

TH Brandenburg  
Online Studiengang IT Sicherheit  
Fachbereich Informatik und Medien  
Netzwerksicherheit  
Prof. Dr. Michael Pilgermann

Einsendeaufgabe 2  
Wintersemester 2023  
Abgabetermin 19. November 2023

Gruppe 11  
Mathias Baumbach (Matr-Nr. 20213703)  
Mara Schulke (Matr-Nr. 20215853)

**Zusammenfassung****Inhaltsverzeichnis**

<b>1</b>	<b>Durchführung</b>	<b>2</b>
1.1	SSL Test I . . . . .	2
1.2	SSL Test II . . . . .	6
1.3	SSL Test III . . . . .	10
1.4	Verschlüsselung . . . . .	13
1.5	Steganographie . . . . .	16

**Abbildungsverzeichnis**

1	QualysSSL Test – nwsmooc.mooiin.org . . . . .	3
2	QualysSSL Test – Recent Worst . . . . .	5
3	QualysSSL Test – Browser Capabilities . . . . .	6
4	QualysSSL Test – Browser Protocol Features . . . . .	8
5	Google Chrome vs Firefox – 442 – Unsafe / Unsafe . . . . .	10
6	Google Chrome vs Firefox – 444 – Safe / Unsafe . . . . .	10
7	Google Chrome vs Firefox – 444 – Details . . . . .	11
8	Google Chrome vs Firefox – 444 – Chrome Zertifikat . . . . .	11
9	Vercrypt – Exponent . . . . .	13
10	Vercrypt – Einbinden Container . . . . .	14
11	Vercrypt – Passworteingabe . . . . .	14
12	Vercrypt – Einbinden Versteckter Container . . . . .	15
13	Vercrypt – Versteckter Container . . . . .	15
14	Steganographie – SilentEye UI . . . . .	16
15	Steganographie – JPG Entschlüsselung . . . . .	17
16	Steganographie – BMP Entschlüsselung . . . . .	17

# 1 Durchführung

## 1.1 SSL Test I

### Aufgabenstellung

Testen Sie den Server `nwsmooc.mooin.org` mit der SSL-Testseite von Qualys und erklären Sie die Ergebnisse (hinsichtlich Zertifikaten, TLS-Versionen, Handshakes und Details der Protokolle). Erklären Sie für eine weitere Webpräsenz, die als "Recent Worst" bewertet wird, was bei dieser nicht stimmt.


Hinweis: "Recent Worst" ist eine Liste auf der rechten Seite der Testseite. Zur besseren Nachvollziehbarkeit bitte Screenshots hinzufügen. Deren Inhalte sollen aber jeweils von Ihnen erklärt werden.

## Antwort

SSL Report: [nwsmooc.mooin.org](https://nwsmooc.mooin.org) (81.169.230.170)

### Summary

Overall Rating



Certificate: 100  
Protocol Support: 100  
Key Exchange: 100  
Cipher Strength: 100

Visit our [documentation page](#) for more information, configuration guides, and books. Known issues are documented [here](#).

HTTP Strict Transport Security (HSTS) with long duration deployed on this server. [MORE INFO >](#)

### Certificate #1: RSA 2048 bits (SHA256withRSA)

**Server Key and Certificate #1**

Subject	nwsmooc.mooin.org Fingerprint SHA256: 290e3c7218642c2c178ea29b953add8598b81c95d5a18d99287ee10578 Pin SHA256: TKu6QN8dmYU0g0cKyK5aQ8PomPX0vc5AN0HwGTZW+
Common names	nwsmooc.mooin.org
Alternative names	nwsmooc.mooin.org
Serial Number	04f51de4ee860a0a7462a384897eca1b44dc
Valid from	Mon, 25 Sep 2023 22:58:29 UTC
Valid until	Sun, 24 Dec 2023 22:58:28 UTC (expires in 1 month and 9 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	R3 AIA: <a href="http://r3.o.lencor.org/">http://r3.o.lencor.org/</a>
Signature algorithm	SHA256withRSA
Extended Validation	No
Certificate Transparency	Yes (certificate)
OCSP Must Staple	No
Revocation information	OCSP OCSP: <a href="http://r3.o.lencor.org/">http://r3.o.lencor.org</a>
Revocation status	Good (not revoked)
DNS CAA	No ( <a href="#">more info</a> )
Trusted	Yes Mozilla Apple Android Java Windows

**Additional Certificates (if supplied)**

Certificates provided: 3 (3953 bytes)

Chain issues: None

**#2**

Subject	R3 Fingerprint SHA256: 67ad41168b030a0e1b856c96813c04c3a5f599902798865572a3c7e737613dd Pin SHA256: jQU7hHt9w01THSUmWb+Fw0Gpgt21gT3P-PKGD+
Valid until	Mon, 15 Sep 2025 16:00:00 UTC (expires in 1 year and 9 months)
Key	RSA 2048 bits (e 65537)
Issuer	ISRG Root X1
Signature algorithm	SHA256withRSA

**#3**

Subject	ISRG Root X1 Fingerprint SHA256: 6d99fb265eb1c5b37447653dc6483c08e1bfa054c299b6947d77f1c24f Pin SHA256: C5+hpZ7icVwmwQIMcRIPsQWLABX0Qzpa0wHFfBM=
Valid until	Mon, 30 Sep 2024 18:14:03 UTC (expires in 10 months and 14 days)
Key	RSA 4096 bits (e 65537)
Issuer	DST Root CA X3
Signature algorithm	SHA256withRSA

