FOM Hochschule für Oekonomie & Management

Hochschulzentrum Münster

Seminararbeit

im Studiengang Wirtschaftsinformatik

im Rahmen der Lehrveranstaltung
Fallstudie/ Wissenschaftliches Arbeiten

über das Thema

Kryptographie: Verwendung von mathematischen Methoden wie Verschlüsselung und digitalen Signaturen zur Sicherung von Webanwendungen

von

Joshua-Volkan Gramatzki

Betreuer:	Prof. Dr. Gregor Hülsken
Matrikelnummer	647100
Abgabedatum	9. Juni 2023

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

2FA 2-Faktor Authentifizierung
AES Advanced Encryption Standard
API Application Programming Interface
BSI Bundesamt für sicherheit in der Informationstechnik
DEA Data Encryption Algorithm
DES Data Encryption Standard
E2E Ende-zu-Ende
GGT größter gemeinsamer Teiler
HMAC Hashed Message Authentication Code
HTTP Hypertext Transfer Protocol
HTTPS Hypertext Transfer Protocol Secure
IDEA Internet Data Encryption Algorithm
ITU-T Internationale Fernmeldeunion
JS JavaScript
JSON JavaScript Object Notation
JWT JSON Web Token
MAC Message Authentication Code
MC Master Key
MITM man-in-the-midle
NIST National Institute of Standards and Technology
NSA National Security Agency
MD5 Message-Digest Algorithm 5
OAuth Open Authorization
OTP One-Time Passwort
RFC Request for Comments
RSA Rivest, Shamir, Aldeman
SHA Secure Hash Algorithm
SSL Secure Sockets Layer
SSO Single-Sign-On
TCP Transmission Control Protocol
TLS Transport Layer Security

Symbolverzeichnis

Glossar

Algorithmus Exakte Handlungsvorschrift zur Lösung eines Problems. X, 2, 4, 7, 8, 9, 10, 11, 13, 15, 17, 18, 19, 23, 25

Authentifizierung Der Prozess der Überprüfung der Identität eines Clients, Geräts oder Systems, um sicherzustellen, dass sie tatsächlich diejenige Partei sind, für die sie sich ausgeben. VI, 12, 15, 20, 21, 22, 24

Authentizität Bezeichnet die Eigenschaft der Echtheit der Daten. 13

Authorisierung Der Prozess der Zuweisung von Berechtigungen und Zugriffsrechten an eine authentifizierte Partei, um festzulegen, welche Aktionen, Ressourcen oder Informationen sie nutzen oder verwalten darf. 12, 20, 21

Base64 Ein Verfahren zur Codierung von Daten in ASCII-Zeichen, bei dem die Daten in eine darstellbare Form umgewandelt werden, um sie beispielsweise bei der Übertragung oder Speicherung in Textformaten zu verwenden. 19

Client Ein Endpunkt in einem Client-Server Modell, der Anfragen an einen Server sendet und auf dessen Antworten wartet, um Dienste, Ressourcen oder Daten zu erhalten. VIII, IX, X, 14, 18, 19, 20, 21, 24, 25

Client-Server Modell Ein Architekturmodell, bei dem eine Kommnikation zwischen einem Client, der Anfragen stellt, und einem Server, der diese Anfragen bearbeitet und Antworten liefert, etabliert ist. VIII

erweiterter euklidischer Algorithmus Eine erweiterte Version des euklidischen Algorithmus, die neben dem GGT auch die Koeffizienten berechnet, um den GGT als Linearkombination der beiden Zahlen darzustellen. 9

euklidischer Algorithmus Ein mathematischer Algorithmus zur Bestimmung des größten gemeinsamen Teiler (GGT) von zwei Zahlen durch wiederholte Anwendung des Restsatzes. VIII

Geheimtext Verschlüsseter Wortlaut eines Textes, einer Nachricht oder eines Datenblockes. 11

Handshake Ein Austausch von Nachrichten zwischen einem Client und einem Server zu Beginn einer Transport Layer Security (TLS)-Verbindung, um die Parameter der sicheren Kommunikation zu vereinbaren und die Identität des Servers zu überprüfen. 23, 24

Hashfunktion Mathematische Funktion in der Kryptographie, die Daten beliebiger Länge auf eine feste Länge (Hashwert) abbildet und dabei eine eindeutige Identifizierung der Daten ermöglicht. Sie zeichnen sich durch Kollisionsresistenz und Unumkehrbarkeit aus. IX, X, XI

Hashwert Eine feste Länge von Bits, die durch Anwendung einer kryptographischen Hashfunktion auf Daten erzeugt wird, um eine eindeutige Darstellung der Daten zu erhalten. IX, X, XI, 13

Integrität Die Vollständigkeit und Unversehrtheit von Daten muss gewährleistet werden. XI, 12, 13, 15

ISO/OSI-Referenzmodell Ein Schichtenmodell, das die Struktur und den Aufbau von Netzwerkkommunikation standardisiert und in sieben Schichten unterteilt ist, um eine effiziente und standardisierte Kommunikation zwischen verschiedenen Computersystemen zu ermöglichen . 14

Klartext Unverschlüsselter Wortlaut eines Textes, einer Nachricht oder eines Datenblockes. IX, 9, 11, 14, 17, 18

Klartextraum Menge aller verschlüsselbaren Klartexte. 9

Kollisionsresistenz Die Eigenschaft in einer kryptograpsischen Hashfunktion, in der es schwierig ist, zwei unterschiedliche Eingabewerte zu finden, die denselben Hashwert erzeugen. IX

Kompressionsfunktion In der Kryptographie eine Funktion, die Daten auf eine kompakte Form reduziert, wodurch Speicherplatz eingespart oder die Datenübertragungseffizienz verbessert wird. 10

Kryptoanalyse Entschlüsselung von verschlüsselten Informationen und bewertung der Sicherheit von Verschlüsselungsverfahren. 3

Kryptographie Verschlüsselung von Informationen zum Schutz vor unbefugtem Zugriff. X, 2, 3, 12, 15, 16, 23, 25

Kryptologie Oberbegriff für Kryptographie und Kryptoanalyse zur Sicherheit von Informationen und Entwicklung von Techniken zur Verschlüsselung und Entschlüsselung von Daten. 3

Malware Eine bösartige Software, die entwickelt wurde, um unerwünschte oder schädliche aktionen auf einem Computer oder einem Netzwerk auszuführen, wie z.B. Datenverlust, Systembeschädigung oder unbefugten Zugriff ermöglichen. 18

Message-Digest Algorithm 5 (MD5) Kryptographische Hashfunktion zur berechnung eines 128-Bit-Hashwert aus einer beliebigen Nachricht. 10

monoalphabetisch Verschlüsselungsart, bei der der Schlüssel für jedes Zeichen gleich bleibt. 5

Nicht-Abstreitbarkeit Eine durchgeführte Handlung ist eindeutig zurechenbar. 13

private Key Ein geheimer kryptographischer Schlüssel, der zur Entschlüsselung und Signierung von Daten dient und streng vertraulich bleiben muss. X, 8, 13, 14, 19

Public-Key Verschlüsselungsverfahren Eine Form der asymmetrischen Verschlüsselung, bei der ein öffentlicher Schlüssel zum Verschlüsseln von Daten verwendet wird, während der dazugehörige [geheimer Schlüssel] benötigt wird, um die verschlüsselten Daten zu entschlüsseln. 8, 13, 14

public Key Ein öffentlich zugänglicher kryptographischer Schlüssel, der für Verschlüsselung und Verifizierung verwendet wird. X, XI, 8, 14

Rainbow-Table Eine vorausberechnete Tabelle, die zur schnellen Rückberechnung von Hashwerten verwendet wird, um Passwörter oder andere ursprüngligche Daten zu entschlüsseln, indem sie vorab Hashwerte mit den entsprechenden Eingabewerten speichert. 17

Server Ein Computer oder eine Software, die Anfragen von Clients empfängt, verarbeitet und darauf antwortet, indem sie Dienste, Ressourcen oder Daten bereitstellt. VIII, IX, 14, 20, 21, 24, 25

SHA256 Ein kryptographischer Hashalgorithmus, der eine 256-Bit-Hashfunktion verwendet, um eine eindeutige Darstellung von Daten zu erzeugen und häufig für die Integritätsprüfung und digitale Signatur verwendet wird. 19, 25

SQL-Injection Eine Angriffstechnik, bei der schädlicher SQL-Code in eine Andwendungsdatenbank eingeschleust wird, um unautorisierten Zugriff auf die Datenbank zu erlangen oder unerwünschte Aktionen durchzuführen. 18

Steganographie Verbergen von Informationen in einer scheinbar harmlosen Nachricht. 3

Unumkehrbarkeit Die Eigenschaft einer kryptographischen Hashfunktion, bei der es praktisch unmöglich ist, von einem gegebenn Hashwert auf den ursprünglichen Eingabewert zurückzuschließen. IX

Verfügbarkeit Daten müssen zu definierten Zeiten im Einklang mit der Vertraulichkeit und Integrität zur Verfügung stehen. 12, 13

Vertraulichkeit Daten dürfen nur von Personen verarbeitet werden, die dafür bestimmt sind. XI, 12, 13, 15

X.509 Sandard der Internationale Fernmeldeunion (ITU-T) für eine public Key-Infrastruktur zum erstellen digitaler Zertifikate (Unterunterabschnitt 4.2.1). Im November 2020 als ISO/IEC 9594-8 aktualisiert worden. 14

1 ToDos

Dieses Kapitel muss vor Abgabe der Arbeit *unbedingt* entfernt werden

Seitenangaben zu den Quellen hinzufügen

Literaturrecherche: Angeben, wonach ich im Rahmen der Literaturrecherche gesucht habe/ weshalb ich danach gesucht habe/ Was ich mir überlegt habe

Stichwortverzeichnis erstellen

- Eine gute Forschungsanalyse beinhaltet, dass neue Fragen entstehen
  - „Was bedeutet x für y?“
  - Zusammenfassung dieser im Fazit
- Literatur
  - Peer Reviewd Zeitschriften
  - Monographien
  - Lexika vermeiden

2 Einleitung

In der heutigen digitalen Welt ist die Sicherheit von Daten und Informationen von größter Bedeutung. Das Internet und Webanwendungen haben unser Leben erleichtert und verbessert, aber auch neue Herausforderungen in Bezug auf Datensicherheit und Datenschutz mit sich gebracht. Durch den Einsatz von Verschlüsselungs- und Signaturverfahren können Daten und Informationen sicherer übertragen und gespeichert werden. Das ist das Ziel der Kryptographie.

