

Data encryptie en veilige communicatie

- Uitleg
- Opdracht

Uitleg

Waarom encryptie? Encryptie is het omzetten van leesbare data (plaintext) naar onleesbare data (ciphertext) met een sleutel. Alleen wie de juiste sleutel heeft, kan de ciphertext weer omzetten naar plaintext. Hierdoor blijft informatie vertrouwelijk, zelfs als iemand de data onderschept.

Praktijkvoorbeeld: Stel je stuurt een e-mail met een wachtwoord in plaintext. Als een aanvaller het netwerkverkeer afluistert, kan die het wachtwoord gemakkelijk stelen. Met encryptie is dit onmogelijk zonder de sleutel.

Symmetrische encryptie

- **Werking:** Dezelfde sleutel wordt gebruikt om te versleutelen én ontsleutelen.
- **Voordelen:** Snel en efficiënt, geschikt voor grote hoeveelheden data.
- **Nadelen:** Sleutelbeheer is lastig: hoe deel je de sleutel veilig met de ontvanger?

Algoritmevoorbeeld: AES (Advanced Encryption Standard) is de meest gebruikte symmetrische cipher. AES-256 gebruikt een 256-bits sleutel en wordt beschouwd als zeer veilig.

Asymmetrische encryptie

- **Werking:** Er is een sleutel-paar: een publieke sleutel (public key) en een private sleutel (private key).
- **Gebruik:** Iedereen kan met de publieke sleutel versleutelen, maar alleen de houder van de private sleutel kan ontsleutelen.
- **Voordelen:** Oplost het sleuteluitwisselingsprobleem.
- **Nadelen:** Trager dan symmetrische encryptie, niet geschikt voor grote bestanden.

Algoritmevoorbeeld: RSA gebruikt twee grote priemgetallen om sleutelparen te genereren. Een bericht geëncrypt met de publieke sleutel kan alleen ontsleuteld worden met de private sleutel.

Hybride encryptiesystemen

In de praktijk combineer je beide: gebruik asymmetrische encryptie om een symmetrische sessiesleutel uit te wisselen, en symmetrische encryptie voor de eigenlijke data-overdracht.

Voorbeeld: HTTPS (TLS) werkt zo:

1. **Handshak:** Browser en server wisselen certificaten uit en genereren een gedeelde sessiesleutel via asymmetrische cryptografie.
2. **Data-overdracht:** Met de sessiesleutel wordt alle HTTP-verkeer symmetrisch versleuteld.

SSL/TLS en HTTPS

- **SSL (Secure Sockets Layer):** Oudere standaard, vervangen door TLS.
- **TLS (Transport Layer Security):** Huidige protocol voor veilige communicatie op internet.
- **Certificaten:** Uitgegeven door Certificate Authorities (CA). Een certificaat bevat de publieke sleutel van de server en wordt digitaal ondertekend.

Stappen bij HTTPS-verbinding:

1. **ClientHello:** Browser geeft ondersteunde cipher suites en TLS-versies.
2. **ServerHello + Certificaat:** Server kiest cipher suite en stuurt certificaat.
3. **Verifiëren certificaat:** Browser controleert validiteit en of de certificaatketen klopt.
4. **Key Exchange:** Uitwisseling van sleutels, vaak via ECDHE (Elliptic Curve Diffie-Hellman Ephemeral).
5. **Encrypted Communication:** Al het verkeer is vanaf dat moment versleuteld.

End-to-end encryptie (E2EE)

End-to-end encryptie zorgt dat alleen de communicerende partijen de inhoud kunnen lezen, niet eens de server zelf.

Voorbeeld: WhatsApp gebruikt het Signal-protocol. Berichten worden geëncrypt op de smartphone van de verzender en alleen ontgrendeld op de telefoon van de ontvanger.

Voordelen: Maximale privacy. **Nadelen:** Minder mogelijkheden voor server-side features, zoals zoekfunctionaliteit in berichten of back-up op de server.

Opdracht

Kies een encryptie algoritme en probeer uit te zoeken hoe het werkt.

Beschrijf in een verslag de werking en toepassing van het algoritme.