Crypto Wars:
Ende der Ende-zu-Ende-Verschlüsselung ?!

Wie funktioniert Verschlüsselung, wie sicher kann sie sein, welche Angriffsmöglichkeiten und gesetzgeberischen Bestrebungen gibt es?

Mattis Neiling, Datenschutzbeauftragter TU Berlin
m.neiling@tu-berlin.de

Vortrag & Diskussion beim Berliner Admin-Stammtisch am 3. Juni 2021
https://www.flarp.de/

Agenda

	Umfang (ca.)
1. Warm Up	(4 Folien)
2. Wie Verschlüsselung funktioniert (mit einigen Schlenkern)	(12 Folien)
3. Sicherheit & Angriffsmöglichkeiten	(9 Folien)
4. Gesetzgeberische Bestrebungen	(4 Folien)
5. Quellen, Fazit & Diskussion	(7 Folien)
nicht zu vergessen: Die Lückenfüller	(4 Folien)

Agenda

=> 1. Warm Up

2. Wie Verschlüsselung funktioniert
   (mit einigen Schlenkern)
3. Sicherheit & Angriffsmöglichkeiten
4. Gesetzgeberische Bestrebungen
5. Quellen, Fazit & Diskussion

Echte Ende-zu-Ende-Verschlüsselung - e2ee

Anwendungsszenarien

- E-Mail, Messenger
- VoIP
- Videokonferenzen
- Cloud-Speicher
- Websites / HTTPS
- …

Jedoch nicht für:

- VPN

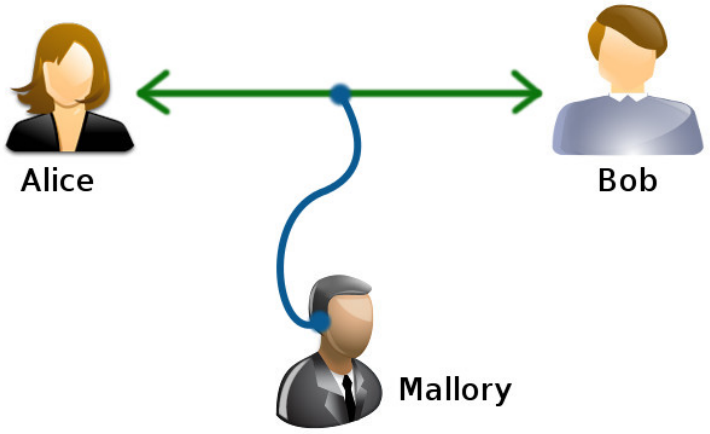
E2ee:

***Ende-zu-Ende-Verschlüsselung** ist fundamental für die **Privatsphäre** im digitalen Zeitalter!*

Alice, Bob & the Man in the Middle

Szenario:

- Kommunikation zwischen A und B
- Angreifer hat Zugriff auf Verbindung „man in the middle“
- Ziel: geheime Datenübertragung
- Lösung: Verschlüsselung

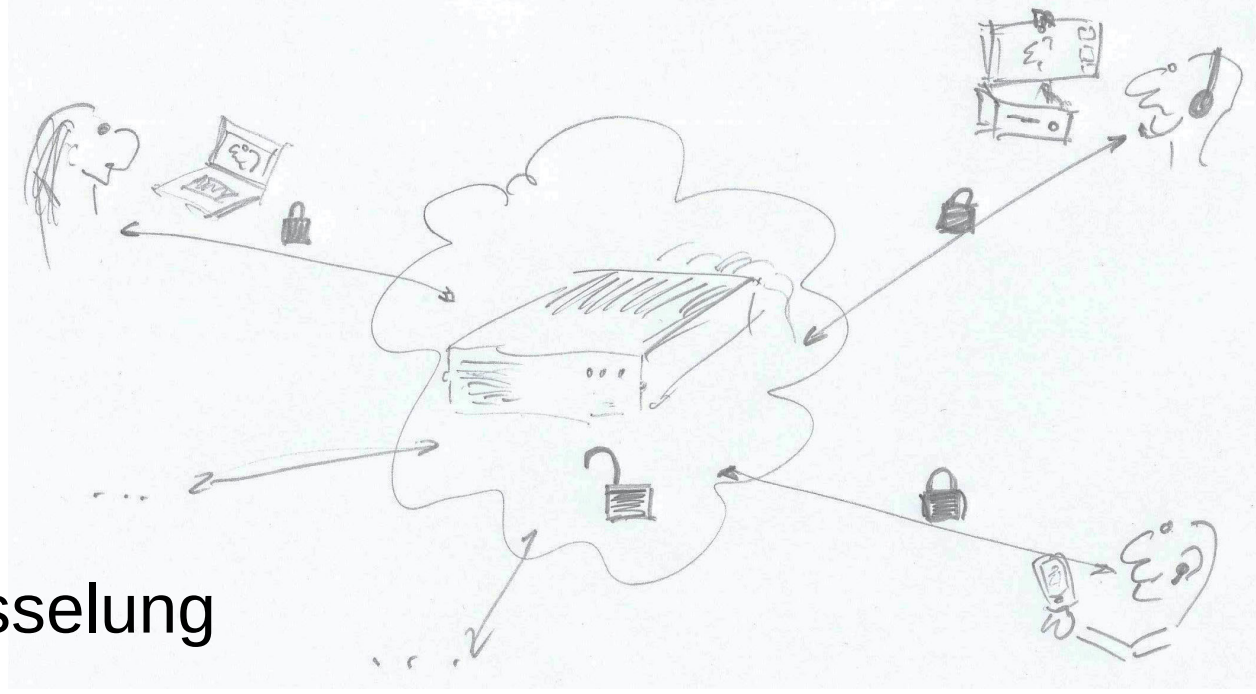


Hinweis: Die Metadaten sind nicht zwingend geheim, z.B. Wer mit wem wann kommuniziert hat, IP-Adressen, etc. !