Kryptographie ist die Wissenschaft der Ver- und Entschlüsselung von Informationen. Sie umfasst mathematische Verfahren zur Sicherung von Daten und Informationen, die über das Internet oder andere Netzwerke übertragen werden. Die Verschlüsselung stellt sicher, dass die Informationen nur von autorisierten Personen gelesen werden können, während digitale Signaturen gewährleisten, dass die Informationen nicht manipuliert wurden.

2.1 Zielsetzung

In dieser Arbeit wird untersucht, wie mathematische Methoden, wie die Verschlüsselung und die digitale Signatur, eingesetzt werden, um Web-Anwendungen zu sichern. Webanwendungen werden immer häufiger zur Speicherung und Verarbeitung von Daten und Informationen verwendet und stellen daher ein attraktives Ziel für Angreifer dar. Durch den Einsatz von Kryptographie kann die Sicherheit von Webanwendungen erhöht werden.

Dabei wird versucht, die Forschungsfrage „Wie sehr tragen die aktuell genutzten kryptographischen Methoden zur Sicherheit von Daten in Webanwendungen bei?“ anhand einer Literaturanalyse zu beantworten.

2.2 Aufbau der Arbeit

Dafür werden zunächst ein paar Grundbegriffe und grundlegende Konzepte der Kryptographie erklärt, sowie verschiedene kryptographische Verfahren und dargestellt. Abschnitt 4 befasst sich mit verschiedenen Möglichkeiten, Verschlüsselungsverfahren in Webanwendungen einzubinden und die Sicherheit von Daten zu gewährleisten, während Abschnitt 5 Probleme und Schwierigkeiten, die während und nach der Implementierung dieser Verfahren auftreten können erläutert.

3 Grundlagen der Kryptographie

3.1 Grundbegriffe

Die folgende grundlegende Erklärung der Grundbegriffe der Kryptographie ist geschrieben in Anlehnung an Brinkmann, Kapitel 1.1.

Zu der Kryptographie, der Kunst Nachrichten oder Daten durch Verschlüsselung geheim zu halten, gibt es noch andere Wissenschaften, die sich mit Verschlüsselungen befassen. Bei der Kryptographie werden Nachrichten so unlesbar gemacht, dass sie nur von Empfängern mit einem geheimen Schlüssel wieder in den ursprünglichen Text umgewandelt werden können.

Parallel zu der Kryptographie gibt es die Steganographie, bei der die bloße Existenz der zu versteckenden Informationen verborgen wird. Dies wird dadurch ermöglicht, dass die geheimen Informationen in anderen, harmlos aussehenden Nachrichten versteckt werden, sodass sie nur von Eingeweihten entdeckt werden können.

Die Kryptoanalyse befasst sich, entgegen der Kryptographie, nicht mit der Verschlüsselung, sondern damit, verschlüsselte Nachrichten und kryptographische Verfahren zu analysieren. Dies wird mit dem Ziel gemacht, Verschlüsselungen zu brechen.

Die Wissenschaft, die Kryptographie und Kryptoanalyse vereint, ist die Kryptologie. Diese wird das Kernthema dieser Arbeit sein.[1]

3.2 Historische Entwicklung und grundlegende Konzepte

Besonders in Zeiten des Krieges war es vonnöten, Informationen sicher von den Befehlshabern zu den Offizieren überbringen zu können. So wurden die ersten kryptographischen Konzepte für den militärischen und politischen Nutzen entwickelt.[2]

Eine grundlegende Erfindung für das Voranschreiten der Kryptographie war die Erfindung des Schlüssels, wobei die Erfindung des variablen Schlüssels die Geburtsstunde der Kryptographie markiert..[3] Durch variable Schlüssel ist es nun möglich, einfach zwischen verschiedenen Verschlüsselungen zu wechseln.

---

1 *Brinkmann*, *E.*, 2001, S. 5.
2 *Beutelspacher*, *A.*, 2017.
3 Ebd.

3.2.1 Caesar-Scheibe

Bei der Caesar-Scheibe (auch Caesar-Chiffre genannt) werden die Buchstaben des Alphabetes um einen bestimmten Wert, dem Wert des Schlüssels, rotiert. Dabei wird das 'z' als Buchstabe vor dem 'a' gesehen.

Dieser Schlüssel ändert sich nicht für kommende Zeichen, sondern ist für die gesamte Nachricht identisch.

Abbildung 1 verbildlicht die Caesar-Chiffre anhand der klassischen Darstellung zweier konzentrischen Kreise mit dem Schlüssel 13.

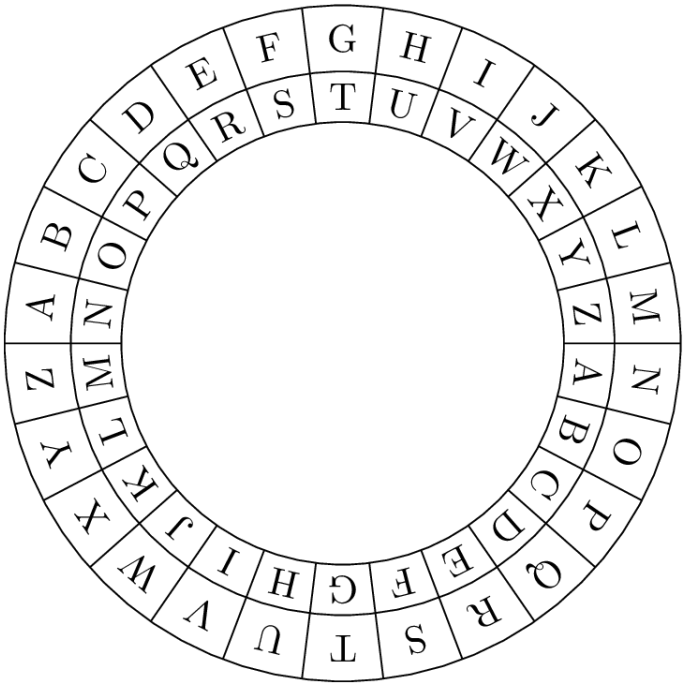


Abbildung 1: Caesar-Chiffre am Beispiel eines Scheibe (Eigene Darstellung)

Caesar hat für seine Verschlüsselungen immer den Schlüssel 3 genutzt,[4] sodass aus CAESAR folglich FDHVDU wird. Dies bietet allerdings praktisch keine Sicherheit, besonders nicht nach heutigen Standards.

Da es sich durch die Jahrtausende hinweg sich immer wieder gezeigt hat, dass es fast unmöglich ist, Algorithmen geheim zu halten, kam mit der Zeit die Sorge auf, dass „[wer] das Verfahren kennt, insbesondere wer es erfunden hat, [...] es auch brechen [kann].“.[5] Der Kryptologe Auguste Kerckhoffs formulierte 1833 in seinem zweiten der sechs Grundsätze

[4] *Beutelspacher, A.*, 2017.
[5] Ebd.

zur Konstruktion eines sicheren Verschlüsselungsferfahrens, dass das System keine Geheimhaltung erfordern darf und ohne Schaden in die Hände von Feinden fallen können.[6]

3.2.2 Polyalphabetische Codes

Um 1500 wurde klar, dass monoalphabetische Codes keine wirkliche Sicherheit bieten. Im 16. Jahrhundert geschah ein Umdenken in der Verschlüsselung. Mehrere gelehrten hatten eine ähnliche Idee für eine neue Dimension dieser.[7] Eines der bekanntesten polyaplhabetischen Verfahren ist die Vigenère-Chiffre, benannt nach ihrem Entwickler, **Blaise de Vigènere**.

Bei der Vigènere-Chiffre wird sich, entgegen zu der Caesar-Chiffre, dargestellt in Unterunterabschnitt 3.2.1, ein Schlüsselwort überlegt. Sollte dieses Wort kürzer sein, als der eigentliche Text, wird es so lange wiederholt, bis jedem Zeichen im Klartext ein Zeichen des Schlüssels zugeordnet ist. Anschließend wird jedes Zeichen im Klartext um das dazu assoziierte Zeichen des Schlüssels verschoben, sodass für jedes Zeichen ein anderes Zeichen im verschlüsselten Text herauskommt.

Tabelle 1[8] zeigt das Vigènere-Verfahren mit dem Schlüssel „ALICE“ auf dem Klartext „DIES IST EIN VERSCHLÜSSELTER TEXT“

Tabelle 1: Verschlüsselung mit Vigènere-Verfahren[9]

Klartext	D	I	E	S		I	S	T		E	I	N		V	E	R	S	C	H	L	U	E	S	S	E	L	T	E	R		T	E	X	T
Schlüssel (Alice)	A	L	I	C		E	A	L		I	C	E		A	L	I	C	E	A	L	I	C	E	A	L	I	C	E	A		L	I	C	E
Verschlüsselter Text	D	T	M	U		M	S	E		M	K	R		V	P	Z	U	G	H	W	C	G	W	S	P	T	V	I	R		E	M	Z	W

Durch diese nicht gleichmäßige Verschiebung bietet die Vigènere-Chiffre gegenüber der Caesar-Chiffre eine deutlich sicherere Verschlüsselung, ist aber allemal nicht vollständig sicher. Sobald der Schlüssel bekannt ist, muss nur jedes Zeichen des verschlüsselten Textes in entgegengesetzter Richtung des Schlüssels verschoben werden.

Dennoch sind polyalphabetische Verschlüsselungsverfahren in Anbetracht der einfachen Durchführung stark und bieten ein großes Potenzial.

