Survey on Entity Authentication and Key Establishment

Harmehak Singh **29 March 2024** **Instructor: Prof. Guang Gong Khangura**

Introduction

In today's world, it is important to keep information safe when it is shared. Consider transferring money online or sharing personal information such as passwords. You'll like to make sure that it gets to the right place and is not viewed by the wrong people. This is the place where authentication approaches are used. They operate as digital bouncers which assures that only the intended people have access to the resources.[1] This report is about 10 different entity authentication protocols and how they work. Some of the protocols use special keys for safe communication while others do not require keys. We will look at how each of these approaches works, where they may have flaws and what makes them powerful. It is extremely important to understand these approaches for everyone as it helps them to keep their digital information secure.[2]

1. Kerberos [3]

For conducting the survey about Kerberos, I referenced the research paper titled "An Optimized Kerberos Authentication Protocol." This research paper introduces modifications to the widely used Kerberos authentication protocol. The proposed modifications aim to overcome the weakness of user-chosen passwords which are susceptible to guessing attacks. It does not derive the secret key from the user password. This entity authentication protocol uses a trusted third-party i.e. the Key Distribution Center for secure communication between the clients and the server.

1.1 Key Used: It primarily uses a pre-shared key technique.

1.2 Authentication Tag Generation: When a user attempts to access a service it sends a request for authentication to the Authentication Server (AS). When the AS receives the request it checks the submitted credentials that include a username and password, to ensure their authenticity. If the credentials are properly confirmed then the AS creates a Ticket Granting Ticket (TGT) for the user. The TGT contains critical information like the user's identification, a session key that is used for future connections with the Ticket Granting Server (TGS) and an expiration period. To maintain security, the TGT has been encrypted with the TGS secret key. The AS then sends the encrypted TGT back to the user for future use in accessing other services.[4]

1.3 Authentication Tag Validation: When a user wants to access to a service they initiate a request to the TGS for a service ticket using their TGT and the service identifier. The TGS decrypts the TGT which confirms the user's identity and verifies the requested service. Once the validation is successful then the TGS constructs a service ticket which contains the user's identity, a session key and service-specific information. This service ticket is encrypted using the secret key of the target service and is dispatched back to the user. After receiving the user presents the service ticket to the designated service. The service decrypts the service ticket using its secret key and cross-validates the user's identity along with the session key. If it is validated correctly then the service grants access to the user. During this process the authentication tag undergoes validation at multiple points: The AS verifies user credentials during the initial authentication, the TGS confirms the legitimacy of the TGT and generates service tickets, and finally, the service authenticates service tickets for access.[4]

1.4 Mutual or One-Way Authentication: Kerberos supports mutual authentication.

1.5 **Scheme secure:** It includes improvements to reduce vulnerabilities related to password guessing attacks. By splitting up the secret key from user passwords and using a profile-based strategy it significantly reduces the possibility of direct password breach. But the protocol's security is dependent on proper secret key storage, strong cryptographic algorithms and reliable profile management. In this formal security proofs are not adequate and continuous attention is required to maintain a safe Kerberos system.[1]

1.6 Security proved in literature: Yes, its security is proved in literature.

1.7 Possible Vulnerabilities [5]

- **Secret Key Storage**: The security of the protocol is dependent on the secure storage of secret keys within the Kerberos infrastructure. Any breach of these keys might result in unauthorized access.
- **Profile Management**: Proper handling of user profiles is important. Issues in profile generation or storage can compromise security.

1.8 Possible Countermeasures [5]

- **Regular Audits**: Regular audits of the Kerberos infrastructure like profile management and key storage can be conducted.
- **Key Isolation**: Isolate the secret keys within the Key Distribution Centre.

1.9 Application Scenario, Possible Vulnerabilities and Countermeasures

1.9.1 Application Scenario: Enterprise Networks - Kerberos is widely used for secure authentication within corporate networks. Employees can easily access resources such as files, printers, and databases without sending passwords across the network.

1.9.2 Vulnerabilities [5]

- **Password Guessing Attacks:** Attackers can guess passwords or can crack the passwords to get access to secret keys.
- **Insider Threats:** Malicious insiders with access to the Kerberos database may misuse secret keys.

1.9.3 Countermeasures [5]

- Enforce strong password regulations to avoid weak passwords.
- Use Hardware Security Modules (HSMs) to provide safe storage.
- Combine Kerberos with Multi Factor Authentication to increase security.

1.10 An attack which can impersonate both entities and the consequence of the attack using a real-world application scenario: Consider a scenario in which an attacker attempts to obtain unauthorized access to sensitive data kept on a file server. The attacker obtains the user's TGT via network traffic. The TGT contains encrypted service tickets for a variety of services. The attacker concentrates on a service ticket intended for the file server. Using offline brute-force techniques, the attacker breaks the service ticket's encryption, which is based on the user's password. Once decrypted, the attacker has a valid service ticket for the file server. The attacker submits the legal service ticket to the file server while posing as the legitimate user. The file server verifies the ticket, presuming it is for an authorized user. As a result, the attacker obtains unauthorized access to sensitive files, which may compromise confidential information. It is important to note that this attack stays secret since it does not require direct connection with the Key Distribution Center (KDC) or the user. Real-world implications include business espionage, intellectual property theft, healthcare data breaches, etc.

