerverschlüsselung

Verschlüsseln von Datenträgern

- Für beste Performance sollte der CPU über AES-NI (Advanced Encryption Standard Instruction Set) verfügen
- Hardwareverschlüsselung von Festplatten ist nicht sicher, oft wird nicht mal Verschlüsselt sondern nur ein Passwort im Controller gesetzt
- BIOS Passwörter sind Snakeoil
- Open Source Produkte sind vertrauenswürdig, Closed Source Produkte können dies niemals sein
- Der unverschlüsselte Bereich (Bootloader) eines einmal entwendeten Notebooks ist nicht vertrauenswürdig

TrueCrypt
Platzhirsch für Windows

- TrueCrypt bietet unter Windows eine Vollverschlüsselung des Systems an
- Brauchbare Lizenz und daher im Quellcode einsehbar. Leider nicht OpenSource gemäß OSI Spezifikation
- Systemverschlüsselung für andere Systeme als Windows nicht vorgesehen.
- Passwörter werden durch PBKDF2 (Password-Based Key Derivation Function) verstärkt

LUKS
Standard unter Linux

- LUKS (Linux Unified Key Setup) baut auf dm-crypt des Kernels auf
- Vorgänger waren u.a. Loop-AES, cryptoloop und viele andere, welche heute nicht mehr verwendet werden sollten
- LUKS bietet 8 Keyslots für Passwörter und Keyfiles
- Passwörter werden durch PBKDF2 (Password-Based Key Derivation Function) verstärkt
- Linux-Only Lösung, andere Betriebssysteme können LUKS nur eingeschränkt oder gar nicht verwenden
- Wird häufig / meistens mit LVM kombiniert um auch SWAP, /tmp und alles außer /boot zu verschlüsseln

Mac OS X
Verschlüsselung ist nicht sexy

An dieser Stelle sollte eigentlich eine Lösung präsentiert werden die zumindest im Quellcode einsehbar ist, gibt es aber scheinbar für dieses stylische System nicht.

Transportverschlüsselung

Transportverschlüsselung

Der (sichere) Weg ist das Ziel

- Wird verwendet um zu verhindern das dritte den Transportweg von Datenverbindungen auswerten können
- Nur sinnvoll in Zusammenarbeit mit Signaturen einsetzbar

Möglichkeiten

Möglichkeiten den Transport zu sichern

- Verschlüsselung des Protokolls via SSL / TLS
- Verschlüsselung via VPN (Virtual Private Network)
- Verschlüsselung via SSH (Secure Shell)

SSL / TLS
Protokollverschlüsselung

- SSL (Secure Socket Layer) und TLS (Transport Layer Security) sind anerkannter Standard für eine Vielzahl an Protokollen
- Zentrales Vertrauensmodell durch Verwendung sog. Certificate Authorities (CA)
- Schützt nur den Transport zwischen genau zwei Punkten, ein Zertifikat bedeutet nicht das ein Anbieter vertrauenswürdig ist
- Gesamtsystem wird aktuell zuverlässig durch inkompetente / kompromittierte CAs zerstört
- Anwendungen haben in der Regel eine eigene Liste an CAs denen standardmäßig getraut wird
- Unzureichender Schutz, da naive Anwender Zertifikatswarnungen einfach weg klicken

VPN
Virtual Private Network

- Stellt eine sichere Verbindung in das eigene Netz her
- Alle Verbindungen können so über ein sicheres Netz geroutet werden, lokale Angreifer z.B. auf LAN Parties haben keine Möglichkeit für einen Angriff
- Auch unverschlüsselte Protokolle können hiermit genutzt werden
- Viele zueinander inkompatible Standards (IPSec, OpenVPN, Anbieterfoo, . . .)

SSH
Secure Shell

- Ursprünglich entwickelt um sicheren Zugang für Textkonsolen zu ermöglichen
- Public Key Verschlüsselung und somit Login ohne Passwort durch geeignetes und sicheres Verfahren möglich
- Gemeinsamer Standard für SSH vorhanden, Support durch jedes Betriebssystem zumindest als Client
- Portforwarding und VPN ermöglichen es unsichere Protokolle zu tunneln

Vielen Dank für die Aufmerksamkeit!