---

6 *Petitcolas*, *F.*, o. J., Übersetzt aus dem Französischen.
7 *Beutelspacher*, *A.*, 2017.
8 *Thielert*, *S.*, 2007, S. 6.
9 ebd., S. 6

3.2.3 Enigma

Der Beginn des 20. Jahrhunderts läutete die Blütezeit der kryptografischen Maschinen ein. Diese wurden mit wachsender Komplexität der Verschlüsselungen immer notwendiger. Die wohl berühmteste Chiffriermaschine der Welt, die Enigma, wurde 1918 von ihrem Erfinder **Arthur Schebius** zum Patent angemeldet.[10]

Der folgende Abschnitt ist nach der Patentschrift № DE416219C1I[11] geschrieben.

Den anderen Kryptografiemaschinen ihrer Zeit war die Enigma um ein Vielfaches überlegen.
Die Verschlüsselung mit der Enigma durchgeht verschiedene Ebenen. Zunächst verläuft das elektrische Signal der Tasten durch ein Steckbrett, mit welchem man durch das Stecken eines Kabels zwischen zwei Buchstaben diese tauschen kann. Anschließend durchläuft das Signal drei auswechselbare, drehende Walzen, die jeweils 26 Stellungen haben. Zusätzlich zu den drei drehbaren Walzen befindet sich rechts an der Maschine eine unbewegliche „Eintrittswalze“ und links daneben eine „Umkehrwalze“, als Reflektor. Sowohl die Eintrittswalze, als auch die Umkehrwalze war bei den meisten Enigmas nicht drehbar, aber wechselbar.
Alle dieser insgesamt 5 Walzen sind über 26 elektrische Kontakte miteinander verbunden. Jeder dieser Kontakte leitet das elektrische Signal, das von der Tastatur kommt, auf einem anderen Weg durch das Walzensystem, bis das Signal an der Umkehrwalze angelangt.
Dort wird es, wie der Name sagt, umgekehrt und durchläuft die 3 Walzen erneut, in umgekehrter Reihenfolge und verlässt den Walzensatz über die Eintrittswalze. Jeder Tastendruck auf der Tastatur dreht mindestens eine Walze um eine Stellung weiter - die erste Walze wird mit jedem Tastendruck gedreht, die zweite, wenn die erste eine ganze Umrundung durchgeführt hat und so weiter - bevor der Stromkreis geschlossen wird.[12]

Diese besondere Konstruktion der Walzen und des Steckbrettes gab der Enigma

$$60 * 676 * 16900 * 150738274937250 = 103325660891587134000000 \quad (1)$$

[10] *Shebius*, *A.*, 1918.
[11] Ebd.
[12] Ebd.

also ungefähr 10^{28} Kombinationen, bestehend aus den 60 Walzenkombinationen[13], 676 Ringstellungen[14], die 16900 Walzenstellunge[15][16] und den insgesamt 150738274937250 Steckermöglichkeiten[17].[18]

3.3 Kryptographische Verfahren und Algorithmen — symmetrische und asymmetrische Verschlüsselung, Hash-Funktionen etc.

3.3.1 Symmetrische Verschlüsselungsalgorithmen

Bei symmetrischen Verschlüsselungsalgorithmen wird zum verschlüsseln und zum Entschlüsseln derselbe Schlüssel verwendet. Abbildung 2[19] zeigt, wie der selbe Schlussel zum Ver- und Entschlüsseln genutzt wird. Dies ermöglicht eine einfache und schnelle Kommunikation. Jedoch führt dies dazu, dass die Geheimhaltung und die sichere Verteilung des Schlüssels schnell zu einer Sicherheitslücke führen kann.

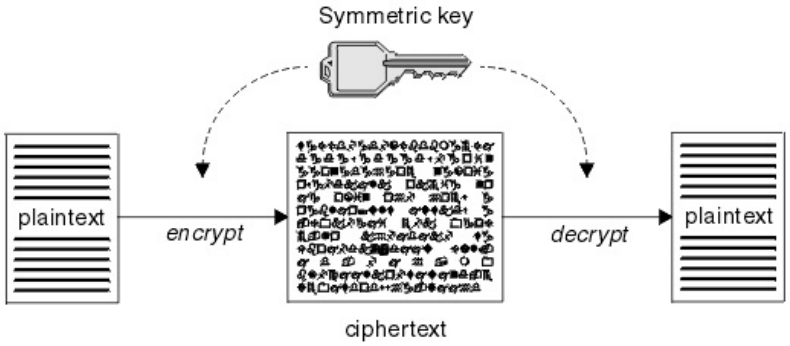


Abbildung 2: Schaubild für eine symmetrische Verschlüsselung[20]

Symmetrische Verschlüsselungsverfahren werden deshalb als sicher anerkannt, wenn man ohne den Schlüssel den Klartext nicht aus dem verschlüsselten Text ermitteln kann.[21]

---

[13] Aus einem Pool von 5 möglichen Walzen wurden immer 3 genutzt $\Rightarrow 5 * 4 * 3 = 60$

[14] Jeweils 26 Ringstellungen der rechten und mittleren Walze, die linke Walze $\Rightarrow 26^2 = 676$

[15] Während es rechnerisch bei 3 Walzen $26^3 = 17576$ Walzenstellung gäbe, kann die Enigma durch eine Bauanomalie nur $26^2 * 25 = 16900$ Stellungen erreichen

[16] *Hamer*, *D. H.*, 1997.

[17] Wenn man die einzelnen möglichkeiten auflistet, wie viele Steckermöglichkeiten es bei n Steckern gibt, zeigt sich, dass bei 10 steckern ein Maximum erreicht ist

[18] *Guo*, *J.*, *Hu*, *Y.*, *Wang*, *Y.*, 2022, S. 810.

[19] *Krawisz*, *D.*, 2013.

[20] ebd.

[21] *Thielert*, *S.*, 2007, S. 5.

3.3.1.1 Data Encryption Standard (DES)

DES, oft auch Data Encryption Algorithm (DEA), wurde 1975 im ”Federal Register” der USA veröffentlicht und als Kooperation zwischen dem National Institute of Standards and Technology (NIST), der National Security Agency (NSA) und IBM entwickelt.[22]

DES ist ein Blockchiffre-Verfahren, was bedeutet, dass die Daten in immer gleich großen Blöcken (bei DES in Blöcken von 64 Zeichen) und mit einer immer gleichen Schlüssellänge (56 bit bei DES) verschlüsselt werden.[23]

3.3.1.2 Internet Data Encryption Algorithm (IDEA)

IDEA wurde 1990 als eine erweiterte Version zu DES entwickelt und verwendet eine Schlüssellänge von 128 bit.[24]

3.3.2 Asymmetrische Algorithmen und RSAVerfahren

Asymmetrische Verschlüsselungsverfahren verwenden, im Gegensatz zu den symmetrischen Verschlüsselungsverfahren (Unterunterabschnitt 3.3.1), zwei Schlüssel. Einen public Key, welcher zum Verschlüsseln der Daten verwendet wird und einen private Key, welcher zum Entschlüsseln der Daten verwendet wird. Wie der Name sagt, muss nur der private Key geheim gehalten werden.

Das Rivest, Shamir, Aldeman (RSA) ist das erste entwickelte Public-Key Verschlüsselungsverfahren und besitzt auch heute noch eine große Relevanz.[25]

3.3.2.1 Schlüsselerzeugung

Für die Erzeugung eines Schlüsselpaares für den RSA-Algorithmus werden einige Zahlen benötigt. Die Erste ist als Sicherheitsparameter eine Zahl $k \in \mathbb{N}$, welche die Größe des Produktes der beiden für die Verschlüsselung gewählten Primzahlen angibt.. Zudem werden zwei, voneinander statistisch unabhängig, zufällige Primzahlen p und q gewählt,

[22] *Johnson*, *T. R.*, 1998, S. 232.
[23] *Thielert*, *S.*, 2007, S. 6.
[24] Ebd., S. 6.
[25] *Buchmann*, *J.*, 2016, S. 168.

sodass der RSA Modul n, der Wert, der später zum Ver- und Entschlüsseln genutzt wird, gebildet werden kann aus $n = p * q$. Zusätzlich wird eine natürliche Zahl e gewählt, für die

$$1 < e < \varphi(n) = (p-1)(q-1) \text{ und } \gcd(e,(p-1)(q-1)) = 1 \tag{2}$$

gilt und daraus mit den folgenden Bedingungn

$$1 < d < (p-1)(q-1) \text{ und } d * e \equiv 1 \mod (p-1)(q-1) \tag{3}$$

eine weitere Zahl $d \in \mathbb{N}$ gebildet.

Da $\gcd(e,(p-1)(q-1)) = 1$ ist, existiert eine solche Zahl d definitiv. Berechnet werden kann sie mit dem erweiterten euklidischen Algorithmus. Dabei ist zu beachten, dass e stets ungerade ist. Der öffentliche Schlüssel bildet sich dabei aus dem Paar (e,n) und der private Schlüssel aus der Zahl d.[26]

Damit RSA eine sichere Verschlüsselung garantieren kann, müssen die beiden Primzahlen p und q passend gewählt werden. Dafür ist es üblich, dass k als gerade Zahl gewählt wird, die mittlerweile mindestens 1024-Bit lang ist.[27]

3.3.2.2 Verschlüsselung

Um mit dem RSA Algorithmus eine Nachricht zu verschlüsseln, wird der öffentliche Schlüssel (e,n) benötigt. Aus einem Klartext $m \in \mathbb{Z}_m$ mit $\mathbb{Z}_m$ als RSA Klartextraum erhält man den verschlüsselten Text c mit

$$c = m^e \mod n \tag{4}$$

m kann wieder rekonstruiert werden mit

$$m = c^d \mod n \tag{5}$$

wobei c der zuvor erhaltene verschlüsselte Text ist, d ist der private Schlüssel und n ist ein Teil des öffentlichen Schlüssels.[28]

[26] *Buchmann, J.*, 2016, S. 169.
[27] Ebd., S. 169.
[28] *Davis, T.*, 2003, S. 6.