1.11 Security Analysis: Kerberos helps to ensure that only authorized users may access a network's applications and services. It accomplishes this with the use of secret codes to determine who is logging in. However attackers may attempt to decode these codes using different techniques for getting access.[1] To prevent this one needs to employ extremely difficult-to-guess codes and ensure that all computer clocks are properly set.[3]

2.Transport Layer Security Protocol [6]

For conducting the survey about TLS Protocol, I referenced the standard titled “The Transport Layer Security (TLS) Protocol Version 1.3.” In this standard, it describes TLS 1.3, which is a secure communication protocol. TLS 1.3 dramatically increases security, speed, and privacy compared to its predecessors. Key improvements include the elimination of old cryptographic techniques, a simplified handshake procedure for quicker connections, and necessary forward secrecy to safeguard previous messages even when future keys are stolen. The paper describes TLS 1.3's two primary components: the handshake protocol, which creates cryptographic parameters and authenticates participants, and the record protocol, which encrypts communication data based on these parameters.

2.1 Key Used: It uses a combination of public-key and symmetric pre-shared key mechanisms.

2.2 Authentication Tag Generation: The authentication tag is created during the encryption process using the Authenticated Encryption with Associated Data (AEAD) technique. The AEAD algorithm merges the secret key,

plaintext and any related data to generate the ciphertext and authentication tag. This step guarantees ensures that the tag is distinct to the encrypted material.

2.3 Authentication Tag Validation: After obtaining the data that was encrypted and the authentication tag, the intended recipient decrypts the data using the same AEAD algorithm and secret key and generates an authentication tag from the obtained ciphertext and related data. The recipient then matches the independently created authentication tag to the received tag. If both tags match, it means the information was not altered and is valid.

2.4 Mutual or One-Way Authentication: TLS 1.3 provides both one-way and mutual entity authentication. It depends upon the usage of certificates and pre-shared keys.

2.5 Scheme Secure: TLS 1.3 methods are regarded secure because they use contemporary cryptographic standards and techniques that improve security and efficiency.

2.6 Security proved in literature: Yes, the security of TLS 1.3 has been proven in the literature. A major effort introduces a novel security proof for TLS 1.3 in the random oracle paradigm. This proof strongly lowers the security of TLS 1.3 to the multi-user security of its components, providing realistic security assurances for real-world deployments.

2.7 Possible Vulnerabilities

- **0-RTT Replay Attacks**: TLS 1.3's 0-RTT option trades off security for speed, leaving it susceptible to replay attacks.
- **Misconfigured Certificates**: Certificates that are not properly set or checked might introduce problems.

2.8 Possible Countermeasures

- Use it carefully and only with non-sensitive data.
- Maintain correct certificate management processes.

2.9 Application Scenario, Possible Vulnerabilities and Countermeasures: TLS 1.3's application scenario primarily includes secure online surfing and any communication over the internet needing secrecy and integrity. Misconfigurations that undermine security are one example of a potential vulnerability. Countermeasures include tight adherence to security settings and frequent upgrades to ensure best practices, such as using strong, latest cryptographic algorithms and effective certificate management.[7]

2.10 An attack which can impersonate both entities and the consequence of the attack using a real-world application scenario: In TLS 1.3, vulnerabilities are frequently caused by implementation and configuration issues rather than the protocol itself. For example maintaining outdated TLS protocols for compatibility might leave systems vulnerable to downgrade attacks. In this an attacker forces the utilization of weaker encryption. Thus impersonating both sides in a transaction. In the context of online banking such an assault might threaten safe communications which can lead to the theft of sensitive data such as login passwords and financial details which can further lead to unauthorized access, monetary loss and reputational harm to the institution and its clients.[7]

3. Secure Shell Authentication Protocol [8]

For conducting the survey about Secure Shell (SSH) Authentication Protocol, I referenced the standard titled “The Secure Shell (SSH) Authentication Protocol.” This standard describes how to authenticate a user to an SSH server. It defines techniques such as passwords, public keys and host-based authentication. The document guarantees safe server-client connection by preventing unwanted access and ensuring the integrity and confidentiality of the data transferred. It is an essential component of the SSH protocol, allowing for a range of secure remote activities over unprotected networks.

3.1 Key Used: It uses public-key approach.

3.2 Authentication Tag Generation: The SSH protocol generates the Message Authentication Code (MAC) during data transfer. It provides authenticity as well as integrity. The MAC consists of data, a sequence number and a shared secret key created during the first handshake. This key is part of a set that is only used during the session to secure data transmission.

3.3 Authentication Tag Validation: The SSH server gets encrypted data along with the MAC. Then it checks for integrity and authenticity. The server decrypts the data with the shared secret key and then calculates the MAC using the original data and same sequence number. If the calculated MAC corresponds to the received one then it verifies data authenticity and integrity. Thus protecting it against manipulation or impersonation.

3.4 Mutual or One-Way Authentication: It provides Mutual Entity Authentication. It occurs when establishing a secure connection between a client and a server.

3.5 Scheme Secure: When properly built and deployed, the authentication systems described in the article are deemed secure. SSH ensures the secrecy, integrity, and authenticity of communications over insecure networks by utilizing well-established cryptographic techniques and concepts.[9]

3.6 Security proved in literature: Yes, the security of SSH in the text has been thoroughly examined.

3.7 Possible Vulnerabilities [9]

- Improper Configuration Settings: Incorrect configurations may result in unforeseen security flaws.
- Compromised Keys: Forged SSH keys might result in unwanted access.