**Certification Paths**

[Click here to expand](#)

Abbildung 1: QualysSSL Test – nwsmooc.mooin.org

Der Web-Server hat insgesamt eine A+-Bewertung erhalten und ist dementsprechend auf dem aktuellen Stand / robust was die SSL-Konfiguration angeht. Das Zertifikat des Servers ist ein 2048-Bit RSA-Zertifikat, das mit SHA256+RSA signiert wurde. In dem Report von Qualys sind einige wichtige Eckdaten über das Zertifikat des Servers enthalten:

- Gültigkeit: Von 25. September 2023 bis zum 24. Dezember 2023
- Aussteller: R3
- Status: Das Zertifikat wurde nicht widerrufen.
- Vertrauenswürdigkeit: Ja – Mozilla, Apple, Android, Java und Windows vertrauen diesem Zertifikat.

Außerdem gibt der Bericht die Zertifikate weiter oben in der Zertifikats-Kette an:

- Zertifikat #2: Aussteller R3, RSA 2048 Bit, gültig bis 15. September 2025.
- Zertifikat #3: Aussteller ISRG Root X1, RSA 4096 Bit, gültig bis 30. September 2024.

—

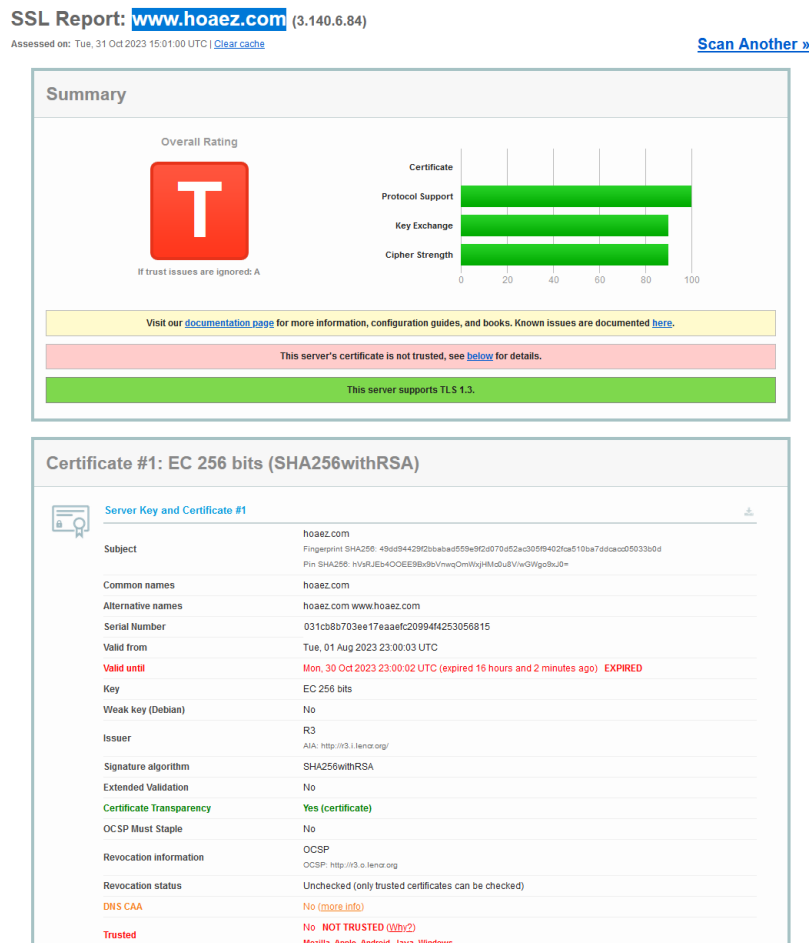


Abbildung 2: QualysSSL Test – Recent Worst

Der Server [www.hoaez.com](https://www.hoaez.com) hat lediglich eine Gesamtbewertung von T erhalten, was bedeutet, dass dem Zertifikat des Servers nicht vertraut werden kann (e.g. da es abgelaufen ist, oder durch eine Misskonfiguration). Um dennoch einen Report erstellen zu können, hat Qualys die Vertrauenswürdigkeit bis auf weiteres ignoriert.

Unter der Zusammenfassung gibt es einen Abschnitt, der Details zum Zertifikat des Servers liefert. Diese deuten explizit darauf hin, dass das Zertifikat (zum Test-Zeitpunkt seit 16 Stunden) abgelaufen und somit nicht mehr gültig ist. Es ist anzunehmen, dass durch eine Erneuerung des Zertifikates, alle Probleme behoben werden können und der Server wieder eine ausreichend gute Bewertung erhalten würde.

Das Zertifikat an sich ist verwendet Elliptic-Curve mit 256 Bits und wurde ebenfalls mit SHA256+RSA signiert.

Dieser Report ist ein gutes Beispiel dafür, als Administrator im Idealfall auf sich-selbsterneuernde Zertifikate zu setzen, um einem Vertrauensverlust der Nutzer (durch eine Warnmeldung des Browsers) entgegen zu wirken.

