Εργασία Κρυπτογραφίας (Erasmus)

ΛΑΜΠΡΟΥ ΧΡΗΣΤΟΣ 2022052

Εισαγωγή

Στόχος της εργασίας είναι η υλοποίηση ενός πρωτοκόλλου ασφαλούς επικοινωνίας, όπου η Alice μπορεί να στείλει ένα μήνυμα στον Bob χωρίς να έχουν προηγουμένως συμφωνήσει σε κοινό μυστικό κλειδί. Αυτό επιτυγχάνεται με συνδυασμό **asymmetric** και **symmetric cryptography**.

Περιγραφή Πρωτοκόλλου

Στάδιο προετοιμασίας:
Ο Bob δημιουργεί ένα στατικό ζευγάρι κλειδιών (**private key** και **public key**). Το **public key** του είναι διαθέσιμο στην Alice.

Στάδιο αποστολής:

- Η Alice δημιουργεί ένα προσωρινό (**ephemeral**) ζευγάρι κλειδιών.
- Υπολογίζει το κοινό μυστικό με τη μέθοδο **ECDH** χρησιμοποιώντας το **private key** της και το **public key** του Bob.
- Από το κοινό μυστικό παράγει ένα συμμετρικό κλειδί μέσω της συνάρτησης **HKDF** με **SHA-256**.
- Κρυπτογραφεί το μήνυμα με **AES-128-GCM**, που παρέχει εμπιστευτικότητα και ακεραιότητα.
- Στέλνει στον Bob το **ciphertext**, το **nonce** και το **public key** της.

Στάδιο λήψης:

- Ο Bob χρησιμοποιεί το **private key** του και το **public key** της Alice για να υπολογίσει το ίδιο κοινό μυστικό.
- Παράγει το ίδιο συμμετρικό κλειδί μέσω **HKDF**.
- Αποκρυπτογραφεί το μήνυμα και ελέγχει την ακεραιότητα με **AES-GCM**.

Επιλογές Αλγορίθμων

- Καμπύλη **ECDH**: SECP256R1 (128-bit ασφάλεια).
- **KDF**: HKDF με SHA-256, για ασφαλή παραγωγή συμμετρικού κλειδιού.
- Συμμετρική κρυπτογράφηση: **AES-128-GCM**, για εμπιστευτικοτητα και ακεραιότητα.
- Γλώσσα: **Python 3**, με τη βιβλιοθήκη **cryptography** για απλότητα και αξιοπιστία.

Παράδειγμα Υλοποίησης σε Python

```
from cryptography.hazmat.primitives.asymmetric import ec
from cryptography.hazmat.primitives.kdf.hkdf import HKDF
from cryptography.hazmat.primitives import hashes, serialization
from cryptography.hazmat.primitives.ciphers.aead import AESGCM
import os

print("ECDH PROTOCOL - ALICE SENDS MESSAGE TO BOB\n")

# Bob creates a static key pair
bob_private_key = ec.generate_private_key(ec.SECP256R1())
bob_public_key = bob_private_key.public_key()

# Alice creates an ephemeral key pair
alice_private_key = ec.generate_private_key(ec.SECP256R1())
alice_public_key = alice_private_key.public_key()

# Alice creates a shared secret (prA + puB)
shared_secret = alice_private_key.exchange(ec.ECDH(), bob_public_key)

# Derive a 128-bit symmetric key using HKDF
derived_key = HKDF(
    algorithm=hashes.SHA256(),
    length=16,
    salt=None,
    info=b'handshake data'
).derive(shared_secret)

# Alice encrypts the message using AES-GCM
plaintext = b"Hello Bob! This is Alice."
nonce = os.urandom(12)  # 12-byte nonce for AES-GCM
```

```
aesgcm = AESGCM(derived_key)
ciphertext = aesgcm.encrypt(nonce, plaintext, None)

# Alice sends ciphertext, nonce, and her public key
message_to_send = {
    "ciphertext": ciphertext,
    "nonce": nonce,
    "alice_public_bytes": alice_public_key.public_bytes(
        encoding=serialization.Encoding.PEM,
        format=serialization.PublicFormat.SubjectPublicKeyInfo
    )
}

# Bob receives and loads Alice's public key
alice_public_key_received = serialization.load_pem_public_key(
    message_to_send["alice_public_bytes"]
)

# Bob creates the same shared secret (prB + puA)
shared_secret_bob = bob_private_key.exchange(ec.ECDH(),
alice_public_key_received)

# Bob derives the same symmetric key using HKDF
derived_key_bob = HKDF(
    algorithm=hashes.SHA256(),
    length=16,
    salt=None,
    info=b'handshake data'
).derive(shared_secret_bob)

# Bob decrypts the message using AES-GCM
aesgcm_bob = AESGCM(derived_key_bob)
decrypted_message = aesgcm_bob.decrypt(
    message_to_send["nonce"],
    message_to_send["ciphertext"],
    None
)

print(f"Message received by Bob: {decrypted_message.decode()}")
```

```
PS C:\Users\xrist\OneDrive\Υπολογιστής> python ecdh_secure_message.py
ECDH PROTOCOL – ALICE SENDS MESSAGE TO BOB

Message received by Bob: Hello Bob! This is Alice.
```

Συμπέρασμα

Με την παραπάνω υλοποίηση, η Alice μπορεί να στείλει με ασφάλεια ένα μήνυμα στον Bob χωρίς να υπάρχει κοινό μυστικό εκ των προτέρων. Το πρωτόκολλο εξασφαλίζει εμπιστευτικότητα, ακεραιότητα και ασφάλεια, και αποτελεί μια απλοποιημένη εκδοχή μεθόδων που χρησιμοποιούνται σε σύγχρονα πρωτόκολλα όπως το TLS και σε εφαρμογές ανταλλαγής μηνυμάτων.