3.8 Possible Countermeasures [9]

- Configure SSH according to recommended practices such as eliminating unnecessary capabilities and restricting access.
- Implement key management techniques like frequent key rotation and secure private key storage.

3.9 Application Scenario, Possible Vulnerabilities and Countermeasures: Secure remote access allows workers to safely access business resources from remote locations. To protect data, it is necessary to incorporate strong authentication techniques like multi-factor authentication along with SSH. Access control mechanisms and network security solutions like as firewalls and VPNs play a critical role in mitigating vulnerabilities such as weak authentication, brute force attacks, etc. Organizations may keep their remote access environment safe by implementing strong password rules and routinely upgrading remote equipment.[9]

3.10 An attack which can impersonate both entities and the consequence of the attack using a real-world application scenario: In terms of safe remote access inadequate authentication techniques might be regarded a severe weakness. Credential theft and password-based attacks might take advantage of this vulnerability and impersonate both entities. A MITM attack might employ authentication flaws to impersonate both the remote user and the server. For instance, compromising the initially established SSH connection enables the attacker to obtain user credentials. They can then impersonate the user to create a connection, thereby harming confidentiality of information and integrity.[10]

4. Internet Key Exchange Protocol [11]

For conducting the survey about Internet Key Exchange Protocol, I referenced the paper titled “Security Analysis of IKE’s Signature-Based Key-Exchange Protocol.” This uses Key establishment approach. This work presents a security study of the Diffie-Hellman key-exchange protocol validated with digital signatures.

4.1 Key Used: The study explores the security of IKE's signature-based mode that uses public-key approach. It examines the variation of the Station-to-Station protocol used in Photuris wherein the Diffie-Hellman (DH) key is signed.

4.2 Authentication Tag Generation: When any associated data (AAD) exists, it is used to compute the Message Authentication Code (MAC) using a hash function. This hash function is commonly GHASH. The ciphertext is then created. The authentication tag is determined by mixing it with the AAD. This authentication tag functions as a cryptographic checksum, confirming the integrity of the data and related information throughout transmission and storage.

4.3 Authentication Tag Validation: When the encrypted data and authentication tag are received, the person who received them decrypts the ciphertext with the shared secret key created during the Diffie-Hellman key exchange. The receiver then recalculates the authentication tag utilizing the original data and same sequence number. The recalculated tag is compared to the provided authentication tag. If the recalculated tag equals the received one, it validates the data's integrity and authenticity, indicating that it had not been altered throughout transmission and came from the intended sender.

4.4 Mutual or One-Way Authentication: It provides Mutual Authentication.

4.5 Scheme Secure: The paper discusses the security of the IKE protocol's signature-based mode. Initially it had some flaws but they were solved using a superior mechanism known as "sign-and-mac." This new solution significantly improves the overall security of IKE's signature-based authentication.

4.6 Security proved in literature: The security of these schemes is dependent on good implementation, configuration options and following the best practices. While they have been scrutinized and improved but ensuring secure deployments remains critical.

4.7 Possible Vulnerabilities

- **Insecure Initial Variant:** The first version of IKE's signature-based mode that was based on a variant of the STS protocol was discovered to be insecure. This vulnerability has the potential to allow unwanted access or compromise communication.
- **Identity Confidentiality:** The practical environment in which peer identities are not always known at the start of the protocol creates dangers. Confidentiality may be compromised if an attacker learns identities during the process.

4.8 Possible Countermeasures

- **Sign-and-MAC method:** This approach guarantees that the Diffie-Hellman key is signed and strengthens authentication.
- **Formal study:** Extensive thorough examination of IKE and its supporting SIGMA protocols is required to identify potential vulnerabilities and assure its security.

4.9 Application Scenario, Possible Vulnerabilities and Countermeasures: IKE is critical in the establishment of Virtual Private Networks (VPNs) because it creates secure communication channels among remote sites, users and a central network. Vulnerabilities like poor authentication and key reuse persist. Strong authentication and implementation of Perfect Forward Secrecy are critical. Mutual authentication can help against MITM attacks. [12]

4.10 An attack which can impersonate both entities and the consequence of the attack using a real-world application scenario: In this attack scenario the attacker creates a faked domain that closely mimics an actual organization's domain. The attacker creates emails, webpages, etc utilizing this fake domain to impersonate both parties. This type of attack has resulted in illegal account access, data breaches and financial losses. [12]

4.11 User Data Protection in Internet Key Exchange Protocol

IKE protocol establishes a Security Association (SA) to protect data. It integrates authentication, encryption and key management mechanisms to ensure access to authorized parties. IKE uses X.509 certificates for authentication and a Diffie-Hellman key exchange to generate a shared session secret to derive cryptographic keys to encode and decode data.[13]

4.12 Key Updates

The keys are updates in two phases. In the initial phase a SA is established that is used to set up multiple IPsec SAs in the second phase. In first phase the Diffie-Hellman algorithm creates a shared encryption key. The second phase negotiates security associations and services with devices selecting the protocol and algorithm after configuring the ISAKMP tunnel. Keys are updated whenever needed. The ExtendedKeyUpdate message follows the Finished message.[12][13]

5. RADIUS Protocol [14]

For conducting the survey about RADIUS, I referenced the research paper titled “Efficient and Secure Key Distribution Protocol for Wireless Sensor Networks.”