## 1.2 SSL Test II

### Aufgabenstellung

Führen Sie den Client-Test von Qualys aus und erklären Sie die Ergebnisse. (Siehe <https://clienttest.ssllabs.com:8443/ssltest/viewMyClient.html>)

### Antwort

Wir haben den Client-Test mit der Mozilla Firefox Version 119.0.1 durchgeführt und folgendes Ergebnis erhalten:

The screenshot displays the Qualys SSL Labs test results for a browser. The header includes the Qualys logo and navigation links. The main title is 'SSL/TLS Capabilities of Your Browser'. Below the title, the user agent string is shown: 'User Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/119.0'. The results are organized into five sections, each with a title and a status message:

- Protocol Support**: Your user agent has good protocol support. Your user agent supports TLS 1.2 and TLS 1.3, which are recommended protocol version at the moment.
- CVE-2020-0601 (CurveBall) Vulnerability**: Your user agent is not vulnerable. For more information about the CVE-2020-0601 (CurveBall) Vulnerability, please go to [CVE-2020-0601](https://cve-2020-0601). To test manually, click [here](#). Your user agent is not vulnerable if it fails to connect to the site.
- Logjam Vulnerability**: Your user agent is not vulnerable. For more information about the Logjam attack, please go to [weakdh.org](https://weakdh.org). To test manually, click [here](#). Your user agent is not vulnerable if it fails to connect to the site.
- FREAK Vulnerability**: Your user agent is not vulnerable. For more information about the FREAK attack, please go to [www.freakattack.com](https://www.freakattack.com). To test manually, click [here](#). Your user agent is not vulnerable if it fails to connect to the site.
- POODLE Vulnerability**: Your user agent is not vulnerable. For more information about the POODLE attack, please read [this blog post](#).

Abbildung 3: QualysSSL Test – Browser Capabilities

Im Kopf der Ergebnisseite wird eine Zusammenfassung der Bewertung ausgegeben. Es wird zunächst die Protokoll-Unterstützung mit grün/gut bewertet – in unserem Fall ob TLS 1.2 und TLS 1.3 unterstützt wird. Danach folgen spezifische Schwachstellentests:

- **Curveball / CVE-2020-0601** - Schwachstelle im Windows-OS bei der Validierung von digitalen Zertifikaten die potentiell angreifbar für MitM macht.
- **Logjam** - Sicherheitslücke im Diffie-Hellman-Schlüsselaustausch, welcher bei verschlüsselten Verbindungen wie TLS verwendet wird. Ermöglicht den Schlüsselaustausch zu manipulieren und verschlüsselte Verbindungen zu entschlüsseln.

- **FREAK (Factoring RSA Export Keys)** – Sicherheitslücke, die Angreifern es ermöglichte Verschlüsselung von HTTPS-Verbindungen zu umgehen.
- **POODLE (Padding Oracle On Downgraded Legacy Encryption)** - Schwachstelle in SSLv2/3 die es ermöglichte die Verschlüsselung zu umgehen.

Keine der Schwachstellen sind auf unserem Client vorhanden.







Protocol Features			
 <b>Protocols</b>			
TLS 1.3			Yes
TLS 1.2			Yes
TLS 1.1			No
TLS 1.0			No
SSL 3			No
SSL 2			No
 <b>Cipher Suites (In order of preference)</b>			
TLS_AES_128_GCM_SHA256 (0xc1301) Forward Secrecy			128
TLS_CHACHA20_POLY1305_SHA256 (0xc1303) Forward Secrecy			256
TLS_AES_256_GCM_SHA384 (0xc1302) Forward Secrecy			256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b) Forward Secrecy			128
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f) Forward Secrecy			128
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 (0xc0c9) Forward Secrecy			256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (0xc0cb) Forward Secrecy			256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc030) Forward Secrecy			256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc034) Forward Secrecy			256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA (0xc00a) VIEAK			256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA (0xc009) VIEAK			128
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA (0xc013) VIEAK			128
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA (0xc014) VIEAK			256
TLS_RSA_WITH_AES_128_GCM_SHA256 (0xc01c) VIEAK			128
TLS_RSA_WITH_AES_256_GCM_SHA384 (0xc01d) VIEAK			256
TLS_RSA_WITH_AES_128_CBC_SHA (0xc01f) VIEAK			128
TLS_RSA_WITH_AES_256_CBC_SHA (0xc01e) VIEAK			256
(1) When a browser supports SSL 2, its SSL 2-only suites are shown only on the very first connection to this site. To see the suites, close all browser windows, then open this exact page directly. Don't refresh.			
 <b>Protocol Details</b>			
Server Name Indication (SNI)		Yes	
Secure Renegotiation		Yes	
TLS compression		No	
Session tickets		No	
OCSP stapling		Yes	
Signature algorithms		SHA256/ECDSA, SHA384/ECDSA, SHA512/ECDSA, RSA_PSS_SHA256, RSA_PSS_SHA384, RSA_PSS_SHA512, SHA256/RSA, SHA384/RSA, SHA512/RSA, SHA1/ECDSA, SHA1/RSA	
Named Groups		x25519, secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072	
Next Protocol Negotiation		No	
Application Layer Protocol Negotiation		Yes	h2 http/1.1
SSL 2 handshake compatibility		No	
Mixed Content Handling			
 <b>Mixed Content Tests</b>			
Images	Passive		No
CSS	Active		No
Scripts	Active		No
XMLHttpRequest	Active		No
WebSockets	Active		No
Frames	Active		No
(1) These tests might cause a mixed content warning in your browser. That's expected. (2) If you see a failed test, try to reload the page. If the error persists, please get in touch.			
<b>Related Functionality</b>			
Upgrade Insecure Requests request header ( <a href="#">more info</a> )			Yes

Abbildung 4: QualysSSL Test – Browser Protocol Features

Es folgt eine Detailauflistung der Protokoll-Features, die ergänzt ist um eine Auflistung der unterstützten Cipher Suites in Reihenfolge der Präferenz (von oben nach unten). Dabei werden viele kryptographische Verfahren als grün/gut bewertet, einige aber als weak. Dies ist z.B. auf die Duldung von SHA (anstatt SHA256) zurückzuführen, oder weil der CBC-Modus (Cipher Block Chaining) erlaubt wird. CBC hat bekannte Schwächen, beispielsweise auf Angriffe wie den sogenannten "Padding Oracle Attack". Die Ermöglichung von CBC in Verbindung mit älteren Authentifizierungsalgorithmen wie SHA1 wird hier vermutlich als unsicher (auf englisch: weak) angesehen.

