Short-Lived Certificate in Zero Trust Model

Ayushi Shrivastava, Frimpong Osei, Kenneth Nnadi

University of Oregon, Department of Computer Science

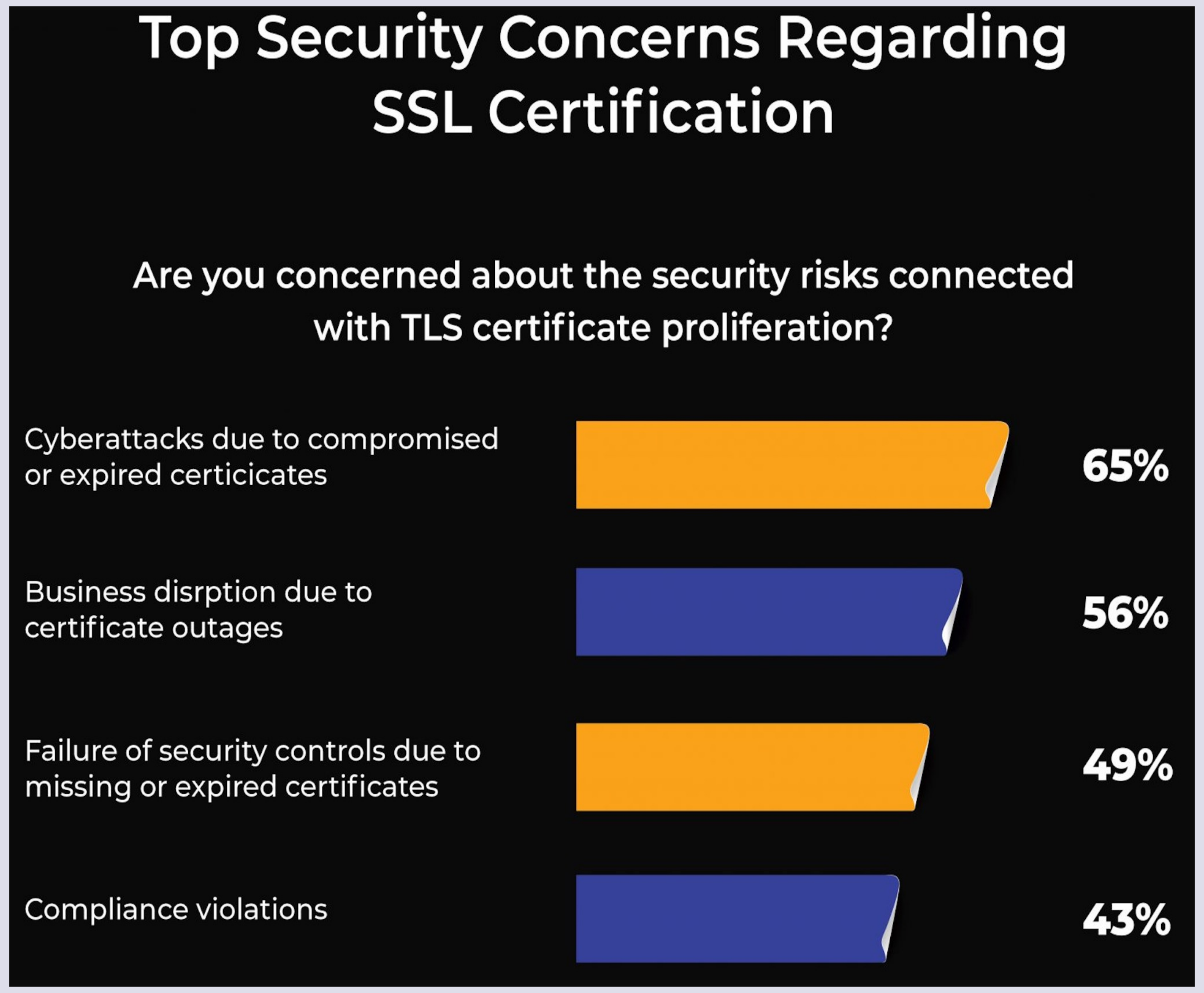
Introduction

This project aims to implement short-lived certificates within the zero-trust security model to enhance security measures and mitigate the vulnerabilities of traditional security models. Traditional security models, such as One Time Password and password-based authentication, have limitations that make them vulnerable to exploitation by threat actors.