5.1 Key Used: It uses a pre-shared key approach.

5.2 Authentication Tag Generation: When a user attempts to connect with a Network Access Server (NAS), the NAS requests a username and password. After obtaining the login details, the RADIUS client transfers it to the RADIUS server in an Access-Request packet along with the login information hidden using the RSA Message Digest Algorithm MD5. The RADIUS server then verifies the authenticity of the user's information through

different methods of authentication like the Password Authentication Protocol, Challenge Handshake Authentication Protocol, etc. [15]

5.3 Authentication Tag Validation: After receiving the request, the RADIUS server will either accept or refuse it. In the case of approval, an Access-Accept packet is sent to the RADIUS client, providing the user network access. If the request is considered illegitimate then the RADIUS server sends an Access-Reject packet to the RADIUS client, thus blocking access to the network.[15]

5.4 Mutual or One-Way Authentication: It uses Mutual Authentication.[14]

5.5 Scheme Secure: RADIUS provides an effective network access control architecture. Its security is heavily reliant on the privacy and strength of the common secret.

5.6 Security proved in literature: Yes, the RADIUS protocol's security has been thoroughly documented in academic papers.

5.7 Possible Vulnerabilities

- **Replay Attacks**: Attackers can collect and replay RADIUS communications to get unauthorized access.
- **Brute-Force and Dictionary Attacks:** Attackers can use brute-force and dictionary attacks to figure out the secret that is shared or user passwords.

5.8 Possible Countermeasures

- Prevent replay attacks by using message authentication codes (MACs) and timestamps.
- Enforce strong password restrictions, set up account lockouts, and utilize safe cryptographic hashing techniques.

5.9 Application Scenario, Possible Vulnerabilities and Countermeasures: RADIUS is widely used in many networking situations to provide centralized authentication, authorization, and accounting (AAA) services. It provides network access control for a variety of network services such as wireless networks, Virtual Private Networks, etc. Potential vulnerabilities include shared secret compromise, replay attacks, and eavesdropping. Countermeasures that businesses should use are that they should keep strong shared secrets, robust encryption, monitor traffic regularly, etc. [17]

5.10 An attack which can impersonate both entities and the consequence of the attack using a real-world application scenario: In a business Wi-Fi network with RADIUS authentication, the attacker can use Man-in-the-Middle attack. An attacker may create a rogue access point to spoof the genuine network and steal employee credentials while simultaneously masquerading the RADIUS server. With stolen credentials, the attacker has illegal access to key corporate resources that can lead to data breaches, network penetration and more credential-based crimes.[18]

6. Extensible Authentication Protocol [19]

For conducting the survey about Extensible Authentication Protocol (EAP), I referenced the paper titled “The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method.”

6.1 Key Used: It uses pre-shared key approach.

6.2 Authentication Tag Generation: EAP-PSK generates the authentication tag or the Message Authentication Code by utilizing the Cipher-based Message Authentication Code mode of operation. This maintains message's authenticity and integrity by computing the MAC of important message components like the pre-shared key, peer and server identities, random numbers, etc.

6.3 Authentication Tag Validation: The recipient of a message who might be the server or the client, recalculates the MAC using the exact same input as the sender and the same Pre-Shared Key to verify its validity. If the calculated MAC corresponds to the MAC transmitted along the message then the communication is deemed legitimate. In this way the sender is verified. This validation step verifies that the message was not tampered with during transit and was transmitted by an entity with the right PSK.

6.4 Mutual or One-Way Authentication: It uses Mutual Authentication.

6.5 Scheme Secure: The security of EAP-PSK is mainly reliant on maintaining the secret key between the user's device and the server. If the key remains secret then the EAP-PSK provides a secure approach to verify IDs. It relies on the security of the technologies it utilizes.

6.6 Security proved in literature: Yes, the security of this protocol is proved in literature.

6.7 Possible Vulnerabilities

- **Lack of Perfect Forward Secrecy:** EAP-PSK does not provide perfect forward secrecy. If a PSK is hacked then not only the future interaction compromised but also the previous conversations encrypted with the stolen PSK may be deciphered if intercepted by an attacker.
- **Limited Protection Against Man-In-The-Middle Attacks**: EAP-PSK does not have an extra encryption layer making it vulnerable to Man-in-the Middle attacks.

6.8 Possible Countermeasures

- Use extra layers of security like TLS.
- Set up secure techniques for storing and transferring Pre Shared Key.

6.9 Application Scenario, Possible Vulnerabilities and Countermeasures: A scenario that can use EAP is when an online banking application uses EAP for authentication. Issues like phishing attacks can be a great concern in this case. Attackers may establish fake banking portals and send misleading emails to lure customers into disclosing their login information. Implementing multi-factor authentication is one way to mitigate security risks. Financial institutions might also use sophisticated encryption methods for information in transit and at rest to prevent attackers from deciphering valuable information.[18]

6.10 An attack which can impersonate both entities and the consequence of the attack using a real-world application scenario: The attack which can impersonate both entities is replay attack. In this type of attack an attacker intercepts communications between a device and a server during an encrypted session and subsequently sends them again pretending to be the original user. In a business Wi-Fi network that uses EAP for secure access, there an attacker can intercept the EAP authentication exchange among a worker's device and a network's authentication server. Even without decrypting the communication, the attacker can exploit the intercepted information to obtain unauthorized network access as the legitimate user. This can lead to potential data breaches and exposure of sensitive corporate information.[20]

7. Two-Factor Authentication Protocol [21]

For conducting the survey about Two-Factor Authentication Protocol, I referenced the paper titled “A Robust and Effective Two-Factor Authentication (2FA) Protocol Based on ECC for Mobile Computing.” It employs Elliptic Curve Cryptography (ECC) to improve security in mobile computing settings.