Unter der Überschrift **Protocol Details** wird als gut/grün hervorgehoben, dass TLS compression deaktiviert ist. Dies war ein Feature in älteren TLS-Versionen und sollte Daten komprimieren vor der Verschlüsselung, war jedoch angreifbar und wurde daher in Folgeversion deaktiviert. Ebenfalls deaktiviert sein sollte die SSL 2 handshake compatibility – dies ist die bereits erwähnte POODLE Schwachstelle. Die letzte farbliche Hervorhebung in diesem Abschnitt lautet **Session tickets – No**. Warum die Deaktivierung von Session-Tickets als weak gekennzeichnet wurde, konnten wir nicht 100% feststellen. Der Blog der Betreiber-Webseite beschreibt diesen Eintrag hauptsächlich aus Sicht eines Web-Server/-Seiten Betreibers so, „dass session tickets ein alternativer Session-Management Mechanismus sei, welcher separate Encryption Keys verwendet, die selten rotiert werden und lieber nicht verwendet werden sollten, wenn man die Implementation nicht genau im Detail versteht.“ (vgl. <https://blog.qualys.com/product-tech/2013/06/25/ssl-labs-deploying-forward-secrecy>, abgerufen am 09.11.2023). Es erschien uns eher als Vorteil denn als Nachteil dass dieses Feature im Firefox deaktiviert sei.

Den Abschluss der Ergebnisseite bildet der Abschnitt **Mixed Content Handling**, in dem die Kompatibilität mit diversen Webtechnologien (z.B. CSS, Frames usw.) getestet wird.

### 1.3 SSL Test III

#### Aufgabenstellung

Rufen Sie die Test-Webseiten <https://aaacertificateservices.comodoca.com:442/> sowie <https://aaacertificateservices.comodoca.com:444/> mit Firefox und mit Google Chrome auf. Erklären Sie, was passiert. Sind die Resultate bei allen vier Tests so wie erwartet?

#### Antwort

Um das Verhalten der beiden Browser näher zu betrachten haben wir uns dazu entschieden, die Tests simultan durchzuführen, um zu veranschaulichen, wann die beiden Browser sich anders verhalten. Als erstes haben wir den Server auf dem Port 442 getestet:

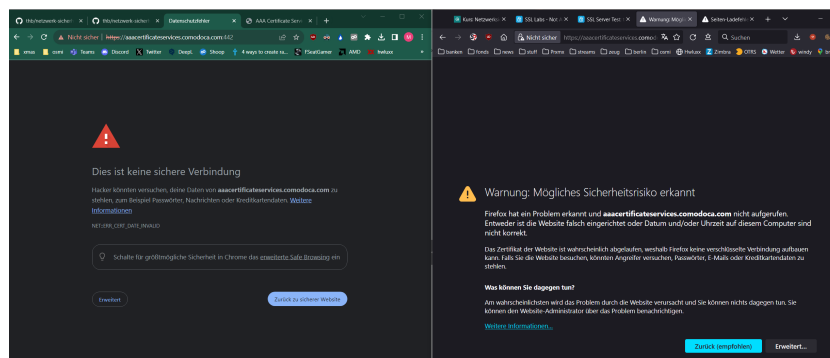


Abbildung 5: Google Chrome vs Firefox – 442 – Unsafe / Unsafe

Bei diesem Web-Server / Port erkennen beide Browser direkt ein abgelaufenes Zertifikat und warnen den Nutzer davor, diese Webseite zu verwenden (da die Identität des Servers nicht sicher gestellt werden konnte).

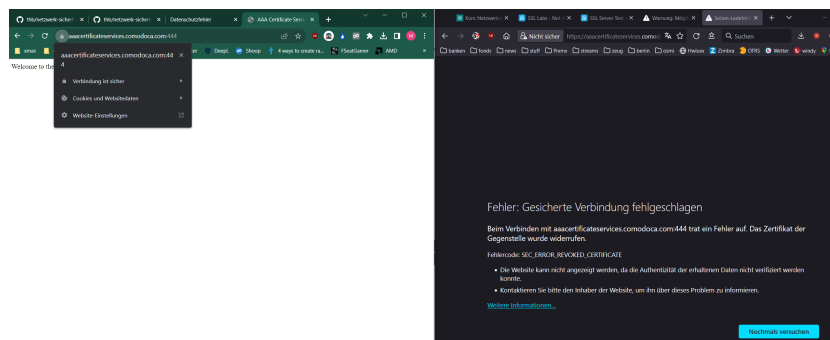


Abbildung 6: Google Chrome vs Firefox – 444 – Safe / Unsafe

Bei dem zweiten Test (also dem Server auf Port 444) verhalten sich die Browser interessanterweise unterschiedlich: Chrome zeigt dies als sichere Verbindung an, obwohl Firefox warnt und angibt, dass das Zertifikat widerrufen worden sei.

