Wie HTTPS funktioniert

Lassen Sie mich erklären, wie HTTPS funktioniert, einschließlich SSL, Zertifikaten, dem Verschlüsselungsprozess und den Unterschieden zwischen TLS 1.2 und TLS 1.3, auf eine klare und schrittweise Weise.

Was ist HTTPS?

HTTPS steht für **Hypertext Transfer Protocol Secure**. Es ist die sichere Version von HTTP, dem Protokoll, das zum Übertragen von Daten zwischen Ihrem Browser und einer Website verwendet wird. Der „sichere“Teil kommt durch das Hinzufügen einer Verschlüsselungsschicht und Authentifizierung, die durch **SSL (Secure Sockets Layer)** oder dessen modernes Pendant, **TLS (Transport Layer Security)**, bereitgestellt wird. HTTPS ist also im Wesentlichen HTTP, das über TLS (oder historisch SSL) läuft, um Ihre Daten sicher zu halten.

Wie funktioniert HTTPS? Der TLS-Handshake

Wenn Sie eine Website mit HTTPS besuchen (z. B. `https://example.com`), stellt Ihr Browser und der Server der Website eine sichere Verbindung her, bevor sie Daten austauschen. Dieser Prozess wird als **TLS-Handshake** bezeichnet. Hier ist, wie es in einfachen Schritten funktioniert:

1. **Client Hello**:
    - Ihr Browser sendet eine Nachricht an den Server mit der Aussage: „Hallo! Hier ist die TLS-Version, die ich unterstütze (z. B. TLS 1.3), die Verschlüsselungsalgorithmen (Cipher-Suites), die ich verwenden kann, und eine zufällige Zeichenkette (client random).“
2. **Server Hello**:
    - Der Server antwortet: „Hallo zurück! Ich werde diese TLS-Version und diese Cipher-Suite aus Ihrer Liste verwenden. Hier ist meine zufällige Zeichenkette (server random).“
3. **Zertifikat**:
    - Der Server sendet sein **SSL-Zertifikat**, das seinen **öffentlichen Schlüssel** enthält und von einer vertrauenswürdigen **Zertifizierungsstelle (CA)** signiert ist. Dieses Zertifikat beweist die Identität des Servers.
4. **Client Key Exchange**:

- Ihr Browser überprüft das Zertifikat, um sicherzustellen, dass es gültig und von einer vertrauenswürdigen CA signiert ist. Wenn es besteht, generiert der Browser ein **pre-master secret**, verschlüsselt es mit dem öffentlichen Schlüssel des Servers und sendet es an den Server.

5. **Sitzungsschlüssel**:
   - Sowohl der Browser als auch der Server verwenden die client random, server random und pre-master secret, um unabhängig denselben **Sitzungsschlüssel** zu generieren. Dieser Schlüssel wird verwendet, um alle Daten während der Sitzung zu verschlüsseln und zu entschlüsseln.
6. **Finished**:
   - Beide Seiten senden eine „finished“-Nachricht, die mit dem Sitzungsschlüssel verschlüsselt ist, um zu bestätigen, dass die sichere Verbindung bereit ist.

Sobald der Handshake abgeschlossen ist, werden alle Daten (wie Webseiten, Formulare oder Dateien) mit dem Sitzungsschlüssel verschlüsselt, sodass sie für jeden, der sie abfangen könnte, unlesbar sind.

Was sind SSL-Zertifikate und wie funktionieren sie?

Ein **SSL-Zertifikat** ist ein digitales Dokument, das die Identität einer Website nachweist und die Verschlüsselung ermöglicht. Hier ist, was Sie wissen müssen:

- **Inhalt**: Das Zertifikat enthält den Domänennamen der Website, ihren öffentlichen Schlüssel und eine digitale Signatur von einer **Zertifizierungsstelle (CA)**.
- **Zweck**: Es stellt sicher, dass der Server legitim ist (z. B., dass Sie sich wirklich mit `example.com` verbinden und nicht mit einer gefälschten Seite) und stellt den öffentlichen Schlüssel für die Verschlüsselung bereit.
- **Überprüfung**: Ihr Browser überprüft:
  1. Ist das Zertifikat gültig (nicht abgelaufen oder widerrufen)?
  2. Ist es von einer vertrauenswürdigen CA signiert? (Browsern haben eine eingebaute Liste vertrauenswürdiger CAs wie DigiCert oder Let’s Encrypt.)
- Wenn die Überprüfungen bestehen, vertraut der Browser dem Server und setzt den Handshake fort.