Transportverschlüsselung vs. e2ee

Transportverschlüsselung: Ein Endpunkt ist der Server

- Server hat Zugriff auf unverschlüsselte Daten
- Je Verbindung unterschiedliche Schlüssel
- u.U. Zwischenserver, die ent- und verschlüsseln



Ende-zu-Ende-Verschlüsselung

- Endpunkte sind Sender und Empfänger
- Schlüssel sind nur diesen bekannt

Exkurs: Was ist ein Algorithmus?

- Programme „lösen“ Aufgaben
- Dazu nutzen sie definierte „programmierte“ Abläufe, die aus einzelnen Schritten bestehen
- Dabei werden Zustände (Variablen) verändert
- Die Abläufe können **linear** sein, **Verzweigungen** und **Wiederholungen** enthalten

- Definition:
  Ein Algorithmus ist eine Abfolge von Anweisungen zur Lösung eines bestimmten Problems.

Irgendwie auch mit einem Rezept vergleichbar

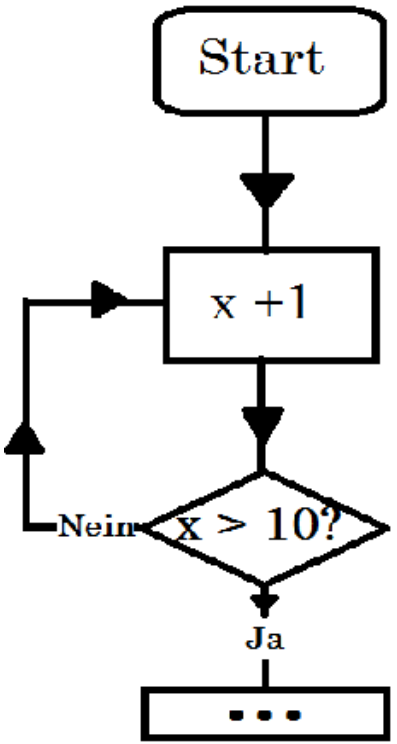


Abb. Ablaufdiagramm in der Informatik, MarcPGH CC BY-SA 2.0
https://commons.wikimedia.org/wiki/File:Ablaufdiagramm.png

Agenda

1. Warm Up

=> 2. Wie Verschlüsselung funktioniert
   (mit einigen Schlenkern)

3. Sicherheit & Angriffsmöglichkeiten

4. Gesetzgeberische Bestrebungen

5. Quellen, Fazit & Diskussion

Symmetrische Verschlüsselung

- Ein geheimer Schlüssel für Ver- und Entschlüsselung
- z.B. AES - Advanced Encryption Standard
  - sehr effizient
  - Schlüssellänge 128, 196 oder 256 Bit
  - bislang kein praktisch durchführbarer Angriff bekannt

➔ (geheime) Schlüsselübergabe nötig

➔ (übertragene) Schlüssel sollte regelmäßig erneuert werden

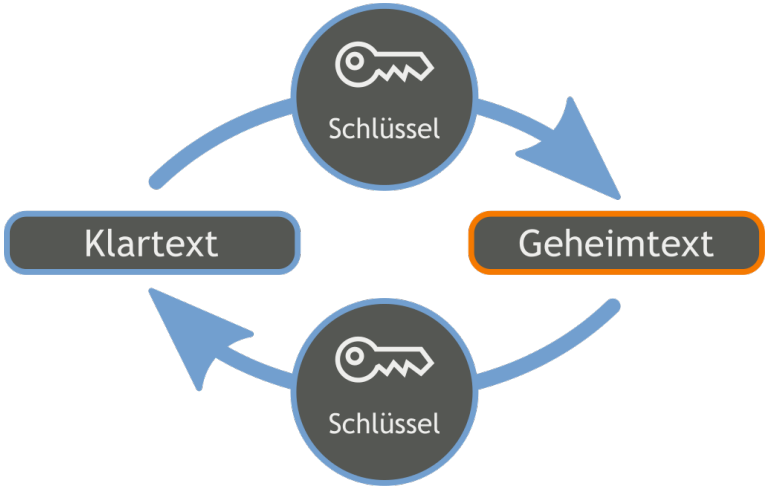


Bild: symmetrisches Kryptosystem: Verschlüsselung und Entschlüsselung mit dem gleichen Schlüssel, Bananenfalter, CC BY-SA 2.0 commons.wikimedia.org/wiki/File:Orange_blue_symmetric_cryptography_de.svg