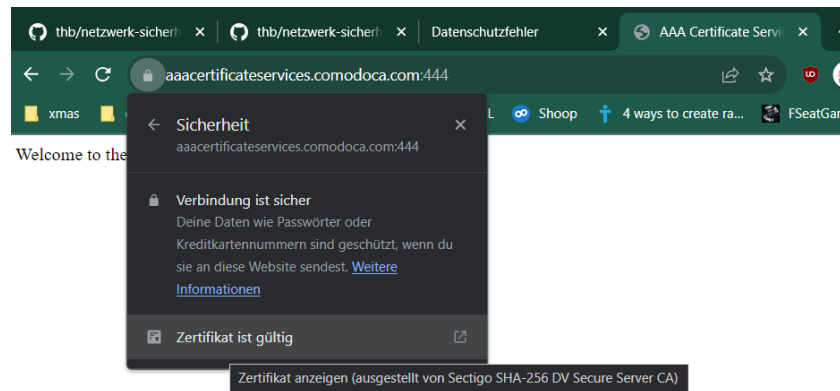


Abbildung 7: Google Chrome vs Firefox – 444 – Details

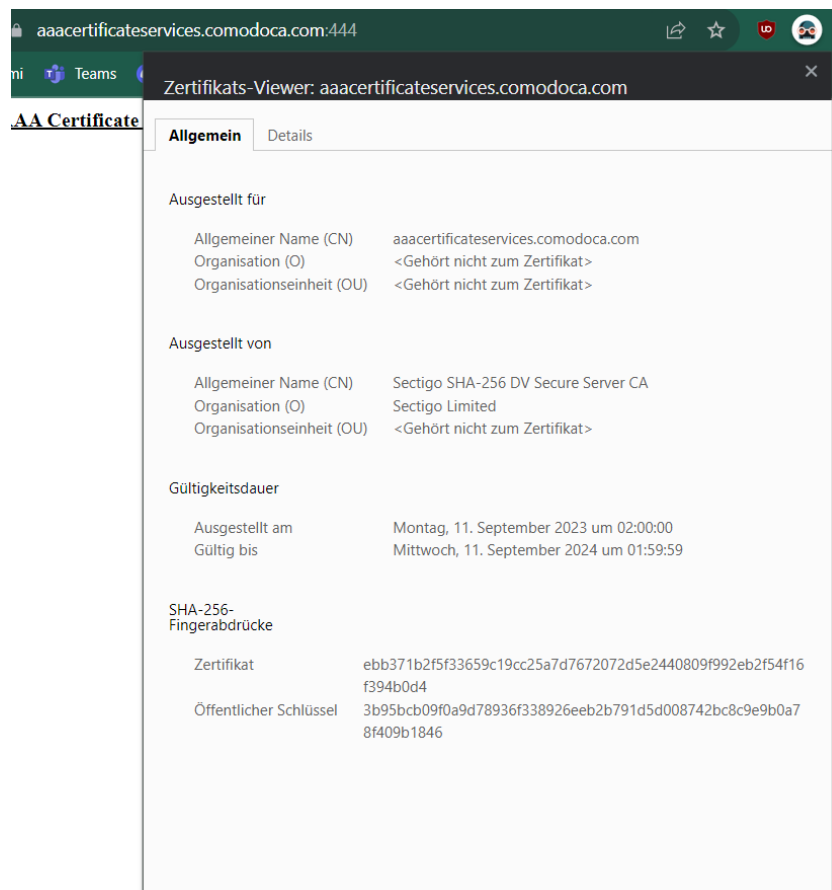


Abbildung 8: Google Chrome vs Firefox – 444 – Chrome Zertifikat

Auch bei näherer Betrachtung des Zertifikats im Chrome-Certificate-Viewer, lässt sich keine Spur von dem Widerruf finden.

Die unterschiedlichen Verhaltensweisen der Browser könnten mehrere Gründe haben:

1. **CRL und OCSP:** Chrome und Firefox verwenden wahrscheinlich unterschiedliche Methoden zur Überprüfung des Widerrufsstatus eines Zertifikats. Chrome überprüft möglicherweise nicht den Widerrufsstatus oder konnte den Certificate Revocation List (CRL) oder Online Certificate Status Protocol (OCSP) Server nicht erreichen. Wenn der Server nicht erreichbar ist, kann Chrome die Verbindung unter bestimmten Bedingungen zulassen.
2. **Caching:** Es könnte ein Caching-Problem vorliegen, bei dem Chrome das Zertifikat oder seinen Status zwischengespeichert hat und keinen Live-Check durchführt, um zu sehen, ob das Zertifikat widerrufen wurde.
3. **Widerrufsinformationen:** Es ist auch möglich, dass Firefox aktualisierte Widerrufsinformationen hat, die Chrome nicht hat. Dies kann passieren, wenn die Widerrufsinformationen kürzlich aktualisiert wurden und Chrome das Update noch nicht erhalten hat.
4. **Browser-Konfiguration:** Die Sicherheitseinstellungen des Browsers könnten unterschiedlich konfiguriert sein. Zum Beispiel könnte Chrome so eingerichtet sein, dass es mit bestimmten Arten von Zertifikatsfehlern fortfährt, die Firefox nicht zulässt, oder umgekehrt.
5. **Certificate Pinning:** Chrome könnte das Zertifikat für diese Seite gepinnt haben, was die normale Überprüfung der Vertrauenskette umgehen würde, die sonst zu einer erkannten Widerrufsstatus führen könnte.