Short-lived certificates ensure regular reauthentication, reducing the likelihood of unauthorized access over an extended period within the zero-trust model. Short-lived certificates offer advantages over long-lived certificates by reducing the validity period, limiting the time frame for misuse by attackers and minimizing the impact of compromised credentials.

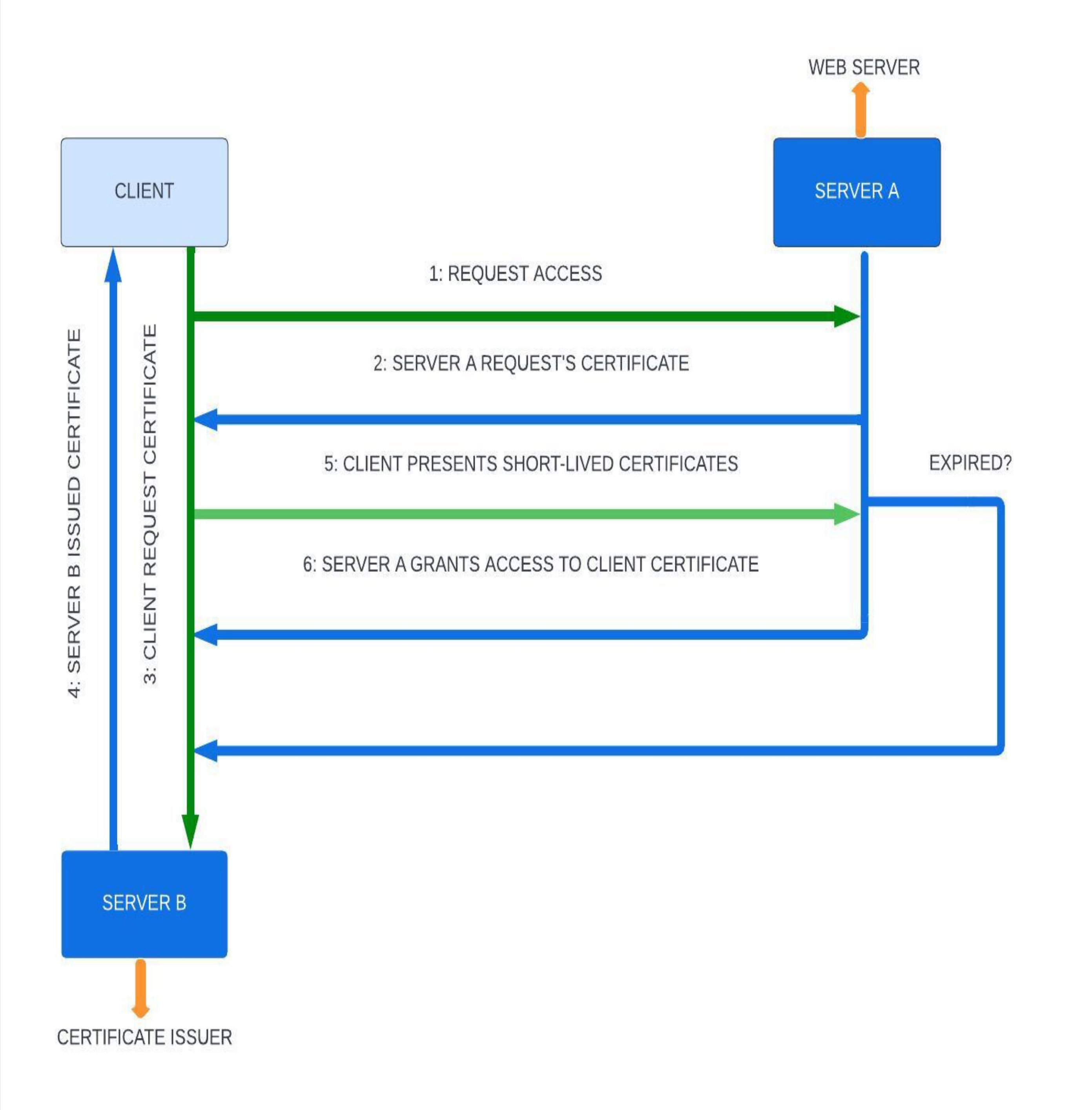
In this poster we focus on the technical aspects of short-lived certificates, including generation, distribution, and revocation processes. By leveraging the advantages of short-lived certificates within the zero-trust model, this project aims to enhance security, mitigate vulnerabilities, and provide robust protection for sensitive information and systems against unauthorized access and cyber-attacks.