3.3.3 Hashfunktionen — Secure Hash Algorithm (SHA)

Im Grundlegenden sind Hashfunktionen Algorithmen, die einen Text beliebiger Länge zu einem neuen Text mit vorgegebener Länge komprimieren.[29] Generiert werden sie mit Hilfe von sogenannten Kompressionsfunktionen.

Damit Hash- und Kompressionsfunktionen in der Kryptographie zur Authentifizierung, wie z.B. zur Speicherung von Passwörtern genutzt werden können, müssen noch ein paar Kritierien erfüllt werden. Diese werden folgend erklärt:

Definition 3.1. *Eine Einweghashfunktion ist eine Funktion h, die die folgenden Bedingungen erfüllt:*[30]

1. *Die Beschreibung von h muss öffentlich bekannt sein und sollte keine geheimen Informationen erfordern (Erweiterung des Kerckhoff'schen Prinzips[31]).*
2. *Das Argument X kann von beliebiger Länge sein und das Ergebnis $h(X)$ hat eine feste Länge von n-Bits (mit $n \geq 64$).*
3. *Gegeben h und X, muss die Berechnung von $h(X)$ einfach[32] sein.*
4. *Die Hashfunktion muss in dem Sinne monodirektional sein, dass es bei einem Y im Abbild von h schwer[32] ist, eine Nachricht X zu finden, so dass $h(X) = Y$ ist, und dass es bei X und $h(X)$ schwer[32] ist, eine Nachricht $X' \neq X$ zu finden, so dass $h(X') = h(X)$.*

Da es nicht bekannt ist, ob es Einwegfunktionen gibt, die optimal arbeiten, werden in der Definition die Begriffe einfach und schwer verwendet, da es heutzutage noch keine Algorithmen bekannt sind, die eine Einweghashfunktion schnell genug umkehren kann[33]

Die Secure Hash Algorithms (SHAs) sind verschiedene kryptologische Hashfunktionen und eine modifizierte Version des Message-Digest Algorithm 5 (MD5) (MD5), welche zur Berechnung eines Prüfwertes für beliebige Nachrichten dienen und unter anderem die Grundlage zur Erstellung einer digitalen Signatur, genauer erläutert in Unterunterabschnitt 4.2.1, sind.[34]

---

[29] *Preneel*, *B.*, 2003, S. 15.
[30] Ebd., S. 17.
[31] *Petitcolas*, *F.*, o. J.
[32] „einfach“ und „schwer“ sind hier im Kryptographischen Sinne zu verstehen und beziehen sich auf das Zusammenspiel von Laufzeit und Rechenaufwand eines Algorithmus
[33] *Buchmann*, *J.*, 2016, S. 234.
[34] *Encryption Consulting LLC*, 2023.

2012 wurde SHA-3 (auch bezeichnet als Keccak) von NIST standardisiert und wird heute als sicher angesehen,[35] aber auch die Algorithmen unter SHA-2 sind heute stark verbreitet. Unter SHA-2 und SHA-3 werden dabei nicht einzelne Algorithmen sondern Algorithmusgruppen verstanden, deren zugrunde liegenden Algorithmen sich primär in der Länge des Prüfwertes, den sie ausgeben unterscheiden. Häufig genutzt werden heute die Algorithmen SHA-256 und SHA-512.

Da bei den SHA-Algorithmen selbst kleine Änderungen im Klartext schon für einen stark geänderten Geheimtext.

[35] *Buchmann*, *J.*, 2016, S. 239.

4 Anwendung von Kryptographie in Webanwendungen

In der heutigen vernetzten Welt sind Webanwendungen allgegenwärtig und spielen eine wichtige Rolle, um Dienste bereitzustellen und sensible Informationen auszutauschen. In diesem Kapitel geht es um den Einsatz von Kryptographie in Webanwendungen. Damit sollen die Sicherheitsanforderungen an Vertraulichkeit, Integrität und Verfügbarkeit erfüllt werden.

Zunächst werden die grundlegenden Sicherheitsanforderungen an Web-Anwendungen erläutert. Anschließend wird gezeigt, wie Kryptographie eingesetzt werden kann, um sensible Daten vor unberechtigtem Zugriff zu schützen.

Die Verschlüsselung der Datenübertragung ist ein zentraler Aspekt der Sicherheit von Web-Anwendungen. Um Daten während der Übertragung zu schützen, konzentriert sich der nächste Abschnitt auf das Hypertext Transfer Protocol Secure (HTTPS)-Protokoll und die Secure Sockets Layer (SSL)/ Transport Layer Security (TLS)-Verschlüsselung.

Die Sicherheit von Passwörtern wird ebenfalls behandelt. Methoden wie Salted Hashing und Key Derivation Functions werden vorgestellt, um Passwörter sicher zu speichern.

In den Abschnitten über Authentifizierung und Authorisierung werden verschiedene Verfahren und Protokolle, wie z. B. Token-basierte Verfahren und OAuth, vorgestellt, um die Identitätsprüfung und Zugriffskontrolle in Web-Anwendungen zu gewährleisten.

Ziel dieses Kapitels ist ein umfassender Einblick in den Einsatz von Kryptographie in Webanwendungen zur Gewährleistung von Vertraulichkeit, Integrität und Verfügbarkeit.

4.1 Sicherheitsanforderungen an Webanwendungen — Vertraulichkeit, Integrität, Verfügbarkeit

Das Bundesamt für sicherheit in der Informationstechnik (BSI) listet für die Webanwendung drei primäre Schutzziele

- Vertraulichkeit,
- Integrität sowie
- Verfügbarkeit

von denen zudem abgeleitet u.a. die beiden sekundäre Schutzziele

- Authentizität und
- Nicht-Abstreitbarkeit

aufgeführt sind.[36]

Das Zusammenarbeiten der Schutzziele, insbesondere der drei primären Schutzziele, zur garantie einer sicheren Webanwendung wird im Folgenden untersucht.

4.2 Verschlüsselung von Datenübertragungen — SSL/TLS-Verschlüsselung, HTTPS-Protokoll

Es gibt verschiedene Methoden, eine Datenübertragung im Internet zu verschlüsseln. Die wahrscheinlich bekannteste ist das HTTPS-Protokoll, welches im Unterunterabschnitt 4.2.2 untersucht wird. Unterunterabschnitt 4.2.1

4.2.1 Digitale Zertifikate — SSL/TLS-Verschlüsselung

befasst sich mit der Generierung von digitalen Zertifikaten zur Verifizierung der Echtheit der Daten.

Eine digitale Signatur ist ein Public-Key Verschlüsselungsverfahren, bei dem einer Nachricht ein einzigartiger Schlüssel angehangen wird, der meist daraus generiert wird, dass der Hashwert (Unterunterabschnitt 3.3.3) einer Nachricht mit dem private Key des Empfängers verschlüsselt wird.[37] Als NIST-Standard wird dabei ein SHA-Algorithmus verwendet.

[36] *Bundesamt für Sicherheit in der Informationstechnik*, 2022, S. 8.
[37] *Kaur*, *R.*, *Kaur*, *A.*, 2012, S. 297.

Der Klartext, die Signatur und der public Key des Senders werden dann zusammengepackt und mit dem public Key des Empfängers zusammen verschlüsselt. Diese signierte und verschlüsselte Nachricht wird dann übermittelt.[38]

Um die Nachricht zu entschlüsseln, wird wie bei einer normalen Public-Key Verschlüsselungsverfahren der private Key des Empfängers entsprechend des genutzten Verschlüsselungsvefahren auf die Nachricht angewandt. Anschließend wird der Klartext der Nachricht gehashed und die Signatur mit dem public Key des Senders entschlüsselt. Diese beiden Werte werden verglichen, ob sie identisch sind. Sind sie es, so ist die Nachricht verifiziert, andernfalls nicht.[39]

Ein wichtiges Werkzeug der sicheren Webentwicklung sind SSL und TLS Zertifikate. TLS ist dabei der Nachfolger für SSL, beide nutzen aber das gleiche selbe X.509 Zertifikat[40]. Mit SSL kann ein Client überprüfen, ob das Zertifikat des angefragten Servers von einer vertrauten Autorität ausgestellt wurde oder nicht, indem die digitalen X.509 Zertifikate verifiziert werden. Andernfalls wird die Verbindung abgelehnt.[41]

Im ISO/OSI-Referenzmodell liegt SSL auf Ebene 4, der Transportebene[42] und ist eine „externe Überprüfung durch einen vertrauenswürdigen Dritten, der garantiert, dass ein Webserver der ist, der er vorgibt zu sein“.[43]

Das TLS-Protokoll lässt sich, wie in Tabelle 2 dargestellt, in zwei Schichten darstellen.

Tabelle 2: TLS-Protokoll Schichten

TLS Handshake Protocol	TLS Change Cipher Spec. Protocol	TLS Alert Protocol	TLS Application Data Protocol
TLS Record Protocol			

4.2.1.1 TLS-Record Protokoll

Das TLS-Record Protocol fragmentiert die Daten in Blöcke, komprimiert sie wenn gewünscht, wendet Message Authentication Code (MAC) darauf an, verschlüsselt die Daten Ende-zu-Ende (E2E) und sie verschickt..[44] Zudem bauen die vier Protokolle

- TLS-Handshake-Protokoll,

[38] *Kaur*, *R.*, *Kaur*, *A.*, 2012, S. 297.
[39] Ebd., S. 297.
[40] Aus Gründen der Überblicklichkeit und der Ähnlichkeit der beiden Zertifikate, werden im Folgenden SSL/TLS vereinfachend als SSL bezeichnet.
[41] *Zhang*, *L.* et al., 2014, S. 1.
[42] *nerdcoding*, 2018.
[43] *Cloudfare*, o. J., Grund Nr. 3.
[44] Vgl. Transport Layer Security (TLS) Funktionsweise & Erklärung, 2016.