Da wir allerdings beide Browser kontrollieren und beide Browser zum Zeitpunkt des Tests auf dem neusten Stand waren und wir den Web-Server vorher noch nie verwendet haben, deutet dies auf eine Diskrepanz bei den CRL / OCSP Abfragen hin – wahrscheinlich, da unterschiedliche Informationen verwendet werden.

In einem solchen Fall, sollte immer dem Browser mit Sicherheitsbedenken vertraut werden.

## 1.4 Verschlüsselung

### Aufgabenstellung

Installieren Sie VeraCrypt auf Ihrem Rechner. Hierzu erhalten Sie zusätzlich eine VeraCrypt-Datei. In der Datei ist ein normaler und ein versteckter Container zu finden, die jeweils eine Datei enthalten. Das Passwort für den normalen Container ist der Exponent  $e$  des RSA-Schlüssels vom `nwsmooc.moo.in.org`-Server. Dokumentieren Sie Ihre Vorgehensweise mit Screenshots und geben Sie anschließend das im versteckten Container gefundene Kennwort an.

Hinweis: Zur Bedienung von VeraCrypt können Sie sich beispielsweise hier ein Video ansehen: <https://youtu.be/atb2pdx394>.

### Antwort

Um das zur Verfügung gestellte VeraCrypt File öffnen zu können, soll der Exponent  $e$  des RSA-Schlüssels der Webseite `nwsmooc.moo.in.org` verwendet werden. Diesen Exponenten kann man sich ganz einfach ausgeben lassen, in dem man sich im Mozilla Firefox Browser jene Webseite öffnet und auf das Schlosssymbol drückt, um sich das Zertifikat anzusehen.

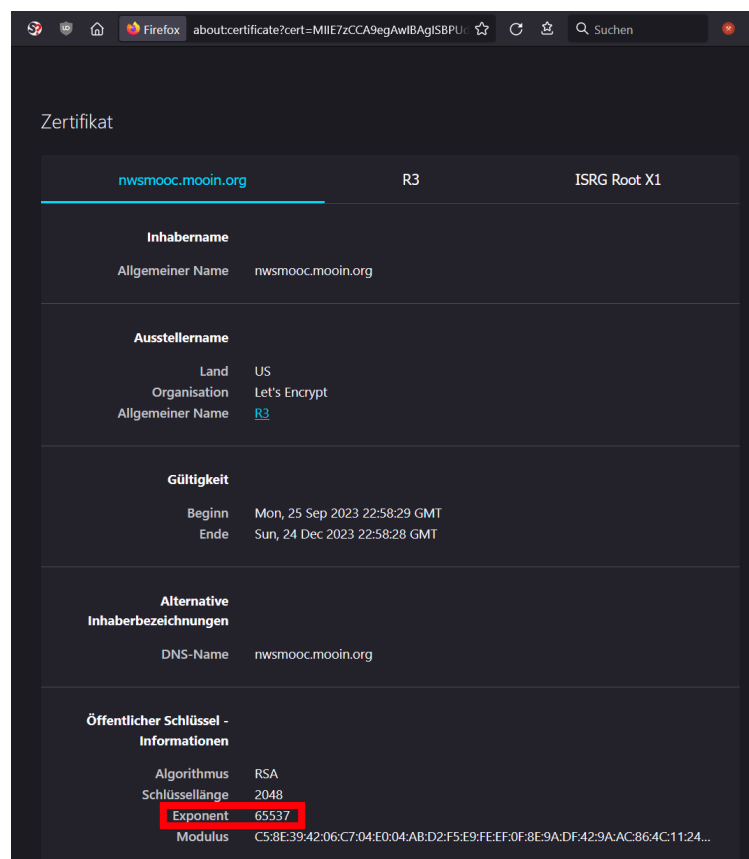


Abbildung 9: Vercrypt – Exponent

Im Abschnitt “Öffentlicher Schlüssel – Informationen” bekommt man den Exponenten des

öffentlichen Schlüssels ausgegeben (siehe Abb. 9). Allerdings gelang uns der Zugriff auf den Container auch vor der Analyse des Zertifikats, da  $e = 65537_{10}$  in der Praxis üblich ist.

Für das RSA Verfahren werden große Primzahlen bevorzugt verwendet und der Algorithmus muss dabei unterschiedliche Operationen bei der Bearbeitung basierend auf dem Binärkode tätigen. Da  $65537_{10} = 10000000000000001_2$  führt dies zu einer effizienten Arbeitsweise des Algorithmus – Einsen werden multipliziert, Nullen potenziert.