7.1 Key Used: It uses a public-key cryptography technique.

7.2 Authentication Tag Generation: In this the authentication tag is generated by creating either a digital signature or an encrypted token that contains the sender's identity as well as the message's integrity. This technique commonly employs the sender's private key to encrypt a message hash. This results in a tag that can only be confirmed by the associated public key.

7.3 Authentication Tag Validation: In the process of validating an authentication tag, the recipient uses the sender’s public key to decrypt or authenticate the tag. This confirms the integrity of the message and verifies the sender’s identity. If the decryption or verification is successful then it means that the message is the same and was sent by the owner of the associated private key. This procedure is dependent on the security features of ECC that has the infeasibility of inferring the private key from the public key.

7.4 Mutual or One-Way Authentication: The Two-Factor Authentication Protocol which is based on ECC uses Mutual Authentication.

7.5 Scheme security and Literature: The paper demonstrates the protocol's resilience and efficacy via heuristic analysis and security verification with the ProVerif tool. It is an automated validator for cryptographic protocols. The performance comparison of six schemes reveals superiority and similarity in efficiency and security. This shows a strong balance between the key characteristics.

7.6 Possible Vulnerabilities

- **Side-Channel Attacks**: These attacks make use of information acquired from the system's physical configuration, such as time data, energy consumption, and electromagnetic emissions.

- **Scalability Concerns**: As the user base grows, effectively and securely managing public keys can become a substantial difficulty.

7.7 Possible Countermeasures

- Implement secure key generation and storage mechanisms.
- Employ hardware security modules for critical cryptographic operations.

7.8 Application Scenario, Possible Vulnerabilities and Countermeasures: 2FA is used in online banking where in addition to the username and password being provided an additional second verification factor like a code through SMS or an authenticator app is used to access their accounts. Vulnerabilities such as SMS interception via SIM swapping and phishing attempts that trick users into inputting their 2FA codes on fake sites endanger the effectiveness of 2FA. Countermeasures like use of authenticator app which can generate codes offline should be used.

7.9 An attack which can impersonate both entities and the consequence of the attack using a real-world application scenario: The attack that can impersonate the entities is SIM Swap Scam. This attack is specifically designed to target the mobile phone number linked to a user's account, which is frequently used to receive 2FA codes through SMS. Consider a scenario in which a user uses 2FA for logging into social media accounts. In this type of attack the attacker can trick a phone service provider to transfer a person's cell phone number to their SIM card. This enables the attacker to receive SMS messages and phone calls including two-factor authentication codes. Using these codes the attacker can bypass 2FA and get access to the victim's social media accounts. The attacker can abuse sensitive information as well as shut out the real user.[22]

8. Challenge-Handshake Authentication Protocol [23]

For conducting the survey about Challenge-Handshake Authentication Protocol (CHAP), I referenced the paper titled "PPP Challenge Handshake Authentication Protocol (CHAP)." This paper discusses about periodic identity verification using a three-way handshake.

8.1 Key Used: It uses a pre-shared key approach for authentication.

8.2 Authentication Tag Generation: The Authentication Tag is generated when you apply a one-way hash function to a combination of the authenticator's challenge, a secret known by both the authenticator and the peer and any additional data. The calculated hash value is subsequently returned to the authenticator as a response.

8.3 Authentication Tag Validation: For validation, the authenticator after having issued the challenge and being aware of the shared secret carries out an identical hash function. If the calculated hash aligns with the peer's response then the authentication is deemed successful.

8.4 Mutual or One-Way Authentication: CHAP makes use of Mutual Authentication.

8.5 Scheme security and Literature: The security of CHAP is heavily reliant on the capability of the hash function used and the confidentiality of the shared secret. CHAP is considered safe against passive attacks but it is susceptible to active attacks under certain situations. It has been proved in literature.

8.6 Possible Vulnerabilities: CHAP is typically safe for its intended use. But it is vulnerable to replay attacks if an intruder records the challenge and response. Another weakness is the possibility for dictionary attacks on weak secrets.

8.7 Possible Countermeasures: Using strategies like time-bound sessions that allow challenges to expire quickly can be used to counter replay attacks.

8.8 Application Scenario, Possible Vulnerabilities and Countermeasures: In the context of a VPN connection CHAP facilitates authentication between the client and the server. It is susceptible to replay attacks in which the intercepted data is maliciously reused and brute-force attacks targeting weak secrets. Countermeasures such as implementing encryption for data transmission, using intricate secrets that are regularly updated and adding extra authentication layers can be employed to enhance security.[24]

8.9 An attack which can impersonate both entities and the consequence of the attack using a real-world application scenario: Man-in-the-Middle attacks can impersonate both participants in a communication session by intercepting and even altering the challenge and answer messages transmitted between them. This can lead to

illegal access to network resources and sensitive data, specifically in environments that rely on CHAP for authentication, like VPNs or remote server access.

9. OpenID Connect Protocol [25]

For conducting the survey about OpenID Connect Protocol, I referenced the paper titled "Detecting Risky Authentication Using the OpenID Connect Token Exchange Time."