Asymmetrische Verschlüsselung

- Zwei Schlüssel (auch Zertifikate genannt):
  - ein öffentlicher Schlüssel (public key) und
  - ein geheimer privater Schlüssel (private key)

(1) Verschlüsselung mit öffentlichem Schlüssel des Empfängers

(2) Entschlüsselung mit (eigenem) privatem Schlüssel

- Vertrauenswürdiges Verzeichnis für Public Keys nötig, z.B. Zertifizierungsstelle / Certification Authority (CA)
- Rechenaufwändig, daher hybride Verschlüsselung:
  - Daten(pakete) werden symmetrisch verschlüsselt
  - Symm. Schlüssel wird asymmetrisch verschlüsselt übertragen bzw. kodiert
- Beispiele: TLS/SSL, S-Mime, PGP, digitale Signaturen

Verschlüsselung nach Diffie-Hellmann (DH)

- Symmetrischer Schlüssel „shared secret“ wird aus asymmetrischen Schlüsselpaaren generiert (s. Abb.)
- Kryptographischer Algorithmus h(*,*)
  - $h(a_P,b_Ö) = h(a_Ö,b_P)$
  - privater Schlüssel von Bob (b_P) und öffentlicher Schlüssel von Alice ($a_Ö$)

    liefert identischen Wert wie
  - privater Schlüssel von Alice (a_P) und öffentlicher Schlüssel von Bob ($b_Ö$)

➔ Keine Übertragung des **gemeinsamen** Schlüssels nötig!

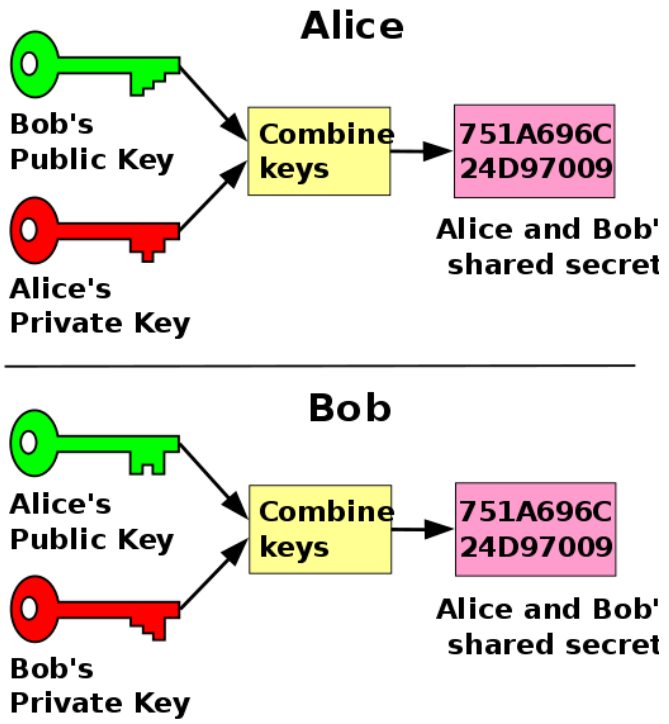
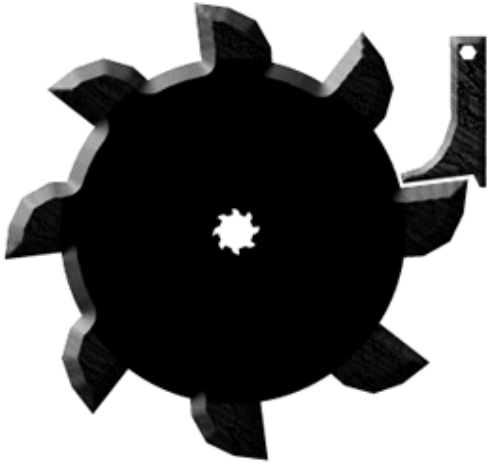


Abb: Diffie-Hellmann (DH)

Bild: David Göthberg. Public-key cryptography / asymmetric cryptography, shared secret agreement, CC BY-SA 2.0 commons.wikimedia.org/wiki/File:Public_key_shared_secret.svg

Perfect Forward Secrecy (PFS)

- Problem:
  - Angreifer* erbeutet privaten Schlüssel zum Zeitpunkt x
  - er kann damit mitgeschnittene Daten vor Zeitpunkt x entschlüsseln
- Lösung
  - Mit jeder Übertragung/Nachricht wird ein zusätzlicher, zufälliger Schlüssel gesendet, der für die nächste Übertragung gilt
  - Auch als „Double Ratchet“-Algorithmus bekannt
  - im Einsatz bei TLS (ab v1.3 verpflichtend :-)), Signal, WhatsApp, …



*) Angreifer sind per se böse und daher sollte die männliche Form genügen

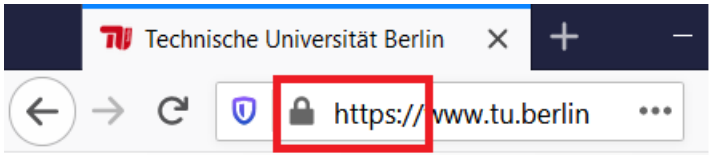
Organisatorisches Drumherum: Protokolle

Protokoll = Festgelegter Ablauf, u.a. für

- Aushandlung der Protokollversion
- Schlüsselgenerierung und -verwaltung
- Schlüsseltausch / Zertifikatverifizierung
- Ver- und Entschlüsselung
- Datenübertragung