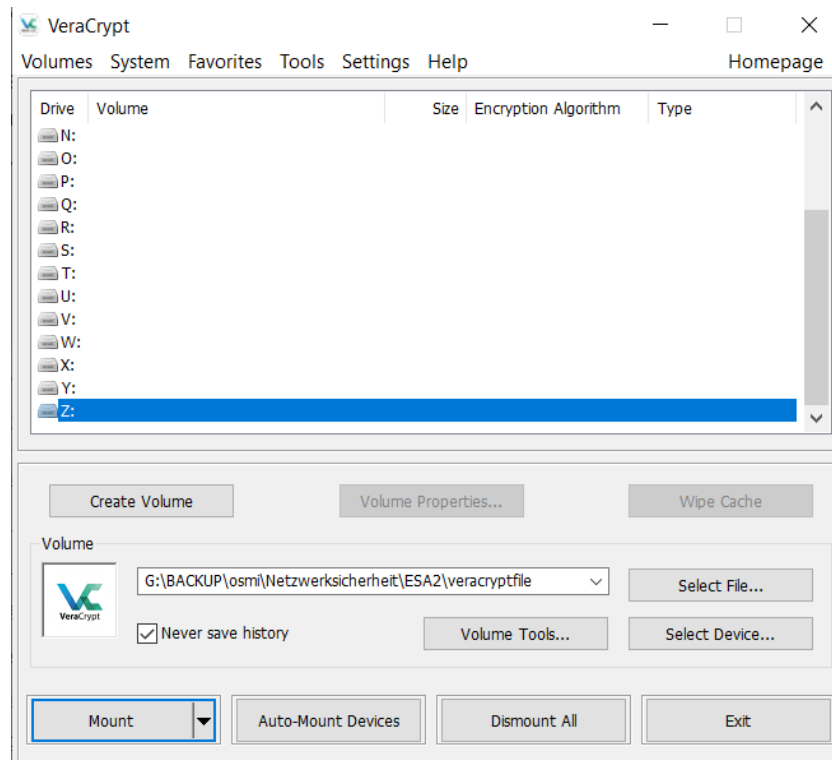


Abbildung 10: Vercrypt – Einbinden Container

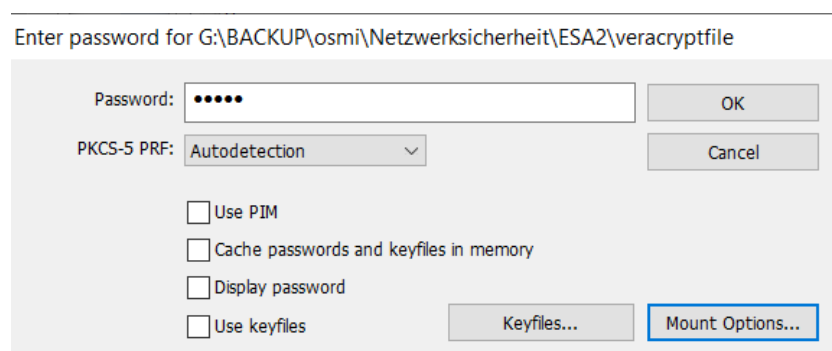


Abbildung 11: Vercrypt – Passworteingabe

Nun können wir mit VeraCrypt (unter Angabe des Passworts 65537) die im Kurs bereitge-

stellte Datei als Laufwerk einbinden und somit Zugriff auf den Inhalt erhalten.

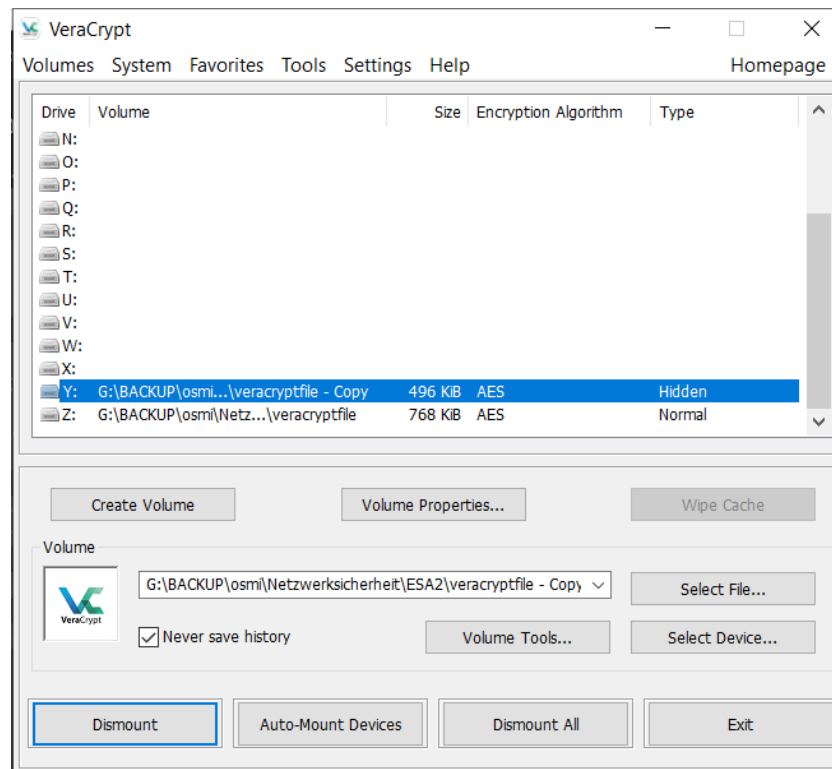


Abbildung 12: Vercrypt – Einbinden Versteckter Container

Die enthaltene Textdatei “Datei-angezeigter-Container.txt” lautet: “Das Zugangskennwort für den versteckten Container lautet: VerSteCon0815”

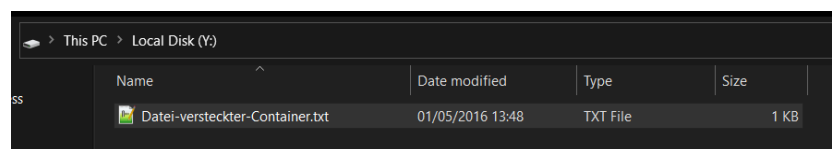


Abbildung 13: Vercrypt – Versteckter Container

Wir konnten uns an das Modul Digitaler Selbstschutz erinnern, in dessen Report- Inhalt VeraCrypt bereits vorkam, und haben den Container kopiert und ein 2. Mal eingebunden, wobei dieses Mal das Passwort “VerSteCon0815” eingegeben werden muss um Zugriff auf den versteckten Container zu erhalten.