Why Short-Lived?

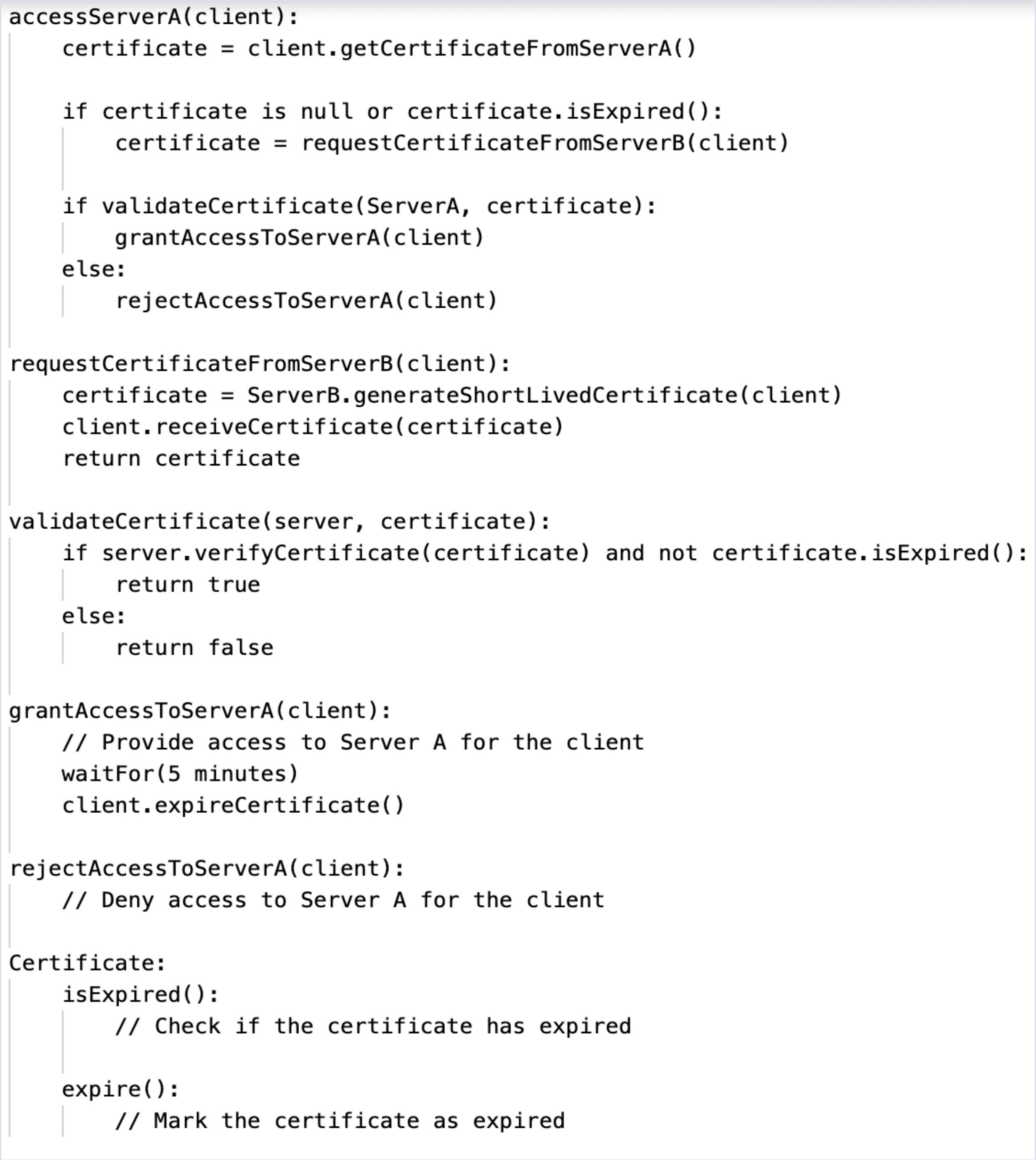


- **Reduced exposure to risk:** reduces risk exposure by limiting misuse of compromised certificates.
- **Faster detection and response**: enables faster detection and response to security incidents.
- **Improved key management:** improves key management through automated renewal and rotation.

Zero Trust Operational Model


WEB SERVER
CLIENT
SERVER A
1: REQUEST ACCESS
2: SERVER A REQUEST'S CERTIFICATE
5: CLIENT PRESENTS SHORT-LIVED CERTIFICATES
EXPIRED?
6: SERVER A GRANTS ACCESS TO CLIENT CERTIFICATE
4: SERVER B ISSUED CERTIFICATE
3: CLIENT REQUEST CERTIFICATE
SERVER B
CERTIFICATE ISSUER

Pseudo Code



```
accessServerA(client):
    certificate = client.getCertificateFromServerA()

    if certificate is null or certificate.isExpired():
        certificate = requestCertificateFromServerB(client)

    if validateCertificate(ServerA, certificate):
        grantAccessToServerA(client)
    else:
        rejectAccessToServerA(client)

requestCertificateFromServerB(client):
    certificate = ServerB.generateShortLivedCertificate(client)
    client.receiveCertificate(certificate)
    return certificate

validateCertificate(server, certificate):
    if server.verifyCertificate(certificate) and not certificate.isExpired():
        return true
    else:
        return false

grantAccessToServerA(client):
    // Provide access to Server A for the client
    waitFor(5 minutes)
    client.expireCertificate()

rejectAccessToServerA(client):
    // Deny access to Server A for the client

Certificate:
    isExpired():
        // Check if the certificate has expired

    expire():
        // Mark the certificate as expired
```

- We spin up two servers [Server A {Web Server}, Server B {Certificate issuer}]. Here, the Client tries to connect to Server A.