Hohe Komplexität, viele Angriffspunkte

- Betriebssystem-abhängige Implementierungen
- Intransparenz von proprietärem Code einschließlich Treibern „Firmware“ (Microsoft, Apple, Google, Cisco, ...)
- Ständig neue Schwachstellen, z.B. Heartbleed bei OpenSSL (2014)



Sicher Surfen dank TLS / HTTPS

Beispiele:

- TLS – Transport Layer Security
- DTLS – Datagram TLS
- SRTP – Secure Real-time Transport Protocol
- HTTPS – Secure Hypertext Transport Protocol
- DoH – DNS over HTTPS

Verschlüsselung: Technisch komplex

- Zusammenspiel vieler verteilter Komponenten
- Einige Herausforderungen:
  - Programmbibliotheken / Code verifizieren (auch formal)
  - Kompromittierung von Endgeräten
  - große Teilnehmer*innenzahl (Chatgruppen, ViKos)
  - Echtzeit-Kommunikation
  - Browser-basierte Implementierung (ohne App)
  - Aber vor allem:

    Schlüsselverwaltung:
    transparent
    & nachvollziehbar

??!

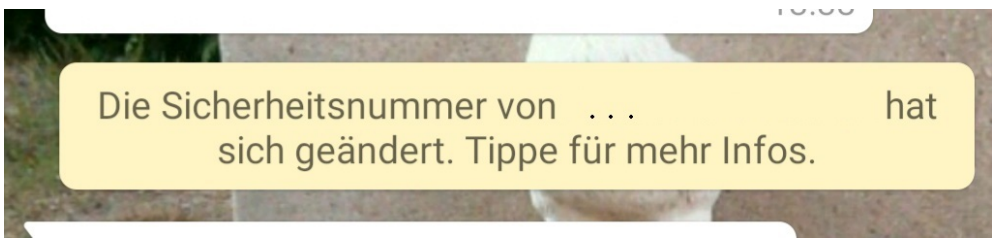


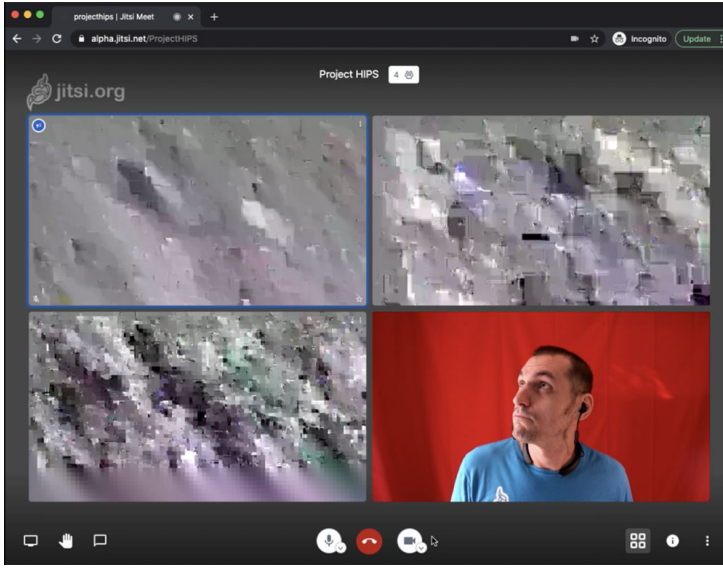
Bild: Ausschnitt aus Whatsapp-Screenshot

Exkurs: e2ee Anwendungsszenarien

- E-Mail
  - Persönliche Zertifikate (S/Mime, PGP)
  - E-Mail-Client oder Betriebssystem speichert Zertifikate (geheime Schlüssel)
  - unabhängige Zertifizierungsstellen (CA) → nachvollziehbar
- Messenger: WhatsApp, Signal & Co.
  - Open Whisper System's Signal Protocol: vollautomatisch
  - App speichert Schlüssel / je Chat und Message extra Schlüssel dank PFS
  - Anbieter verwaltet Schlüssel als „CA" → intransparent
- Kommerzielle Videokonferenztools: Zoom, Webex, Teams ...
  - e2ee - nur mit proprietärer App
  - App speichert Schlüssel
  - Anbieter-eigener (Identity-Server) ist „CA" → intransparent

Exkurs: Echtzeit Videostreaming mit WebRTC

- WebRTC: Web Real-Time Communication
  - JavaScript-basierte Peer-2-Peer-Kommunikation
  - Offener Standard (W3C) für Protokolle und Schnittstellen
  - Echtzeit-Übertragung mit Transportverschlüsselung (SRTP-DTLS)
  - In Web-Browsern, auch bei Apps
- Insertable Streams: e2ee direkt im Browser
  - Experimentelle Umsetzung bei JITSI: Video lokal verschlüsseln und „zweifach verschlüsselt“ über WebRTC übertragen
  - Schlüsseltausch noch nicht implementiert



JITSI, Rauschen ohne e2ee-Schlüssel, Quelle https://jitsi.org/blog/e2ee/

Erst einmal Luft holen.