Somit konnten wir Zugriff auf den versteckten Container und seinem Inhalt “Datei- versteckter-Container.txt” erlangen und den Inhalt auslesen: “HiddenVolumeVeraCrypt”.



## 1.5 Steganographie

### Aufgabenstellung

Installieren Sie SilentEye auf Ihrem Rechner und untersuchen Sie die bereitgestellten Beispieldateien. `mooin.jpg` enthält ein verstecktes Kennwort, `mooin.bmp` eine Datei. Die notwendigen Einstellungen können Sie der Aufgabe mit VeraCrypt entnehmen. Dokumentieren Sie Ihre Vorgehensweise mit Screenshots und geben Sie das gefundene Kennwort an.

Hinweis: Sollte es Schwierigkeiten geben, wenn Sie beide Dateien nacheinander untersuchen, dann schließen Sie SilentEye zwischendurch.

### Antwort

SilentEye ist eine Open-Source-Software, die zur Steganographie verwendet wird, um zusätzliche Informationen in Bildern und Audiodateien zu verstecken oder auszulesen. Wir haben die Software aus dem Internet heruntergeladen und unter Windows installiert. Nach dem Öffnen der Software ist es erforderlich eine Bild-Datei auszuwählen, von der der Nutzer weiß oder vermutet, dass diese eine mittels Steganographie codierte Nachricht enthält. (Siehe Abb. 14)

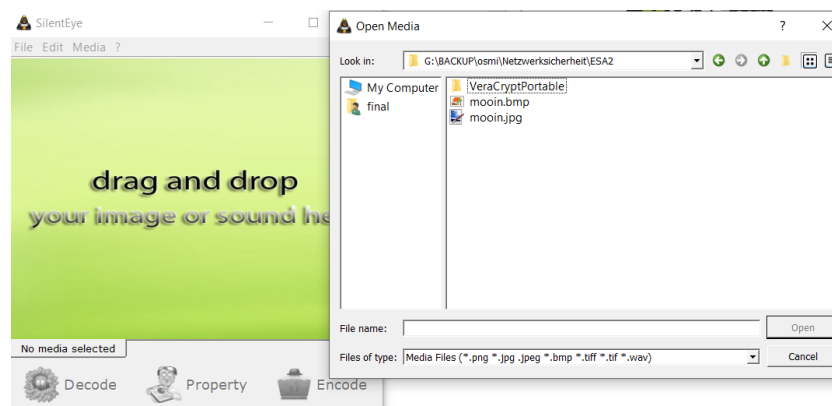


Abbildung 14: Steganographie – SilentEye UI

Um der Aufgabenstellung entsprechend nun die Datei `mooin.jpg` zu untersuchen, öffnen wir die Datei und verwenden die Einstellungen aus der Lösung der Aufgabe 1.4. Anschließend zeigt SilentEye, unter Verwendung der richtigen Parameter, eine decodierte Nachricht in einem Textfeld an – diese lautet **Kennwort: SteganographieInNWSMOOC** (Siehe Abb. 15)

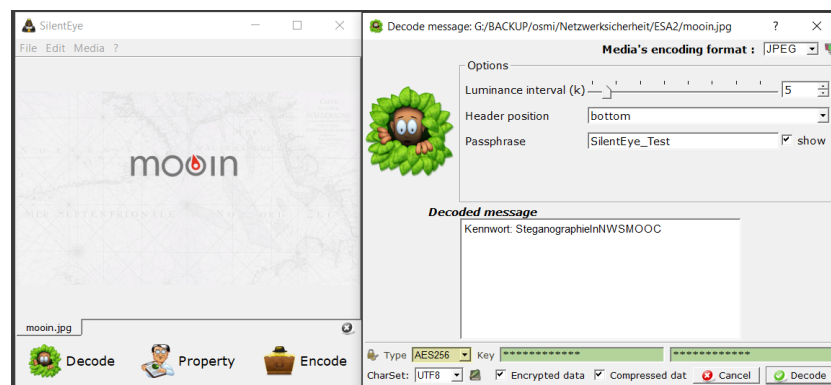


Abbildung 15: Steganographie – JPG Entschlüsselung

Mit diesem Kennwort und den bereitgestellten Einstellungen für das zweite Bild, lässt sich auch dessen versteckter Inhalt sichtbar machen und entschlüsseln. Wir können an dieser Stelle die Datei AufgabensammlungSS16.pdf extrahieren und abspeichern. (Siehe Abb. 16)

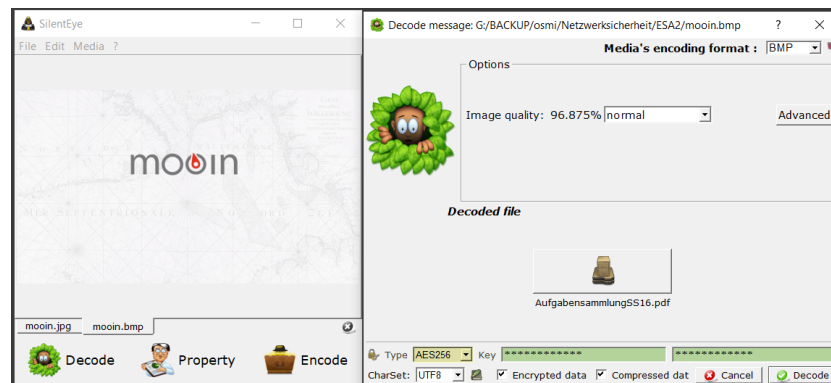


Abbildung 16: Steganographie – BMP Entschlüsselung