- TLS-Change-Cipher-Spec. Protokoll,
- TLS-Alert Protokoll und das
- TLS-Application-Data Protokoll

auf das TLS-Record Protocol auf,[45] von denen das TLS-Handshake-Protokoll und das TLS-Alert Protokoll im Folgenden kurz beschrieben werden. Das TLS-Record Protocol sichert die Schutzziele Vertraulichkeit und Integrität.[46]

4.2.1.2 TLS-Handshake-Protokoll

Das TLS-Protokoll ist auch in sich noch in einzelne Unterprotokolle eingeteilt, eines davon ist das TLS-Handshake-Protokoll, dessen Verlauf in Abbildung 3[47] dargestellt wird.

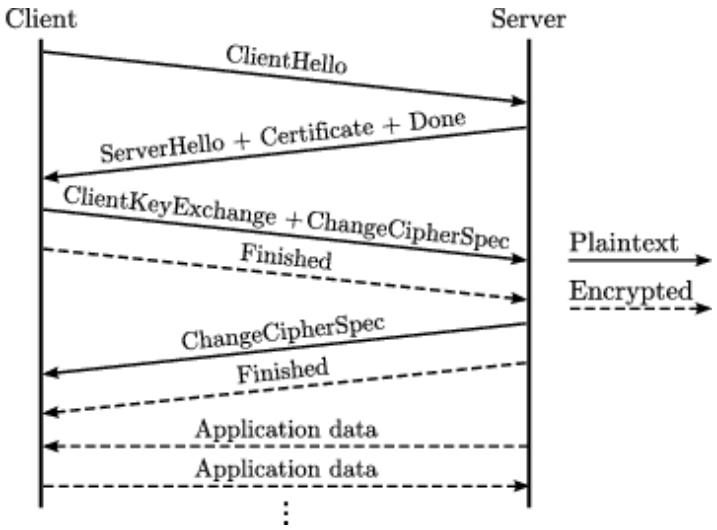


Abbildung 3: Darstellung eines SSL/TLS-Handshakes[48]

Mithilfe dieser Handshakes wird ausgehandelt, welche kryptographischen Algorithmen und Schlüssel verwendet werden und die Kommunikationspartner werden identifiziert und authentifiziert, wobei es üblich ist, dass sich nur der Server bei dem Client authentifiziert, was als one-way authentication bezeichnet wird.[49]

[45] Vgl. Transport Layer Security (TLS) Funktionsweise & Erklärung, 2016.
[46] Vgl. ebd.
[47] Aus *Husák*, *M.* et al., 2015, S. 2.
[48] Aus ebd., S. 3
[49] Vgl. *Morrissey*, *P.*, *Smart*, *N. P.*, *Warinschi*, *B.*, 2010, S. 191.

4.2.1.3 TLS-Alert Protokoll

Das TLS-Alert Protokoll ist für Alerts während der Übertragung zuständig. Es gibt verschieden Alerts, die in die zwei Stufen „Closure“ und „Error“ unterteilt werden, wobei Error Alerts zu einem sofortigen Abbruch der Verbindung führen.[50] Zudem sind sie, wie alle anderen Nachrichten im TLS-Protokoll verschluüsselt und komprimiert.[51]

4.2.2 HTTPS-Protokoll

Wie in Unterunterabschnitt 4.2.1 bereits dargestellt, sorgt SSL dafür, dass eine Anwendung oder ein Nutzer verifizieren kann, dass die angefragten Daten von dem Absender stammen, den man auch angefragt hat. Dies trägt besondere Relevanz bei Webanwendungen, da es sonst möglich ist, unter falschem Namen schädliche Software in ein System einzuspeisen. Das Hypertext Transfer Protocol (HTTP) überträgt Daten über das Transmission Control Protocol (TCP), wohingegen HTTPS die Daten verschlüsselt über das SSL-Protokoll verschickt.[52]

Eine HTTPS-Verschlüsselung einer Internetseite sorgt dafür, dass bei der Übertragung Daten, besonders vertrauliche wie z.B. Bankkontoinformationen, nicht von dritten eingesehen werden können.[53] Zudem sorgt das SSL-Zertifikat dafür, dass der Nutzer weiß, der Absender der Daten ist auch das System, das angefragt wurde, wie in Unterunterabschnitt 4.2.1 dargestellt. Dadurch wird verhindert, dass ein Angreifer einem Nutzer eine identisch aussehende Internetseite bereitstellt und dadurch die Nutzer glauben lässt, sie seien auf der gewünschten Internetseite.[54]

4.3 Passwortsicherheit — Key Derivation Function, Salted- & Peppered-Hashing

4.3.1 Key Derivation Function

Als Key Derivation Function (Schlüsselableitung) wird in der Kryptographie eine Operation bezeichnet, welche aus einem Schlüssel oder einem Passwort verschiedene ande-

[50] Vgl. Transport Layer Security (TLS) Funktionsweise & Erklärung, 2016.
[51] Vgl. *Rescorla*, *E.*, 2018, S. 85.
[52] *Naylor*, *D.* et al., 2014; *Dierks*, *T.*, *R.*, *R.*, 2008.
[53] Vgl. *Cloudfare*, o. J.
[54] Vgl. ebd.

re Schlüsselwerte generiert.[55] Diese Operation kann in verschiedenen Arten durchgeführt werden, zwei häufig angewandte sind zum einen das Salted Hashing (Unterunterabschnitt 4.3.2) und das Peppered Hashing (Unterunterabschnitt 4.3.3), bzw. das kombinierte Anwenden beider dieser Verfahren. Der generierte Schlüssel wird als Master Key (MC) bezeichnet.

4.3.2 Salted Hashing

Als Salt wird in der Kryptographie ein zufälliger String bezeichnet, welcher dem zu verschlüsselnden Klartext angehangen wird, bevor der Hash-Algorithmus durchgeführt wird.[56] Dabei wird am Beispiel von Anmeldedaten einer Internetseite für jeden angelegten Nutzer ein eigener Salt generiert und mit in der Datenbank gespeichert. Dies verhindert unter anderem, dass für zwei Anmeldedaten mit den gleichen Passwörtern der gleiche Hashwert in der Datenbank gespeichert wird.

Durch Salted Hashing das Erstellen einer Rainbow-Table überflüssig, beziehungsweise unnötig aufwändig gemacht, da nicht, wie bei einer normalen Hash-Datenbank ein Rainbow-Table für die gesamte Datenbank generiert werden muss, sondern für jedes Passwort ein Rainbow-Table benötigt wird.[57]

Der Salt hat, damit er ausreichende Sicherheit bieten kann, zwei Anforderungen. Die eine, Einzigartigkeit, wurde weiter oben bereits aufgeführt. Die zweite Anforderung ist, dass er optimalerweise eine bestimmte Länge haben sollte, allerdings gibt es hier keine genaue Anforderung/ Richtlinie, welche Länge als sicher gilt. Das Patent für Salted Hashing selbst gibt keine vor,[58] sodass auch innerhalb einer Datenbank verschiedene Längen für Salts verwendet werden können.

Der NIST-standard für Schlüsselableitung sieht jedoch vor, dass der MC, also die Kombination aus einem Passwort und einem Salt, mindestens eine Länge von 112 bits haben soll.[59]

4.3.3 Peppered Hashing

Analog zum Salt und dem Salted Hashing (Unterunterabschnitt 4.3.2) gibt es auch die Sicherungsmethode des Pepper, oder des Peppered Hashing.

[55] Vgl. *Sönmez Turan*, *M.* et al., 2010, S. 3.
[56] Vgl. *Rosulek*, *M.*, 2021, S. 205.
[57] Vgl. ebd., S. 205.
[58] Vgl. *Holland*, *R. C.*, *Stickle*, *T. C.*, *Beer*, *J. K.*, 2016, S. 14.
[59] Vgl. *Sönmez Turan*, *M.* et al., 2010, S. 6.

Bei diesem wird, wie beim Salted Hashing auch, ein zufällig generierter String an den Klartext angehangen, bevor der Hash-Algorithmus durchgeführt wird, jedoch ist dieser String aber nicht für jeden Einrag unterschiedlich, sondern wird bei der Einrichtung des Servers festgelegt und geheimgehalten.[60] Auch dieses Verfahren schützt vor einer möglichen SQL-Injection. Gleichermaßen gibt es beim Peppered Hashing keine fest vorgelegte Länge, welche der Pepper aufweisen sollte, allerdings empfiehlt es sich auch hier, dass der Pepper möglichst lang gewälht wird.

4.4 Authentifizierung und Autorisierung — Token-Verfahren, OAuth

Um Datensicherheit zu gewährleisten, bzw. Datenzugriff zu steuern, werden in Webanwendungen zwei Konzepte Verfolgt: Authentifizierung und Autorisierung.

Autorisierung befasst sich mit der Zugriffskontrolle über Daten, also welche Geräte oder Nutzer welche Daten lesen und/oder schreiben, Programme ausführen oder Akteure kontrollieren können.[61] Ebenso bezeichnet Autorisierung das Ablehnen oder Entfernen von Zugriff über Daten, z. B.für Malware.[62]

Demgegenüber bezeichnet Authentifizierung den Prozess, einen Client, Gerät oder Mensch, zu identifizieren und ist eine Grundvoraussetzung für Autorisierungen, da in den meisten Fällen Autorisierung ohne eine zuvorgehende Authentifizierung nicht möglich ist.[63] Häufig genutzte Methoden zur Authentifizierung sind die zuvor behandelten digitalen Zertifikate (Unterunterabschnitt 4.2.1), das HTTPS-Protocol (Unterunterabschnitt 4.2.2), sowie Schlüsselableitungen (Unterabschnitt 4.3). Diese werden genutzt, um das Vertrauen zwischen dem Client und der Anwendung herzustellen.[64]

4.4.1 Access Token

Eine häufig genutzte Art, die Autorisierung eines Clients zu prüfen ist es, einen Access Token zu verschicken. Diese bestehen üblicherweise aus drei Schlüssellementen,

- dem Header, mit Daten über die Art des Tokens und den Algorithmus, mit dem dieser generiert wurde,

[60] Vgl. *Webster*, *C. R.*, 2009.
[61] Vgl. *Kim*, *H.*, *Lee*, *E. A.*, 2017, S. 28.
[62] Vgl. ebd., S. 28.
[63] Vgl. ebd., S. 28.
[64] Vgl. ebd., S. 28.