Bild: einatmen – ausatmen, arianta.

www.flickr.com/photos/arianta/8046484551/

Agenda

1. Warm Up
2. Wie Verschlüsselung funktioniert
   (mit einigen Schlenkern)

=> 3. Sicherheit & Angriffsmöglichkeiten

4. Gesetzgeberische Bestrebungen
5. Quellen, Fazit & Diskussion

Wer hat die Schlüssel?

Oder: Wer kann sich Zugang zu den Schlüsseln oder den Inhalten verschaffen.

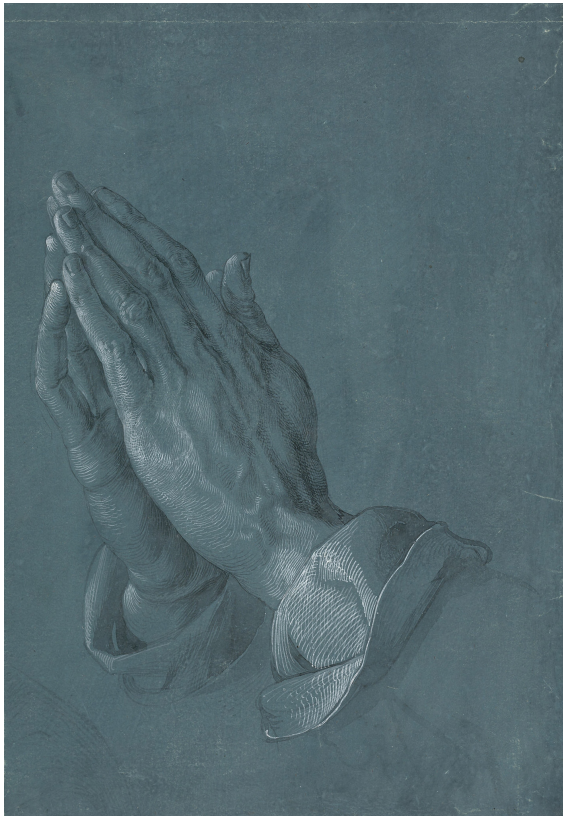
A set of wafer lock try out keys, or sometimes called "jigglers" by Willh26. CC-BY-SA-4.0, https://commons.wikimedia.org/wiki/File:Wafer_Lock_Try-Out_Keys.jpg

Frommer Wunsch:

Private Schlüssel sollten ausschließlich im Zugriff der Eigentümer / Endnutzer stehen.

- Softwarekomponenten verarbeiten privaten Schlüssel
- Zugriff auf Betriebssystemebene möglich
- Intransparenz proprietärer Software
- Sicherheitslücken / Bugs (auch bei Open Source :-( )
- Eingebaute Hintertüren (mittlerweile sogar Vordertüren !)
- Stilles Abhören privater Schlüssel

➔ **Dritte greifen private Schlüssel ab**



Sicherheit

Verschlüsselung ist nur so sicher wie das schwächste Element:

1) Plattformen/Betriebssysteme & Programme
2) Verschlüsselungsalgorithmen & Implementierung
3) Übertragungsmechanismen & Protokolle
4) Schlüsselverwaltung & Zertifizierung
5) Anwender*innen
6) ?

Wer greift an und **warum**?

- **Behörden, z.B. Strafverfolgung**
  => Überwachung, Beweissicherung
- **Kriminelle**
  => Erpressung, Identitätsbetrug, Auftragsarbeiten
- **Dienstbetreiber**
  => im staatlichen Auftrag,
  z.B. wegen Kinderpornographie, Urheberrechtsverletzung ...