9.1 Key Used: The OpenID Connect (OIDC) protocol mainly uses public keys to secure conversations.

9.2 Authentication Tag Generation and Validation: In OpenID Connect (OIDC), an authentication tag is included in a JSON Web Token (JWT). JWTs are a safe mechanism for representing claims among two entities. This authentication tag, an encrypted digest of the message, ensures the token's integrity and authenticity. It is constructed using cryptographic techniques such as HMAC for symmetric keys and RSA for asymmetric keys that are then applied to the token's content. The validation method entails recalculating the digest or the signature using the public key and comparing it to the tag on the received token to guarantee that it has not been changed.

9.3 Mutual or One-Way Authentication: OpenID Connect mainly supports one-way authentication.

9.4 Scheme security and Literature: The paper aims to improve OIDC security by proposing a unique method for identifying hazardous authentications. It suggests using the time necessary for OIDC token exchanges to provide an innovative function for risk-based authentication. This technique intends to identify probable unwanted access attempts by evaluating token exchange length aberrations adding to the literature on safeguarding digital identities and access management in a zero-trust system.

9.5 Possible Vulnerabilities and Countermeasures: The article focuses on weaknesses in the timing of OIDC token exchanges that might signal illegal access attempts. It implies that examining token exchange periods might reveal dangerous authentications which are caused by tunneled connections. To address these vulnerabilities, the paper recommends using the time frame of these exchanges as a unique feature for risk-based authentication, which improves security without requiring user involvement. This technique is consistent with the zero-trust philosophy stressing upon the ongoing verification of identities.

9.6 Application Scenario, Possible Vulnerabilities and Countermeasures: In an e-commerce setting, customers log in to numerous shopping platforms making use of OIDC. Vulnerability like account takeover via credential stuffing is a serious issue. In this the attackers use compromised username-password pairs across several sites. Countermeasures include using CAPTCHA to prevent automated attempts to login, monitoring login attempts for odd trends and requiring the use of unique, strong passwords together with multi-factor authentication to dramatically improve user account security.

9.7 An attack which can impersonate both entities and the consequence of the attack using a real-world application scenario: In a healthcare situation with OIDC for patient portal access, an attacker might exploit the "Token Leakage" vulnerability. It occurs when the access token is mistakenly exposed to external parties via referrer headers and log files. This vulnerability can give unauthorized access to critical patient information. This results in privacy breaches and identity theft. Countermeasures involve providing that login tokens are securely sent rather than tracked as well as unintentionally disclosed via web referrers.

10. Host Identity Protocol [26]

For conducting the survey about Host Identity Protocol (HIP), I referenced the paper titled "Host Identity Protocol Architecture."

10.1 Key Used: It focuses particularly on the usage of public keys as a core component of the HIP design.

10.2 Authentication Tag Generation: In the Host Identity Protocol (HIP), authentication tags are generated by using cryptographic methods such as digital signatures or HMACs to secure messages. A sender uses their private key to sign a message or its hash, and then attaches the resultant signature as an authentication tag. When HMACs are utilized the sender estimates the HMAC by hashing the message using a shared secret accessible to both the sender and the receiver and then attach the HMAC as a tag to validate the message's authenticity and origin.

10.3 Authentication Tag Validation: When HIP receives a message, it validates the accompanying authentication tag to confirm its validity and integrity. For digital signatures the receiver uses the public key of the sender to verify the signature with the message. This guarantees that the message was transmitted by the holder

of the associated private key and remains unchanged. In the scenario of HMACs, the receiver recalculates the HMAC using the secret that was shared and compares it to the received tag to ensure that the message is valid and unmodified. This validation phase is critical to maintaining safe and trustworthy communication inside the HIP framework.

10.4 Mutual or One-Way Authentication: It supports mutual authentication.

supports mutual authentication.

10.5 Scheme security and Literature: It is designed in such a way that make communications secure. It uses cryptographic techniques to make sure that the entities are mutually authenticated. It uses public-key encryption and digital signatures. There are few formal demonstrations of security for HIP. It is deemed secure since it employs trusted procedures.

10.6 Possible Vulnerabilities and Countermeasures:

- HIP aims to minimize DoS attacks by including puzzles during the first handshake procedure that require computational effort from the requester and therefore reduce the possibility of overloading the responder with requests.
- Theoretical breakthroughs in quantum computing have the potential to undermine HIP's cryptographic underpinnings, including the RSA and ECC algorithms. Countermeasures include implementing quantum-resistant cryptographic algorithms and key exchange methods when they become standardized and accessible.

10.7 Application Scenario, Possible Vulnerabilities and Countermeasures: Healthcare Data sharing: HIP may be used to secure patient data sharing between hospitals, clinics and insurance companies. HIP's capability to deliver safe, verified and encrypted communication channels protects the privacy and confidentiality of private medical data as it travels across networks. This scenario takes advantage of HIP's capabilities in identity management and data security to safeguard against illegal access and data breaches. Vulnerabilities in the Host Identity Protocol architecture include endpoint mobility constraints, scalability issues in large networks and compliance with severe rules. Countermeasures are an effective way to reduce these hazards. Improving HIP mobility management with dynamic key exchanges provides safe network transitions that is critical in industries. Addressing scalability using distributed technologies such as cloud computing and blockchain allows for better key and identity management over large networks. [27]

10.8 An attack which can impersonate both entities and the consequence of the attack using a real-world application scenario: The application scenario will be smart home settings, where HIP may be used to encrypt communications between various IoT devices and the central security control system. This guarantees that critical data such as video feeds and security alerts are securely delivered and prevent unwanted access. A Sybil attack in which an attacker generates many fake identities to achieve unfair control over the network is one potential vulnerability in this situation. In this scenario an attacker may try to introduce several malicious devices into the network with each posing as a valid security component. These devices might then be used to gather and modify critical information, impair the normal functioning of the security arrangement.[27]

3 Attacks on AKA and their Countermeasures are [28]

i) **IMSI Catching:** IMSI catchers which are also called as Stingrays are the devices that mimic real base stations. These gadgets can deceive mobile devices into connecting with them and disclosing their IMSI numbers. IMSI numbers enable attackers to follow users and launch more focused attacks.