- der Payload, welche die Informationen über den Client beinhält, darunter auch die Berechtigungen und Ablaufdaten, sowie
- die Signatur, ein Zertifikat (Unterunterabschnitt 4.2.1), das die Echtheit des Tokens garantiert;

zusätzlich können auch weitere Elemente wie z.B. Erweiterungen oder Metadaten angefügt sein, um fallspezifische Anforderungen zu erfüllen.[65]

Eine häufig genutzte Version von Access Token ist der JSON Web Token (JWT). Dieser ist zu großen Teilen bedingt durch seine kompakte Speichergröße.[66] Der JWT besteht dabei aus den oben beschriebenen Elementen, die bevor, mit einem '.' sie zusammengesetzt werden, mit einem Base64-Algorithmus verschlüsselt und komprimiert sind.[67]

Ein Minimalcodebeispiel um in der Programmiersprache JavaScript (JS) einen JWT zu generieren, sieht wie folgt aus

Listing 1: Beispielskript zur Generierung eines JWT

```
const header = {
        "alg": "HS256",
        "typ": "JWT"
}

const payload = {
  "sub": "1234567890",
  "name": "John Doe",
  "iat": 1516239022
}

const signature = HMAC_SHA(
        secret,
        base64urlEncoding(header) + '.' +
        base64urlEncoding(payload)
)


const token = base64urlEncodingheader + '.' + base64urlEncoding(payload + '.'
    base64urlEncoding(signature)
```

wobei `secret` für den private Key des digitalen Zertifikates steht (Unterunterabschnitt 4.2.1 steht, `base64urlEncoding()` eine Funktion ist, Daten mit einem Base64-Algorithmus zu verschlüsseln und `HMAC_SHA256()` eine Methode ist, um einen Hashed Message Authentication Code (HMAC) mit dem SHA256-Algorithmus zu erzeugen.

Dabei wird der Token

[65] Vgl. *Okta*, 2023.
[66] Vgl. *Jones*, *M. B.*, *Bradley*, *J.*, *Sakimura*, *N.*, 2015, S. 4.
[67] Vgl. ebd., S. 5.

```
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
    eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.
    SflKxwRJSMeKKF2QT4fwpMeJf36POk6yJV_adQssw5c"
```

generiert.

4.4.2 Open Authorization (OAuth)

OAuth bezeichnet ein Framework, welches einem Dateninhaber ermöglicht, Dritten begrenzten Zugriff zu gewähren, z. B.bei Single-Sign-Ons (SSOs), oder den gegebenen Zugriff zu beschränken.[68]

Das OAuth Protokoll verwendet die folgenden 6 Elemente,

- einen Client als den Service, der nach Authorisierung fragt,
- einem Ressourceninhaber als die Entität, die die Information, für die der Client Zugriff anfragt,
- einen Ressourcenserver als den Service, der den Zugriff zu den angefragten Informationen vergibt,
- einen Autorisierungsserver, der die Echtheit der Zertifikate des Ressourceninhabers verifiziert und die Überprüfungen für die Authorisierung durchführt,
- einen Token-Verfahren (Unterunterabschnitt 4.4.1), der von dem Autorisierungsserver generiert wird und dem Client den Zugriff von dem Ressourcenserver anfragen lässt und
- einen Authentifizierungscode, den der Autorisierungsserver während der Anfrage überprüfen kann,

um Zugriff anzufragen, beziehungsweise ihn zu überprüfen.[69]

Abbildung 4[70] stellt die Kommunikation zwischen dem Client und der ServiceApplication Programming Interface (API) dar. Zunächst stellt der Client oder eine Applikation eine Anfrage an den Ressourceninhaber nach einerAuthorisierung. Wenn diese an den Client bestätigt wird, stellt dieser mit der Bestätigung eine Anfrage nach einem Access Token an

[68] Vgl. *Hardt*, *D.*, 2012, S. 1; Vgl. *Leiba*, *B.*, 2012, S. 75.
[69] Vgl. *Leiba*, *B.*, 2012, S. 75.
[70] Vgl. *Hardt*, *D.*, 2012, S. 7, Abbildung 1.
[71] Vgl. ebd., S. 7, Abbildung 1

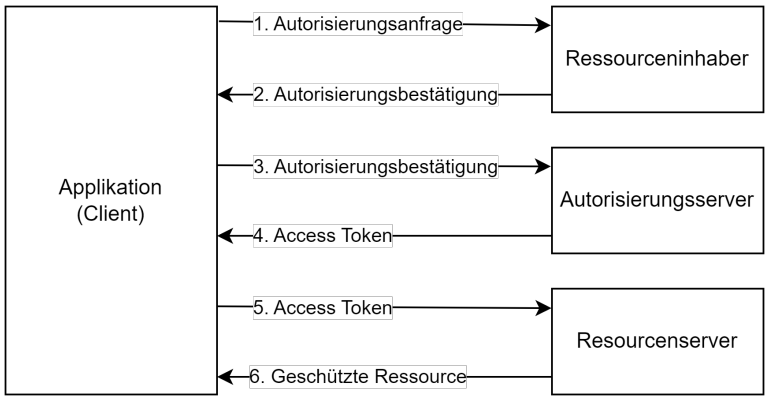


Abbildung 4: Kommunikationsverlauf im OAuth-Protokoll[71]

den Autorisierungsserver. Mit dem Access Token stellt der Client zuletzt noch eine Anfrage an den Ressourcenserver und erhält von diesem die gewünschten Ressourcen.

OAuth wird primär für SSOs genutzt, um z. B.einer Anwendung Zugriff auf bestimmte Daten einer anderen Anwendung zu gewähren, ohne die Anmeldedaten direkt zu übergeben, wie etwa Bilder aus einem Cloudservice an einen Druckservice zu übergeben.[72]

4.4.3 2-Faktor Authentifizierung (2FA)

Wie Tabelle 3[73] zeigt, sind Passwörter heutzutage mit Abstand die am weitesten verbreitete Methode, sich bei einem Service zu authentifizieren.

Tabelle 3: Statistik zu Authentifizierungsverfahren zum Schutz von Daten und Geräte[74]
Details: Deutschland; 08. bis 14.08.2018; 1.025 Befragte; ab 18 Jahre

Merkmal	Anteil der Befragten
Ich nutze ein Passwort	63
Ich nutze einen Fingerabdruckssensor	26
Ich nutze eine Zahlenkombination	22
Ich nutze ein Muster	12
Ich nutze eine Gesichtserkennung	9
Ich nutze eine Spracherkennung	8
ich nutze die Online-Ausweisfunktion meines Personalausweises	7
Ich nutze einen Iris-Scanner	7
Ich nutze einen Venen-Scanner	5
Sonstiges	0
Ich nutze keine Authentifizierungsverfahren	12

[72] Vgl. *Leiba*, *B.*, 2012, S. 74.
[73] *Statista*, 2019.

Jedoch bietet ein einzelnes Passwort als Authentifizierungsverfahren keinen vollständigen Schutz,[75] da Angreifer z. B.durch andere unsichere Software auf dem System einfach an die Nutzerdaten gelangen können. Besonder, wenn man für mehrere verschiedene Systeme das gleiche Passwort nutzt oder einen SSO-Service für mehrere Dienste nutzt, sind diese Angriffen deutlich leichter ausgesetzt als wenn für jeden Dienst ein anderes, sicheres, Passwort genutzt wird.

2FA-Dienste tragen dazu bei, dass diese Unsicherheiten abgeschafft werden, in dem z. B.ein zusätzliches, verknüpftes Gerät eine Benachrichtigung bekommt, in der man den Authentifizierungsvorgang bestätigen muss, wie es unter anderem verschiedene Banksysteme mit online banking und einem TAN-Mechanismus machen, oder ein externer Dienst für 2FA-One-Time Passwörter (OTPs) genutzt wird, die in bestimmten Zeitintervallen neue Codes generieren, der zusätzlich zu der primären Authentifizierungsmethode eingegeben werden müssen.

[74] *Statista*, 2019
[75] Vgl. *Jacomme*, *C.*, *Kremer*, *S.*, 2021, S. 2.

5 Herausforderungen bei der Implementierung von Kryptographie in Webanwendungen

In den meisten Fällen überwiegt der Schutz, den kryptographische Methoden bieten, den Aufwand und die Probleme, die bei der Implementierung auftreten können. Nichtsdestotrotz wird man versuchen, die Probleme, wie z. B. die in Unterabschnitt 5.1 beschriebenen Leistungseinbußen oder die in Unterabschnitt 5.3 erläuterte erschwerte Benutzbarkeit, zu beheben bzw. auf ein Minimum zu reduzieren.

5.1 Performance- und Skalierbarkeitsprobleme

Der in Unterunterabschnitt 4.2.2 vorgestellte Vorteil der verschlüsselten Übertragung bringt allerdings auch einen Nachteil mit sich. Durch die Verschlüsselung der Daten verringert sich die Übertragungsrate und verlängert sich die Latenz der Übertragung,[76] dies wird in Tabelle 4[77] dargestellt.

