

[zurück](#)

# Man-in-the-Middle-Angriff bei HTTPS (TLS)

## Ziel des Artikels

Dieser Artikel beschreibt **schrittweise**, wie ein klassischer **Man-in-the-Middle-Angriff (MITM)** auf eine HTTPS-Verbindung funktioniert, **wenn der Client einem manipulierten Zertifikat vertraut**.

Der Fokus liegt auf:

- dem TLS-Handshake
- der Rolle von Zertifikaten
- der **korrekten Interpretation der Sitzungsschlüssel-Erzeugung**

## Beteiligte Rollen

Rolle	Beschreibung
<b>Alice</b>	Client (Browser, App)
<b>Bob</b>	Echter HTTPS-Server
<b>Mallory</b>	Angreifer im Datenpfad (MITM)

## Wichtige Voraussetzung für den Angriff

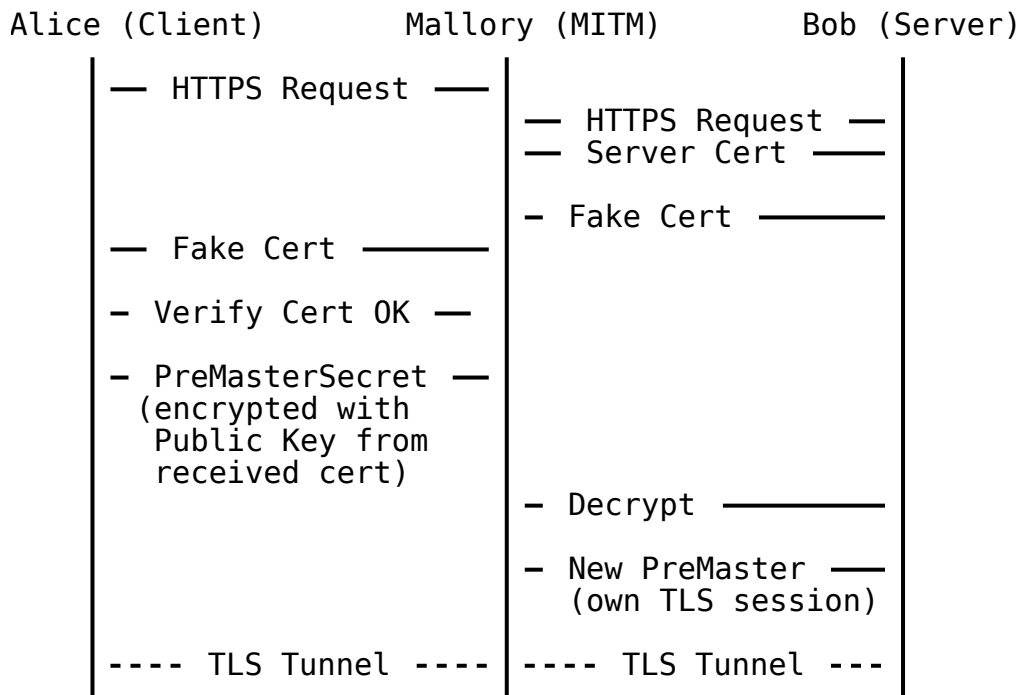


**Der Angriff funktioniert nur, wenn Alice das MITM-Zertifikat akzeptiert.**

Beispiele:

- Installierte Root-CA (Unternehmens-Proxy, Malware)
- Ignorierte Browser-Warnung
- Fehlendes Certificate Pinning

## TLS-MITM - Ablauf (ASCII-Sequenzdiagramm, a2s)



## Schritt-für-Schritt-Erklärung

### 1. Client startet HTTPS

Alice möchte eine HTTPS-Verbindung zu Bob aufbauen.

### 2. MITM übernimmt die Verbindung

Mallory sitzt im Datenpfad (z. B. WLAN, Proxy, Router) und leitet die Anfrage weiter.

### 3. Bob sendet sein echtes Zertifikat

Bob schickt sein **legitimes TLS-Zertifikat** an Mallory.

### 4. Mallory erzeugt ein eigenes Zertifikat

- Gleicher Hostname
- Eigenes Schlüsselpaar
- Signiert durch eine CA, der Alice vertraut

### 5. Mallory sendet Fake-Zertifikat an Alice

Alice erhält **nicht Bobs Zertifikat**, sondern Mallorys.

## 6. Alice prüft das Zertifikat

Das Zertifikat wird akzeptiert → Angriff läuft weiter.

---

## □ Kritischer Punkt: Sitzungsschlüssel (korrigiert)

### 7. Alice erzeugt das Pre-Master-Secret

Standard-TLS-Vorgang.

### 8. **\*\*Alice verschlüsselt das Pre-Master-Secret\*\***

#### Korrekt:

Alice verschlüsselt das Pre-Master-Secret mit dem **\*\*öffentlichen Schlüssel aus dem Zertifikat, das sie für das Serverzertifikat hält\*\***.

#### Wichtig:

- Alice weiß **nicht**, dass es ein MITM-Zertifikat ist
- Sie verwendet **keinen „MITM-Schlüssel“ bewusst**

→ Begriff „MITM-Public-Key“ ist aus Client-Sicht falsch

### 9. Mallory entschlüsselt das Pre-Master-Secret

Da Mallory den **Private Key** besitzt, kann sie den Schlüssel lesen.

### 10. Mallory startet eine zweite TLS-Verbindung

Mallory baut eine **eigene, echte TLS-Verbindung** zu Bob auf.

---

## Ergebnis: Zwei getrennte TLS-Tunnel

Verbindung	Inhalt
-----	-----
Alice ↔ Mallory	Vollständig entschlüsselbar für Mallory
Mallory ↔ Bob	Reguläre TLS-Verbindung

Mallory kann:

- Daten lesen
- Daten verändern
- Daten neu verschlüsseln

Ohne dass Alice oder Bob es merken.

---

## Merksätze (prüfungsrelevant)

**\*\*TLS schützt nicht vor MITM, sondern vor unbekanntem MITM.\*\***

> **\*\*Der Client weiß nie, dass es ein MITM ist – sonst wäre der Angriff gescheitert.\*\***

> **\*\*HTTPS-Sicherheit basiert auf Vertrauen in Zertifikate, nicht auf Verschlüsselung allein.\*\***

---

## Typische Schutzmechanismen

Maßnahme	Wirkung
HSTS	Erzwingt HTTPS
Certificate Pinning	Blockiert fremde Zertifikate
Keine fremden Root-CAs	Verhindert Proxy-MITM
Benutzer ignoriert Warnungen nicht	effektiv, aber selten ☐

---

## Fazit

Ein MITM-Angriff auf HTTPS ist **kein Kryptobruch**, sondern ein **Vertrauensbruch**. Wer die Zertifikatskette kontrolliert, kontrolliert die Verbindung.

---

Wenn du willst, mache ich dir als Nächstes:

- ☐ eine **AP1/AP2-Kurzfassung**
- ☐ eine **Vergleichsseite: echter TLS-Handshake vs. MITM**
- ☐ oder eine **„Warum Firmen HTTPS mitlesen dürfen“-Erklärung**

Sag einfach Bescheid.

From:

<http://wiki.nctl.de/dokuwiki/> - **Veni. Vidi. sudo rm -rf / vici.**

Permanent link:

<http://wiki.nctl.de/dokuwiki/doku.php?id=it-themen:grundlagen:netzwerkdienste:mitm&rev=1766479578>

Last update: **23.12.2025 09:46**