- **Geheimdienste**
  => Spionage, Terrorabwehr

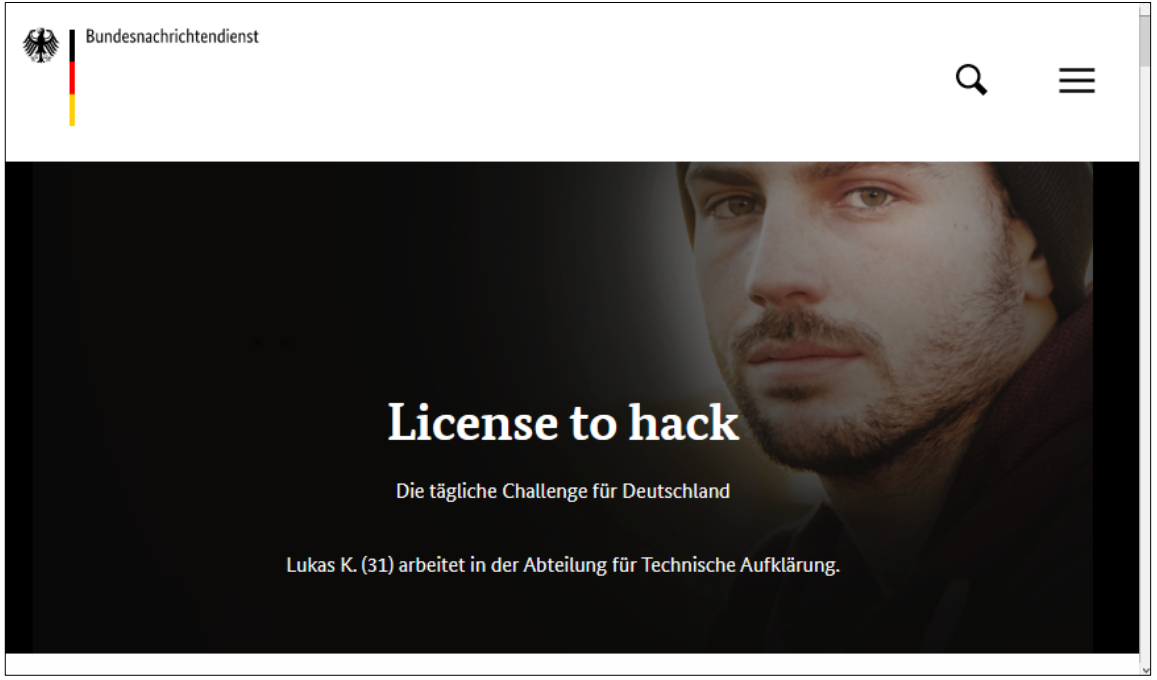


Abb. Screenshot BND Website
https://www.bnd.bund.de/DE/Karriere/Mitarbeiter-Stories/Hacker/hacker_node.html

Angriffsszenarien (1)

Methodisch & technisch unsichere Lösungen

- **Pseudo-Verschlüsselung & „Security by Obscurity**
  z.B. generierte Hashes
  => Sicherheit wird vorgegaukelt
- **Schwache “geknackte“ Verschlüsselungen**
  z.B. DES statt AES, TLS mit Heartbleed-Bug (2014)
  => Abhören zu leicht gemacht
- **Unsaubere Implementierungen**,
  z.B. manipulierte Zufallsgeneratoren, invertierbare Hash-Funktionen, lückenhaftes Error-Handling (z.B. Buffer Overflow)
  => unnötige Angriffsstellen
- **Zu einfache Passwörter**
  => Entschlüsselung möglich, z.B. mit Brute-Force oder Rainbow-Tables

➔ Lösung sollte **Stand der Technik** umsetzen & geprüft sein

Angriffsszenarien (2)

Sicherheitslöcher ermöglichen physischen Zugriff

- **(Unabsichtliche) Sicherheitslöcher**, z.B. Betriebssystem-Exploits
  - Hohe Software-Komplexität => Fehler werden hingenommen
  - Featuritis statt Fokus auf Sicherheit
  - Hersteller nicht „in der Haftung“: Bananenprinzip;
    Fixes oft spät, kurze Versionspflege (Android max. 3 Jahre)
  - Exploits (Angriffsmethoden) werden im Darknet gehandelt
    => Geheimdienste kaufen und nutzen sie
- **Geplante Sicherheitslöcher**
  - Eingebaute Hintertüren für Geheimdienste (u.a. NSA)
  - Techn. Umgehung von Verschlüsselung als Designkonzept
    (Clipper-Chip/Key Escrow, Generalschlüssel)
  - Behördlicher Zugriff auf Cloud-Dienste (E-Mail, Messenger)

➔ Geräte werden gehackt & Trojaner installiert

➔ Staat sollte Bürger schützen statt angreifen

Abb. Das laut Guinness-Buch der Rekorde umfangreichste Schweizer Messer, Slartibartfass CC-BY-SA-4.0, https://commons.wikimedia.org/wiki/File:Giant_Knife_1.jpg

Angriffsszenarien (3)

Weitere (unvollständig)

- **Cypher- + Klartext bekannt**
  => Berechnung möglich (z.B. Funksprüche im 2. WK)
- **Kompromittierte Zertifizierungsstellen (CAs)**
  => Man in the Middle-Angriff möglich, z.B. bei HTTPS
- **Abgreifen privater Zertifikate und Passwörter**
  (bei Übertragung oder externer Speicherung)
- **Passwortherausgabe erzwingen**
  (z.B. biometrisches Entsperren von Geräten)
- **Exotisch: Ausnutzung von Seiteneffekten**, z.B.
  - Stromschwankungen bei AES-Entschlüsselung in CPU, um Schlüssel leichter „raten“ zu können
  - (Röhren-)Bildschirm-Strahlung abhören

Einige Herausforderungen

1) Komplexität kryptographischer Methoden
   - ➔ zunehmend intransparent
   - ➔ schwer nachvollziehbar (kann „Trick 17“ enthalten)
2) Etliche Implementierungen sind proprietär
   - ➔ nicht (unabhängig) überprüfbar
   - ➔ selbst bei Open Source kann die Community von Big Companies ausgebootet werden
3) Staaten und IT-Unternehmen haben kein Interesse an sicherer Verschlüsselung (außer für sich selbst)
   - ➔ im Gegenteil: Private Daten werden vermarktet und für Staatsräson überwacht
4) Verschlüsselung ist technisch und organisatorisch mit Aufwand verbunden „sie ist unbequem“
5) Systeme & Endgeräte sind an vielen Stellen angreifbar

And now for something completely different

Kurze Aufmerksamkeit
für Monty Python!

Bild: "Romani ite domum" supposed graffiti on a reconstruction of a Roman settlement in Britain, at the Hull and East Riding Museum. A reference to a scene in "Life of Brian" by Monty Python.