Countermeasure: To prevent this type of attack mobile networks can use IMSI encryption and temporary identities which alter regularly. This would prohibit IMSI catchers from collecting permanent IMSI numbers for mobile customers.

ii) **Sequence Number Prediction:** AKA relies on sequence numbers in its authentication method. If an attacker can anticipate the sequence numbers then the attacker would be able to evade authentication checks and get unauthorized network access.

Countermeasure: Networks should use unexpected sequence number creation processes. This could include applying safe random number generators and using more unexpected factors in the sequence number computation.

iii) **Downgrade Attacks:** In this attack, the attackers compel a device to connect to a less secure network standard such as switching from 5G to 4G or 3G. Older standards may lack the same security measures as current ones and therefore leave devices more vulnerable to eavesdropping and interception.

Countermeasure: Implement tight network selection criteria for devices that allow them to refuse connections to networks which do not satisfy a minimal security level. Networks can use mutual authentication to verify that both the network and the device are using the most secure communication standard available.

EIA1 Forgery in the Context of EEA1 and EIA1: In 4G-LTE/5G networks, the EEA1 and EIA1 algorithms are crucial for protecting communications. The former provides encryption and the latter maintains the data integrity. EIA1 is vulnerable to forgery attacks either if the keys or algorithm integrity is breached. These weaknesses can result from the method's susceptibility to collision attacks. In these attacks two different inputs produce the same output. It can also arise from poor key management methods that allow attackers to guess the key. These flaws can allow attackers to create authentic-looking communications with a valid MAC. This leads to compromising the integrity protection mechanism intended to prevent data tampering.

Attack 1: Obtaining Two MACs with the Same IVs

Attack Overview: This attack targets instances in which an intruder is able to intercept and influence the production of MACs making use of the same IV. The security of many cryptographic systems is dependent on the distinctiveness of IVs. If an attacker is able to exploit a circumstance in which the same IV is used with various messages or keys then they can disclose data regarding the encryption key.

Practicality: The effectiveness of this attack is determined on the system's implementation defects and operational deficiencies. As an illustration, if a system fails to either sufficiently randomize IVs or ensures their distinctiveness for each execution then an attacker might potentially analyze and manipulate traffic to discover as well as create scenarios in which the same IV is reused.

Success Probability: It is heavily dependent on the precise implementation specifics and the attacker's ability to affect and view the IV generating process. A badly constructed system in which IVs are reused reliably might result in a considerably high success rate. In well-constructed systems that maintain IV uniqueness and randomization then this type of attack is extremely impractical.

Attack 2: Forgery Attack Using One MAC

Attack Overview: This attack attempts to generate a legitimate MAC for a falsified message by exploiting flaws in the MAC creation process. If an attacker obtains a legitimate MAC-IV pair for a specific message and the MAC creation process is ineffective then the attacker can attempt to change the contents of the message while still creating a valid MAC. This will result in a successful forgery.

Practicality: A successful forgery attack necessitates a thorough understanding of the MAC algorithm's flaws and the presence of specific cryptographic pieces such as a valid MAC-IV pair. The attack's viability is heavily determined by the power that lies in the MAC technique and the safety of its key management mechanism.

Success Probability: If the attacker has uncovered a substantial flaw in MAC technique that enables foreseeable manipulations or if the management of keys system has been hacked then the likelihood of success may be high. Nevertheless, with strong cryptographic protocols and safe key management techniques the chances of successfully conducting a forgery attack with only one MAC are slim.

Analysis of public-key cryptography required when 5G is designed to support IoT?[29]

An analysis of why public-key cryptography is essential for IoT in 5G:

- **Scalability and Management of Device Identities:** With billions of IoT devices projected to connect to 5G networks, maintaining identities and guaranteeing device authenticity will be a huge problem. Public Key Cryptography allows for scalable and secure device authentication. Each device can be granted a

unique digital certificate using public-key infrastructure and thus helping in identity management even in large and quickly developing networks.

- **Enhanced Security for Device-to-Device Communication:** IoT ecosystems frequently use direct connection between devices (D2D) rather than ongoing contact with a central server. Public Key Cryptography enables secure D2D communication using encryption and digital signatures. It guarantees that data exchanged between devices is both secret and tamper-proof. It is very important in different application scenarios such as healthcare monitoring, self-driving vehicles, etc.
- **Secure Bootstrapping and Firmware Updates:** IoT devices frequently require initial configuration i.e. bootstrapping and periodic firmware upgrades for maintenance and security. Public Key Cryptography protects these operations by authenticating the origins of software updates and configuration instructions. It helps in preventing malicious firmware upgrades as well as configuration modifications that could compromise the device and the network.
- **Compliance and Regulatory Requirements:** Public Key Cryptography into IoT networks helps to satisfy compliance requirements. It plays an important part in complying to regulations such as Europe's General Data Protection Regulation i.e. GDPR that calls for strong personal data protection security measures.
- **Support for Lightweight Cryptographic Algorithms:** Public Key Cryptography is known for its computational. The introduction of lightweight cryptographic algorithms designed for IoT devices overcomes this issue. These methods are intended to deliver the security benefits of Public Key Cryptography like asymmetric encryption and digital signatures without imposing excessive computational on resource constrained IoT devices.