- Server A requests certificates. The Client Obtains certificate from Server B and presents to Server A.
- If the certificate is valid, access is granted to the Client. Else Server A requests a new certificate from the Client.

Results

Our model implements a highly secure Zero Trust authentication and access control architecture. It utilizes a Public Key Management Infrastructure (PKMI) to issue short-lived certificates to clients accessing our web server. These certificates have a limited validity period and undergo stringent authentication and expiration checks. Clients presenting valid, non-expired certificates are granted secure access to our protected resources. This approach ensures robust security while simplifying the user experience.

Conclusion

In conclusion, our project successfully integrates short-lived certificates into the zero-trust security model, ensuring continuous verification and minimizing the risk of unauthorized access. Through thorough research and implementation, we have enhanced security measures, mitigated vulnerabilities, and provided a robust approach to protect sensitive information and systems against potential cyber threats. By adopting short-lived certificates and adhering to the "Never Trust, Always Verify" principle, we have established a strong foundation for secure authentication within the zero-trust framework.

References

- Iță, C. R., Constantinescu, R. C., Vlădescu, A., & Alexandrescu, B. (2023, March). Security in remote access, based on zero trust model concepts and SSH authentication with signed certificates. In Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies XI (Vol. 12493, pp. 684-691). SPIE.
- Zhong, Y., Zhou, M., Li, J., Chen, J., Liu, Y., Zhao, Y., & Hu, M. (2021). Distributed blockchain-based authentication and authorization protocol for smart grid. Wireless Communications and Mobile Computing, 2021, 1-15.