Die CA fungiert wie eine vertrauenswürdige dritte Partei, die für die Website bürgt. Ohne dies könnten Angreifer so tun, als wären sie jede beliebige Seite und Ihre Daten stehlen.

Der Verschlüsselungsalgorithmus

Die Verschlüsselung in HTTPS basiert auf einer Kombination aus **asymmetrischer** und **symmetrischer Kryptographie**:

1. **Asymmetrische Kryptographie** (während des Handshakes):
   - Verwendet einen **öffentlichen Schlüssel** (öffentlich geteilt) und einen **privaten Schlüssel** (vom Server geheim gehalten).
   - Der Browser verschlüsselt das pre-master secret mit dem öffentlichen Schlüssel des Servers. Nur der Server kann es mit seinem privaten Schlüssel entschlüsseln.
   - Beispielalgorithmen: RSA oder Elliptic Curve Cryptography (ECC).
2. **Symmetrische Kryptographie** (für die Sitzung):
   - Sobald der Sitzungsschlüssel erstellt ist, verwenden beide Seiten ihn, um Daten zu verschlüsseln und zu entschlüsseln.
   - Dies ist schneller als asymmetrische Verschlüsselung und ideal für große Datenübertragungen.
   - Beispielalgorithmus: AES (Advanced Encryption Standard).

Der Handshake verwendet asymmetrische Verschlüsselung, um den Sitzungsschlüssel sicher zu teilen, dann übernimmt die symmetrische Verschlüsselung für Effizienz.

Unterscheide zwischen TLS 1.2 und TLS 1.3

TLS 1.2 und **TLS 1.3** sind Versionen des TLS-Protokolls, wobei TLS 1.3 die neuere, verbesserte Version ist. Hier sind die wesentlichen Unterschiede:

Merkmal	**TLS 1.2**	**TLS 1.3**
Handshake-Geschwindigkeit	Mehrere Hin- und Rückwege zwischen Client und Server, was die Latenz erhöht.	Weniger Hin- und Rückwege, oft nur einer, für schnellere Verbindungen. Kann Daten sofort („0-RTT“) bei Wiederverbindungen senden.
Cipher-Suites	Unterstützt ältere, weniger sichere Verschlüsselungsalgorithmen (z. B. SHA-1).	Entfernt veraltete Cipher-Suites, verwendet nur moderne, sichere (z. B. AES-GCM).
Datenschutz	Einige Handshake-Daten (wie Zertifikate) werden im Klartext gesendet.	Fast der gesamte Handshake ist verschlüsselt, verbirgt mehr Details vor Lauschern.

Merkmal	TLS 1.2	TLS 1.3
0-RTT-Wiederherstellung	Nicht verfügbar.	Ermöglicht sofortiges Senden von Daten bei Wiederverbindungen, birgt jedoch das Risiko von Wiederholungsangriffen, wenn nicht ordnungsgemäß gesichert.
Sicherheit	Anfällig für bestimmte Angriffe aufgrund älterer Funktionen.	Durch Design stärker, entfernt schwache Optionen.

Kurz gesagt, ist TLS 1.3 **schneller**, **sicherer** und **privater** als TLS 1.2, was es zum Standard für modernes HTTPS macht.

Alles zusammenfassen

HTTPS sichert die Webkommunikation durch: 1. Verwenden des **TLS-Handshakes**, um eine verschlüsselte Verbindung einzurichten. 2. Verlass auf **SSL-Zertifikate** von vertrauenswürdigen CAs, um die Identität des Servers zu überprüfen. 3. Kombinieren von **asymmetrischer Kryptographie** (für den Handshake) und **symmetrischer Kryptographie** (für die Datenübertragung), um Ihre Daten zu schützen. 4. Entwickeln mit Protokollen wie **TLS 1.3**, das TLS 1.2 mit besserer Geschwindigkeit, Sicherheit und Privatsphäre verbessert.

Das ist der Kern, wie HTTPS Ihr Surfen sicher hält! Lassen Sie mich wissen, wenn Sie möchten, dass ich tiefer in einen Teil eintauche.