Agenda

1. Warm Up
2. Wie Verschlüsselung funktioniert
   (mit einigen Schlenkern)
3. Sicherheit & Angriffsmöglichkeiten

=> 4. Gesetzgeberische Bestrebungen

5. Quellen, Fazit & Diskussion

Verschlüsselte Kommunikation ist wichtig!

„Digitale Inhalte bleiben nur mit Verschlüsselung vertraulich.

Digitale Selbstbestimmung bedeutet, die Hoheit über die eigenen Daten zu behalten und selbst zu entscheiden mit wem mensch sie teilt.“

Crypto Wars: Was bisher geschah

1.0 1990er - Clipper-Chip / „Key Escrow“

- Verbot starker Schlüssel, Generalschlüssel für US-Behörden
- Gesetzesinitiative scheiterte
- Kryptographie boomte, z.B. PGP

2.0 2000er – 2011 9/11: Kampf gegen den Terror

- Geheimdienste hören gesamte (Internet-)Kommunikation ab

3.0 2015er - Snowden 2013

- Aufschrei der Zivilgesellschaft
- DSGVO entstand
- sichere Kommunikation und Verschlüsselung Thema, auch für große IT-Unternehmen

4.0 2020er – Entschlüsselung erzwingen

- Abhören nur mit größerem Aufwand möglich
- weltweite Gesetzesinitiativen, vorgeschobenes Argument: Kinderpornographie

Abb. Edward Snowden Poster, GDJ, CC0 1.0
https://openclipart.org/detail/331318/edward-snowden-poster

Recht auf **Ent**schlüsselung?

AKTUELL: Konzertierte Aktion einer Allianz demokratischer **Staaten**

- Five Eyes (AUS, GB, CAN, NZ, USA), Indien & Japan:
  - Initiative **gegen** End-zu-Ende-Verschlüssung - „Kampf gegen Kinderpornographie“
- USA:
  - „Earn It“-Act liegt im Kongress, soll **Entschlüsselungsrechte** der Behörden durchsetzen
- EU (Kommission & Europäischer Rat):
  - Entschließung **gegen** sichere Verschlüsselung
  - Europol startet **Entschlüsselungsplattform** als Service
  - Gefordert: **Scannen privater** Chats bei Facebook & Co. um Kindespornographie zu verfolgen (was letztlich nur mit Entschlüsselung geht)
- Deutschland:
  - **Vorratsdatenspeicherung** wieder im Telekommunikationsgesetz
  - BND-Gesetz: **Massenüberwachung** von 30% des Volumens aller globalen Telekommunikationsnetze
  - Polizei & Geheimdienste: **Staatstrojaner** und Sicherheitslücken nutzen „**Lizenz zum Hacken**“
  - **Passwortherausgabe** im Gesetz zur Bestandsdatenauskunft verankert

Quellen: siehe Blog-Beitrag
https://blogs.tu-berlin.de/datenschutz_notizen/2021/04/30/crypto-wars-der-kampf-um-verschluesselung/

Recht auf **Ver**schlüsselung!

- Post- und Fernmeldegeheimnis
  - Garantiert Vertraulichkeit
  - Diensteanbieter fallen jetzt unter TKG
- Expertenrunde im Bundestag (1/2020)
  - Verschlüsselung ist Grundrechtsschutz
  - Demokratie braucht Kryptografie

2.6.2020:
Gemeinsamer offener Brief von Unternehmen und Akteuren der Zivilgesellschaft
https://www.ccc.de/de/updates/2021/offener-brief-alle-gegen-noch-mehr-staatstrojaner

Sicherheit und Vertrauen online schützen:
Gegen eine unbegrenzte Ausweitung von Überwachung und für den Schutz von Verschlüsselung

Agenda

1. Warm Up
2. Wie Verschlüsselung funktioniert
   (mit einigen Schlenkern)
3. Sicherheit & Angriffsmöglichkeiten
4. Gesetzgeberische Bestrebungen

=> 5. Quellen, Fazit & Diskussion

DSGVO ist stumpfes Schwert

- Große IT-Unternehmen missachten sie
  - Geschäftsmodelle setzen auf Datenausbeutung
- Schützt nicht vor staatlicher Willkür
  - Behörden und Geheimdienste können tun was sie wollen
  - Etliche Gesetze und -vorhaben hebeln Datenschutz aus
- Recht muss durchgesetzt werden!
  - Datenschutz-Behörden sind „David gegen Goliath“
  - Höchstrichterliche Entscheidungen brauchen Jahre
- Aktuelle Gesetzesinitiativen zur Entschlüsselung unterwandern sie

Fazit

Was tun.

Abb. Berlin, Lenin-Denkmal am Lenin-Platz, die Aufschrift "Keine Gewalt" ist eine Aktion der Kreuzberger Künstlerinitiative "Büro für ungewöhnliche Maßnahmen", Thurn, Joachim F. (1991), Bundesarchiv CC-BY-SA-4.0 https://commons.wikimedia.org/wiki/File:Bundesarchiv_B_145_Bild-F089663-0010,_Berlin,_Lenin-Denkmal_auf_dem_Leninplatz.jpg

Staatl. Regulierung beeinflussen!

