Applied Cryptography Eindopdracht

Nali Chaushli

500909547

2-02-2024

V2

Pseudo Code

tls_handshake(conn)

- Doel: Deze functie voert het TLS-handshake-protocol uit tussen de server en de client.
- Wat er gebeurt: De server wacht op een "ClientHello" bericht van de client, stuurt een "ServerHello" bericht terug, ontvangt een "ServerKeyExchange" bericht, genereert een EC private key als deze nog niet is gegenereerd, stuurt het servercertificaat naar de client, wacht op een "ClientKeyExchange" bericht, stuurt een "Finished" bericht als de handshake succesvol is.

perform_diffie_hellman_key_exchange(conn)

- Doel: Het doel van deze functie is om de Diffie-Hellman key exchange uit te voeren tussen de server en de client.
- Wat er gebeurt: De server controleert of er al een private key beschikbaar is, genereert de public key van de server, stuurt deze naar de client, ontvangt de public key van de client, en berekent de shared key via de ECDH-methode.

connect()

- Doel: Het doel van deze functie is om een server socket op te zetten en te wachten op een inkomende verbinding van een client.
- Wat er gebeurt: De functie creëert een socket, bindt deze aan het lokale adres en een poort, luistert naar inkomende verbindingen (maximaal één tegelijk) en accepteert de verbinding zodra een client verbinding maakt, waarna het de verbonden socket en het adres van de client retourneert.

send_message(conn, message, shared_key, iv)

- Doel: Deze functie heeft als doel om een bericht te versleutelen en naar de andere partij te sturen.
- Wat er gebeurt: Het bericht wordt voorzien van PKCS7-padding, versleuteld met AES in CBC-modus met het gedeelde sleutel (shared_key) en initialisatievector (iv), en vervolgens verzonden via de socket naar de andere partij.

receive_message(conn, shared_key, iv)

- Doel: Het doel van deze functie is om een versleuteld bericht te ontvangen, ontsleutelen en terug te geven van de andere partij.
- Wat er gebeurt: Het versleutelde bericht wordt ontvangen via de socket, ontsleuteld met AES in CBC-modus met het gedeelde sleutel (shared_key) en initialisatievector (iv), padding wordt verwijderd, en het gedecodeerde bericht wordt teruggegeven.

close_connection(conn, shared_key, iv)

- Doel: Het doel van deze functie is om de verbinding te sluiten en de gedeelde sleutel en initialisatievector te verwijderen.
- Wat er gebeurt: De verbinding wordt gesloten, en de gedeelde sleutel en initialisatievector worden verwijderd.

generate_random_iv()

- Doel: Het doel van deze functie is om een willekeurige initialisatievector (IV) te genereren voor gebruik in symmetrische encryptie.
- Wat er gebeurt: Er wordt een willekeurige 16-byte IV gegenereerd met behulp van os.urandom().

main()

- Doel: Het doel van deze functie is om het gehele communicatieproces tussen de server en de client te coördineren.
- Wat er gebeurt: De server maakt verbinding met de client, voert het TLS-handshake-protocol uit, voert de Diffie-Hellman key exchange uit, genereert een willekeurige IV en stuurt deze naar de client, en wisselt berichten uit totdat 'exit' wordt ingevoerd. Vervolgens wordt de verbinding gesloten en de gedeelde sleutel en IV worden vernietigd.

Waarom EC?

- EC (Elliptische Curve) wordt gebruikt als een alternatief voor RSA vanwege zijn efficiëntie en sterkte in cryptografische toepassingen. Het biedt dus vergelijkbare identificatie- en authenticatiemogelijkheden, maar met kortere sleutellengtes, waardoor het over het algemeen sneller is dan RSA.

Waarom AES?

- AES is een algoritme dat wordt gebruikt om berichten te hashen, en het wordt specifiek gekozen voor dit chatprogramma vanwege zijn effectiviteit en veiligheid in het kader van symmetrische encryptie. Het biedt een basis voor het veilig versleutelen van berichten tussen communicerende partijen (de server en client), wat belangrijk is voor de vertrouwelijkheid van de uitgewisselde informatie in het context van een chatapplicatie.

Waarom CBC-modus voor AES?

- De CBC-modus voegt een initiële vector (IV) toe aan het eerste blok om de variatie in versleutelde blokken te vergroten, waardoor het risico van patroonherkenning wordt verminderd. In het specifieke geval van een chatprogramma, waarbij alle blokken die via de chat worden verzonden moeten worden gedecodeerd, is de CBC-modus bijzonder geschikt. Deze modus zorgt ervoor dat elk blok afhankelijk is van de voorgaande blokken, wat de beveiliging verhoogt en het moeilijker maakt voor potentiële aanvallers om patronen in de gecodeerde communicatie te detecteren.

Risico Analyse

Identiteitsverificatie en Authenticatie:

- **Risico:** Een zwakke identiteitsverificatie en authenticatie kunnen zorgen voor een man in the middle aanval waarbij een aanvaller zich voordoet als de ontvanger of zender.
- **Impact:** Een MitM kan de communicatie onderscheppen, wijzigen of valse informatie sturen.

Sleuteluitwisseling:

- **Risico:** Een onvoldoende beveiliging in sleuteluitwisseling kan zorgen voor het onderscheppen of manipuleren van de gedeelde sleutels.
- **Impact:** Een MitM kan de sleuteluitwisseling verstoren of valse sleutels invoeren, waardoor ze toegang krijgen tot versleutelde berichten.

Integriteit en Authenticiteit:

- **Risico:** een onvoldoende integriteitscontrole kan resulteren in manipulatie van berichten zonder detectie.
- **Impact:** Een MitM kan berichten wijzigen zonder dat de communicerende partijen dit opmerken.

Gebruik van Verouderde Cryptografische Protocollen:

- **Risico:** Het gebruik van verouderde of onveilige cryptografische protocollen kan de veiligheid van de communicatie in gevaar brengen.
- **Impact:** Aanvallers kunnen bekende kwetsbaarheden in verouderde protocollen benutten om de versleuteling te doorbreken, wat leidt tot het compromitteren van vertrouwelijke gegevens.

Gebruik van Willekeurige IV's:

- **Risico:** Onvoldoende willekeurig gegenereerde Initialisatievectoren (IV's) kunnen de beveiliging van de versleuteling verminderen.
- **Impact:** Een voorspelbaar patroon in de IV's kan leiden tot herhaalde versleutelingspatronen, waardoor de veiligheid van de versleutelde communicatie wordt aangetast.