Tabelle 4: Parameter linearer Anpassungen an HTTP- und HTTPS-Übertragungen[78]

	Transferrate (bytes/ms)		Latenz (ms)	
Server	Netscape	Mircosoft	Netscape	Microsoft
Unsicher	946	829	4.5	19.0
Sicher 40 bit	730	689	3.6	3.9
Sicher 128 bit	736	686	25.0	5.1

Diese Daten sind allerdings aus dem Jahr 1998 und mittlerweile wurde HTTPS verbessert, sodass der Performanceaufwand, der nötig ist um eine versclhüsselte Verbindung aufzubauen verringert wurde.[79]

Neuere Versionen des TLS Algorithmus bringen noch deutlichere Performance-Verbesserungen mit sich, da unter anderem nur ein TLS-Handshake, behandelt in Ab-

[76] *Goldberg*, *A.*, *Buff*, *R.*, *Schmitt*, *A.*, 1998, S. 5.
[77] Übersetzt nach: Ebd., S. 5.
[78] ebd., S. 5
[79] Vgl. *Cloudfare*, o. J.

satz 4.2.1.2, benötigt wird, bzw. keiner, wenn bereits eine Verbindung zwischen Client und Server bestand.[80]

Außerdem ist, trotz des Verdoppelns von CPU Geschwindigkeiten alle 18 Monate, ist die Performance von SSL Maschinen noch immer ein Problem.[81]

Eine Möglichkeit, dies zu umgehen wäre es, das Entschlüsseln und Verifizieren des Zertifikates und der Signatur um die Identitaet des Clients zu Authentifizierungauthentifizieren, die Kommunikationsrichtung der Handshakes umzukehren, also das Verifizierun und Entschlüsseln der Daten Serverside anstatt Clientside zu berechnen,[82] wie die Abbildung 5[83] es darstellt. Da ein Großteil der Berechnungen für einen TLS-Handshake während des

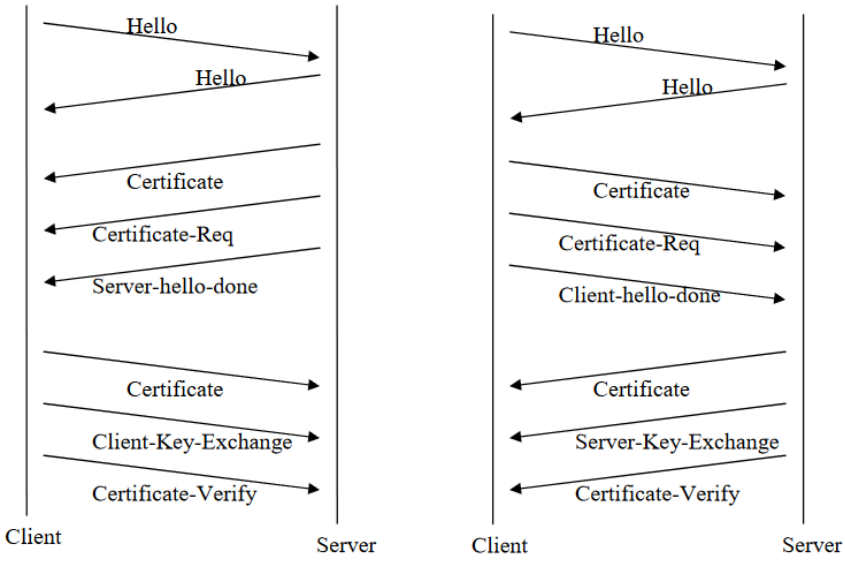


Abbildung 5: Gegenüberstellung zwischen einem normalen SSL-Handshake und einem umgekehrten SSL-Handshake[84]

Verifizieren der Zertifikate durchgeführt wird, wird die Laufzeit des Handshakes verbessert, wenn diese Berechnungen vom Server durchgeführt werden.

[80] Vgl. *Cloudfare*, o. J.
[81] Vgl. *Bicakci*, *K.*, *Crispo*, *B.*, *Tanenbaum*, *A. S.*, 2003, S. 2.
[82] Vgl. ebd., S. 3.
[83] Ebd., S. 3.
[84] ebd., S. 3

5.2 Komplexität und Fehleranfälligkeit der Kryptographie

5.2.1 Komplexität bei der Implementierung von kryptographischen Methoden in Webanwendungen

Durch die Vielzahl an Lektüre zu den verschiedenen Verschlüsselungsmethoden[85] ist es heutzutage nicht mehr so kompliziert, kryptographische Methoden in Webanwendungen einzubauen, um Daten und die Anwendung als Ganzes zu sichern. Gleichzeitig gibt es auch für viele Programmiersprachen mittlerweile vorgefertigte Methoden oder Packages, welche verschiedene kryptographische Methoden bereitstellen oder bündeln. So stellt die Programmiersprache JS die Funktion `digest(algorithm, data)` zur Verfügung, welche einen Datensatz nach einem der SHA-Algorithmen SHA-1, SHA256, SHA-384 oder SHA-512 verschlüsselt.[86] Die `decrypt(algorithm, key, data)` hingegen erlaubt es, mit RSA, beschrieben in RSA oder Advanced Encryption Standard (AES) Algorithmen zu verschlüsseln.

5.2.2 Fehleranfälligkeit von kryptographischen Methoden in Webanwendungen

Auch, wenn das TLS-Protokoll schon viel analysiert wurde,[87] sind die gefundenen Mängel und Schwachstellen eher gering.[88]

Definition 5.1 (Man-in-the-midle (MITM))**.** *MITM bezeichnet nach Request for Comments (RFC) №2828[89] eine Form des aktiven Lauschangriffs, bei dem der Angreifer kommunizierte Daten abfängt und selektiv verändert, um sich als sich als eine oder mehrere der an einer Kommunikationsbeziehung beteiligten Einheiten auszugeben.*

In einem typischen MITM Angriff stellt sich das angreifende System so zwischen den Client und den Server, sodass es mit den Client und den Server unabhängig voneinander kommunizieren kann, die beiden Parteien aber den Eindruck behalten, sie würden direkt miteinander kommunizieren; eine Möglichkeit, sich einen MITM Angriff vorzustellen ist es, als würde zwischen Client und Server ein SSL/TLS Proxy Server geschaltet sein.[90] Dadurch wissen weder der Client noch der Server über den MITM bescheid. Kryptographische Methoden ändern bei einem MITM Angriff nichts mehr an der Sicherheit, da der MITM in der

[85] z. B. *Davies*, *J.*, 2011.
[86] Vgl. *MDN Web Docs*, o. J.
[87] Siehe z. B. *Krawczyk*, *H.*, *Paterson*, *K. G.*, *Wee*, *H.*, 2013; *Paulson*, *L. C.*, 1999; *Dowling*, *B.* et al., 2015; *Cremers*, *C.* et al., 2017.
[88] Vgl. *Oppliger*, *R.*, *Hauser*, *R.*, *Basin*, *D.*, 2006, S. 2239.
[89] Übersetzt aus *Shirey*, *D. R.*, 2000, S. 105.
[90] Vgl. *Sounthiraraj*, *D.* et al., 2014, S. 4.

Verbindung integriert ist und alle Daten abfangen kann, so werden z. B. Anmeldedaten dem MITM sichtlich gemacht.[91]

5.3 Benutzerfreundlichkeit und Usability-Aspekte

5.4 Abwägung von Sicherheitsanforderungen und Nutzerbedürfnissen

[91] Vgl. *Sounthiraraj*, *D.* et al., 2014, S. 4.

6 Fazit

Anhang

Literaturverzeichnis

Bicakci, *Kemal*, *Crispo*, *Bruno*, *Tanenbaum*, *Andrew S.* (2003): Reverse SSL: Improved Server Performance and DoS Resistance for SSL Handshakes, in: o. O., 2003-06-26, [Zugriff: 2023-06-07]

Brinkmann, *Eilert* (2001): Einführung in die Kryptographie und Kryptoanalyse, Kryptographische Verfahren und ihre Anwendung, in: (2001)

Buchmann, *Johannes* (2016): Einführung in die Kryptographie, 6. Aufl., Springer-Lehrbuch, Berlin: Springer Spektrum, 2016, XXVI, 330 Seiten

Bundesamt für Sicherheit in der Informationstechnik (2022): Leitfaden zur Entwicklung sicherer Webanwendungen, Empfehlungen und Anforderungen an die Auftragnehmen, in: (2022), [Zugriff: 2023-05-23]

Cremers, *Cas*, *Horvat*, *Marko*, *Hoyland*, *Jonathan*, *Scott*, *Sam*, *van der Merwe*, *Thyla* (2017): A comprehensive symbolic analysis of TLS 1.3, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, o. O., 2017, S. 1773–1788

Davies, *Joshua* (2011): Implementing SSL/TLS using cryptography and PKI, o. O.: John Wiley und Sons, 2011

Davis, *Tom* (2003): RSA Encryption, Englisch, o. O.: geometer, 2003-10-10, URL: http://www.geometer.org/mathcircles/RSA.pdf [Zugriff: 2023-05-19]

Dierks, *T.*, *R.*, *Rescorla* (2008), RFC 5246: The Transport Layer Security (TLS) Protocol Version 1.2, o. O., 2008-08, URL: https://www.rfc-editor.org/rfc/rfc5246 [Zugriff: 2023-05-24]

Dowling, *Benjamin*, *Fischlin*, *Marc*, *Günther*, *Felix*, *Stebila*, *Douglas* (2015): A cryptographic analysis of the TLS 1.3 handshake protocol candidates, in: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, o. O., 2015, S. 1197–1210

Goldberg, *Arthur*, *Buff*, *Robert*, *Schmitt*, *Andrew* (1998): A COMPARISON OF HTTP AND HTTPS PERFORMANCE, in: (1998)

Guo, *Jiaxin*, *Hu*, *Yichen*, *Wang*, *Yixuan* (2022): Analysis and Illustration of the Enigma Machine. In: 2022 International Conference on Big Data, Information and Computer Network (BDICN), Big Data, Information and Computer Network (BDICN), 2022 International Conference on, BDICN (2022), S. 807–819

Hamer, *David H.* (1997): ENIGMA: Actions Involved in the 'Double Stepping' of the Middle Rotor, in: Cryptologia (1997), [Zugriff: 2023-05-18]

Hardt, *D.* (2012), The OAuth 2.0 Authorization Framework, RFC 6749, USA, 2012-10

Holland, *Ryan Cristopher*, *Stickle*, *Thomas Charles*, *Beer*, *John Kenneth* (2016): Salt Value Service, Englisch, US9246686B1 (USA), *Amazon Technologies*, *Inc.*, 2016-01-26, URL: https://patents.google.com/patent/US9246686B1/en [Zugriff: 2023-05-28]