- Lobbyieren. Verbündete finden. Auch untypische!
  - Parteien und Behörden (auch EU / Brüssel), Presse & Prominente
  - IT-Unternehmen (ja, auch die Großen können Verbündete sein)
  - Wirtschaftsunternehmen und Verwaltung
    (benötigen sichere Kommunikation)
  - Zivilgesellschaft mobilisieren
- Stichhaltige Argumentation, Überzeugungsarbeit leisten!
  - Digitalisierung erfordert in der Demokratie
    den Schutz der Privatsphäre
  - Überwachungsoptionen werden immer
    auch *anders ge~~miss~~*braucht
  - Dystopien wie Orwells' 1984 werden zunehmend
    Realität
  - Sicherheit durch zuverlässige Software & Datenschutz
- Nicht verzweifeln. Weitermachen.
  - Auch wenn es nur kleine Schritte sind.

Bild: 1. Mai 15: DGB-Demo in Berlin, Uwe Hiksch, CC BY-NC-SA 2.0
https://www.flickr.com/photos/uwehiksch/17343441115

Digitale Selbstbestimmung (für Dummies)

- Mitdenken
  - Informiert sein
  - Kontrolle über Passwörter behalten
    (z.B. generierte Passwörter verwalten)
- Überschaubare Systeme betreiben
  - Wenige (geprüfte) Programme & Apps installieren
  - Open Source Software nutzen
  - Updates regelmäßig vornehmen
- Datensparsam(er) leben
  - Unbenötigte Daten löschen
  - Cloud-Speicher vermeiden
  - Bequemlichkeit ablegen
    (z.B. auf Google-Dienste und Sprachassistenten verzichten)

Quellen & weitere Informationen (1)

Kryptographie & Web Technologien

- **Unterhaltsamer Einstieg in Encryption, Algorithmics and its Usage for Web and Email: Joshua Thijssen, Alice & bob public key cryptography 101**
  https://www.slideshare.net/jaytaph/alice-bob-public-key-cryptography-101-7647522
- Microsoft Research: EverCrypt cryptographic provider offers developers greater security assurances
  microsoft.com/en-us/research/blog/evercrypt-cryptographic-provider-offers-developers-greater-security-assurances/
- A Study of WebRTC Security, https://webrtc-security.github.io/
- BSI – Technische Richtlinie BSI TR-02102-1: Kryptographische Verfahren: Empfehlungen und Schlüssellängen
  https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf?__blob=publicationFile&v=12
- Informationen bei Wikipedia, u.a.
  - Human rights and encryption https://en.wikipedia.org/wiki/Human_rights_and_encryption
  - OpenPGP: https://de.wikipedia.org/wiki/OpenPGP
  - Diffie-Hellmann-Schlüsseltausch: https://de.wikipedia.org/wiki/Diffie-Hellman-Schl%C3%BCsselaustausch
  - TLS - Transport Layer Security inkl. bekannter Schwachstellen:
    https://en.wikipedia.org/wiki/Transport_Layer_Security
- Heise.de zu Hintertüren - https://www.heise.de/security/artikel/Best-of-Backdoor-Fails-4660194.html

Quellen & weitere Informationen (2)

Speziell zu Ende-zu-Ende-Verschlüsselung

- Umsetzung bei Jitsi: https://jitsi.org/e2ee-in-jitsi/
- Zoom-Whitepaper: https://github.com/zoom/zoom-e2e-whitepaper
- A Formal Security Analysis of the Signal Messaging Protocol, https://eprint.iacr.org/2016/1013.pdf
- WhatsApp Encryption Overview: https://www.whatsapp.com/security/WhatsApp-Security-Whitepaper.pdf

Crypto Wars

- Crypto Wars 2.0 (Erich Moechel, ORF)
  - Teil I: EU-Richtlinie für „hochklassige Cybersicherheit“ mit Nachschlüsseln https://fm4.orf.at/stories/3010484
  - Teil II: EU-Entschlüsselungspläne offenbar „beschlossene Sache“ https://fm4.orf.at/stories/3010502/
  - Erich Moechels Talk bei der Remote Chaos Experience des CCC https://media.ccc.de/v/rc3-11533-crypto_wars_2_0_de
- Wie alles anfing: Fünf Jahre Kampf gegen Ende-zu-Ende-Verschlüsselung https://netzpolitik.org/2020/wie-alles-anfing-fuenf-jahre-kampf-gegen-ende-zu-ende-verschluesselung/
- Same old story: 40 years of debating encryption (long version) https://percepticon.de/2016/10/same-old-story-40-years-of-debating-encryption-long-version/
- Key Escrow https://en.wikipedia.org/wiki/Key_escrow
- Bundestag, Anhörung am 27. Januar 2020: Fachleute für ein Recht auf Verschlüsselung https://www.bundestag.de/dokumente/textarchiv/2020/kw05-pa-inneres-669564

Und jetzt ein Bier.

Diskussion

Beiträge im TU Datenschutzblog u.a. zu Crypto Wars:
https://blogs.tu-berlin.de/datenschutz_notizen/

Bild: Die Schlüsselmaschine Enigma by William Warby, CC BY 2.0, https://de.wikipedia.org/wiki/Enigma_(Maschine)#/media/Datei:Enigma_(20967055154).jpg