Husák, *Martin*, *Cermak*, *Milan*, *Jirsik*, *Tomas*, *Celeda*, *Pavel* (2015): Network-based HTTPS Client Identification Using SSL/TLS Fingerprinting, in: 10th International Conference on Availabiity, Reliability and Security, o. O., 2015-08, S. 8

Jacomme, *Charlie*, *Kremer*, *Steve* (2021): An Extensive Formal Analysis of Multi-Factor Authentication Protocols, in: ACM Trans. Priv. Secur. 24 (2021), Nr. 2, [Zugriff: 2023-06-03]

Johnson, *Tom R.* (1998): American Cryptology during the Cold War, 1945 - 1989, Book 3: Retrenchment and Reform, 1972-1980, Englisch, Bd. 5.3, 6, o. O., 1998, Kap. 19: The Rebirth of Intelligence during the Carter Administration, S. 232, 262 S., [Zugriff: 2023-05-18]

Jones, *Michael B.*, *Bradley*, *John*, *Sakimura*, *Nat* (2015), JSON Web Token (JWT), RFC 7519, o. O., 2015-05, URL: https://www.rfc-editor.org/info/rfc7519 [Zugriff: 2023-06-01]

Kaur, *Ravneet*, *Kaur*, *Amandeep* (2012): Digital Signature, in: Conference Publishing Services (2012), S. 295–301, [Zugriff: 2023-05-21]

Kim, *Hokeun*, *Lee*, *Edward A.* (2017): Authentication and Authorization for the Internet of Things, in: IT Professional, 19 (2017), Nr. 5, S. 27–33

Krawczyk, *Hugo*, *Paterson*, *Kenneth G*, *Wee*, *Hoeteck* (2013): On the security of the TLS protocol: A systematic analysis, in: Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I, Springer, o. O., 2013, S. 429–448

Leiba, *Barry* (2012): OAuth Web Authorization Protocol, in: IEEE Internet Computing, 16 (2012), Nr. 1, S. 74–77, [Zugriff: 2023-06-02]

Morrissey, *P.*, *Smart*, *N. P.*, *Warinschi*, *B.* (2010): The TLS Handshake Protocol: A Modular Analysis, in: Journal of Cryptology, 23 (2010), Nr. 2, S. 187–223

Naylor, *David*, *Finamore*, *Alessandro*, *Leontiadis*, *Ilias*, *Grunenberger*, *Yan*, *Mellia*, *Marco*, *Munafò*, *Maurizio*, *Papagiannaki*, *Konstantina*, *Steenkiste*, *Peter* (2014): The Cost of the SSïn HTTPS, in: CoNEXT '14 (2014), S. 133–140

Oppliger, *Rolf*, *Hauser*, *Ralf*, *Basin*, *David* (2006): SSL/TLS session-aware user authentication – Or how to effectively thwart the man-in-the-middle, in: Computer Communications, 29 (2006), Nr. 12, S. 2238–2246

Paulson, *Lawrence C* (1999): Inductive analysis of the internet protocol TLS, in: ACM Transactions on Information and System Security (TISSEC), 2 (1999), Nr. 3, S. 332–351

Preneel, *Bart* (2003): Analysis and Design of Cryptographic Hash Functions, Englisch, thesis, Belgium: COSIC, 2003, 323 S., URL: https://www.e-reading.club/bookreader.

php/141503/Analysis_and_Design_of_Cryptographic_Hash_Functions.pdf [Zugriff: 2023-05-24]

Rescorla, *Eric* (2018), The Transport Layer Security (TLS) Protocol Version 1.3, 8446, o. O., 2018-08, URL: https://www.rfc-editor.org/info/rfc8446

Rosulek, *Mike* (2021): The Joy of Cryptography, Englisch, Corvallis, Oregon, USA, 2021-01-03, 286 S., [Zugriff: 2023-05-28]

Shebius, *Arthur* (1918): Chiffriermaschine, 416219 (Berlin), 1918-02-23, URL: https://www.dpma.de/docs/dpma/veroeffentlichungen/de416219a_chiffriermaschiene1918.pdf [Zugriff: 2023-05-10]

Shirey, *Dr. Rob* (2000), Internet Security Glossary, 2828, o. O., 2000-05, URL: https://www.rfc-editor.org/info/rfc2828 [Zugriff: 2023-06-02]

Sönmez Turan, *Meltem*, *Barker*, *Elaine*, *Burr*, *William*, *Chen*, *Lily* (2010), Reccomendation for Password-Based Key Derivation, Part 1: Storage Applications, SP 800-132, United States of America, 2010-12-22, URL: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-132.pdf [Zugriff: 2023-05-30]

Sounthiraraj, *David*, *Sahs*, *Justin*, *Greenwood*, *Garret*, *Lin*, *Zhiqiang*, *Khan*, *Latifur* (2014): Smv-hunter: Large scale, automated detection of ssl/tls man-in-the-middle vulnerabilities in android apps, in: Network and Distributed System Security Symposium (NDSS). Internet Society, San Diego, CA, o. O., 2014, S. 1–14

Statista (2019): Welche Authentifizierungsverfahren nutzen Sie selber zum Schutz ihrer Daten und Geräte, Statista Umfrage Cybersecurity & Cloud 2018, statista, o. O.: statista, 2019-01-03, URL: https://de.statista.com/prognosen/952951/umfrage-in-deutschland-zu-authentifizierungsverfahren-zum-schutz-von-daten [Zugriff: 2023-06-03]

Thielert, *Sandra* (2007): Kryptographische Algorithmen, Wernigerode: Rechenzentrum Hochschule Harz, 2007-05-11, URL: https://www.hs-harz.de/dokumente/extern/Rechenzentrum/Grundschutz/Kryptografische_Algorithmen.pdf [Zugriff: 2023-05-18]

Zhang, *Liang*, *Choffnes*, *David*, *Levin*, *Dave*, *Dumitraş*, *Tudor*, *Mislove*, *Alan*, *Schulman*, *Aaron*, *Wilson*, *Christo* (2014): Analysis of SSL Certificate Reissues and Revocations in the Wake of Heartbleed, Englisch, in: Proceedings of the 2014 Conference on Internet Measurement Conference, IMC '14, Vancouver, BC, Canada: Association for Computing Machinery, 2014-11-05, S. 489–502, [Zugriff: 2023-05-22]

Internetquellen

Beutelspacher, *Albrecht* (2017): Eine kurze Geschichte der Kryptografie, bpb.de, <https://www.bpb.de/shop/zeitschriften/apuz/259145/eine-kurze-geschichte-der-kryptografie/> (2017-11-10) [Zugriff: 2023-04-11]

Cloudfare (o. J.): Warum HTTPS verwenden?, <https://www.cloudflare.com/de-de/learning/ssl/why-use-https/> (keine Datumsangabe) [Zugriff: 2023-05-28]

Encryption Consulting LLC (2023): What is SHA? What is SHA used for? | Encryption Consulting, <https://www.encryptionconsulting.com/education-center/what-is-sha/> (2023-05) [Zugriff: 2023-05-21]

Krawisz, *Daniel* (2013): Chapter 2: Public-Key Cryptography, Crypto-Anarchy and Libertarian Entrepreneurship, <https://nakamotoinstitute.org/mempool/crypto-anarchy-and-libertarian-entrepreneurship-2/> (2013-05-24) [Zugriff: 2023-05-24]

MDN Web Docs (o. J.): SubtleCrypto: digest() method - Web APIs, <https://developer.mozilla.org/en-US/docs/Web/API/SubtleCrypto/digest> (keine Datumsangabe) [Zugriff: 2023-06-08]

nerdcoding (2018): How TLS/SSL and X.509 really works, <https://www.nerdcoding.org/post/2018/2018-12-16-tls-x509/> (2018-12-16) [Zugriff: 2023-05-23]

Okta (2023): Access Token: Definition, Architecture, Usage & More | Okta, <https://www.okta.com/identity-101/access-token/> (2023-02-14) [Zugriff: 2023-05-31]

Petitcolas, *Fabian* (o. J.): The information hiding homepage, Kerckhoffs's principles from « La cryptographie militaire », <https://www.petitcolas.net/kerckhoffs/index.html> (keine Datumsangabe) [Zugriff: 2023-04-14]

Transport Layer Security (TLS) Funktionsweise & Erklärung (2016), <https://www.kryptowissen.de/transport-layer-security-tls.php> (2016-08-30) [Zugriff: 2023-06-07]

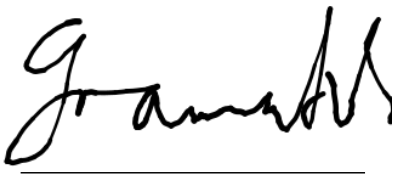
Webster, *Craig R.* (2009): Securing Passwords with Salt, Pepper and Rainbows — Barking Iguana, <http://www.barkingiguana.com/2009/08/03/securing-passwords-with-salt-pepper-and-rainbows/> (2009-08-03) [Zugriff: 2023-05-28]

Ehrenwörtliche Erklärung

Hiermit versichere ich, dass die vorliegende Arbeit von mir selbstständig und ohne unerlaubte Hilfe angefertigt worden ist, insbesondere dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen sind, durch Zitate als solche gekennzeichnet habe. Ich versichere auch, dass die von mir eingereichte schriftliche Version mit der digitalen Version übereinstimmt. Weiterhin erkläre ich, dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde/Prüfungsstelle vorgelegen hat. Ich erkläre mich damit einverstanden/nicht einverstanden, dass die Arbeit der Öffentlichkeit zugänglich gemacht wird. Ich erkläre mich damit einverstanden, dass die Digitalversion dieser Arbeit zwecks Plagiatsprüfung auf die Server externer Anbieter hochgeladen werden darf. Die Plagiatsprüfung stellt keine Zurverfügungstellung für die Öffentlichkeit dar.

Ahaus, 9.6.2023

(Ort, Datum)



(Eigenhändige Unterschrift)