References

[1] Baig, Ahmed Fraz, and Sigurd Eskeland. 'Security, Privacy, and Usability in Continuous Authentication: A Survey'. *Sensors*, vol. 21, no. 17, Sept. 2021, p. 5967.

[2] Zhao, Junhui, et al. 'Authentication Technology in Internet of Things and Privacy Security Issues in Typical Application Scenarios'. *Electronics*, vol. 12, no. 8, Apr. 2023, p. 1812.

[3] E. El-Emam, M. Koutb, H. Kelash and O. Farag Allah, "An optimized Kerberos authentication protocol," *2009 International Conference on Computer Engineering & Systems*, Cairo, Egypt, 2009, pp. 508-513.

[4] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.

[5] Tbatou, Zakariae, et al. 'Kerberos V5: Vulnerabilities and Perspectives'. *2015 Third World Conference on Complex Systems (WCCS)*, IEEE, 2015, pp. 1–5.

[6] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018.

[7] Alqattaa, Ahmed & Aßmuth, Andreas. (2019). Analysis of the Internet Security Protocol TLS Version 1.3.

[8] C. M. Lonvick and T. Ylonen, "The Secure Shell (SSH) Authentication Protocol," RFC 4252, Jan. 2006.

[9] Ćurguz, Jelena. 'Vulnerabilities of the SSL/TLS Protocol'. *Computer Science & Information Technology (CS & IT)*, Academy & Industry Research Collaboration Center (AIRCC), 2016, pp. 245–56.

[10] Wendlandt, Dan & Andersen, David & Perrig, Adrian. (2008). Perspectives: Improving SSH-style Host Authentication with MultiPath Probing. 321-334.

[11] Canetti, R., Krawczyk, H. (2002). Security Analysis of IKE's Signature-Based Key-Exchange Protocol. In: Yung, M. (eds) Advances in Cryptology — CRYPTO 2002. CRYPTO 2002. Lecture Notes in Computer Science, vol 2442. Springer, Berlin, Heidelberg.

[12] P. . -C. Cheng, "An architecture for the Internet Key Exchange Protocol," in *IBM Systems Journal*, vol. 40, no. 3, pp. 721-746, 2001.

[13] D. Carrel and D. Harkins, "The Internet Key Exchange (IKE)," RFC 2409, Nov. 1998.

[14] Alshammari, Majid, and Khaled Elleithy. 'Efficient and Secure Key Distribution Protocol for Wireless Sensor Networks'. *Sensors*, vol. 18, no. 10, Oct. 2018, p. 3569.

[15] Rehman, Md & Govardhan, Dr & Venkat, Tungala & Rao, T.. Design and Implementation of RADIUS–An Network Security Protocol.

[17] C. Metz, "AAA protocols: authentication, authorization, and accounting for the Internet," in *IEEE Internet Computing*, vol. 3, no. 6, pp. 75-79, Nov.-Dec. 1999.
[18] School of Computing and Information Technology, Jomoban Kenyatta University of Agriculture and Technology, PO Box 62000-00200 Nairobi Kenya, et al. 'Identifying Threats Associated With Man-In-The-Middle Attacks during Communication between a Mobile Device and the Back End Server in Mobile Banking Applications'. *IOSR Journal of Computer Engineering*, vol. 16, no. 2, 2014, pp. 35–42.
[19] F. Bersani and H. Tschofenig, "The EAP-PSK Protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method," RFC 4764, Jan. 2007.
[20] Xie, M. , Wang, Y. , Zou, C. , Tian, Y. and Guo, N. (2020) A W-EAP Algorithm for IEC 61850 Protocol against DoS/Replay Attack. *Journal of Computer and Communications*, **8**, 88-101.
[21] Liu, Kaijun, et al. 'A Robust and Effective Two-Factor Authentication (2FA) Protocol Based on ECC for Mobile Computing'. *Applied Sciences*, vol. 13, no. 7, Mar. 2023, p. 4425.
[22] K. Lee, B. Kaiser, J. Mayer, and A. Narayanan, "An Empirical Study of Wireless Carrier Authentication for SIM Swaps," in Proc. of the Sixteenthss Symposium on Usable Privacy and Security (SOUPS 2020), Aug. 2020, pp. 61–79.
[23] W. A. Simpson, "PPP Challenge Handshake Authentication Protocol (CHAP)," RFC 1994, Aug. 1996.
[24] A. F. Gentile, P. Fazio, and G. Miceli, "A Survey on the Implementation and Management of Secure Virtual Private Networks (VPNs) and Virtual LANs (VLANs) in Static and Mobile Scenarios," in Telecom, vol. 2, no. 4, pp. 430–445, 2021, MDPI.
[25] Han, Alex Heunhe, and Dong Hoon Lee. 'Detecting Risky Authentication Using the OpenID Connect Token Exchange Time'. *Sensors*, vol. 23, no. 19, Oct. 2023, p. 8256.
[26] R. Moskowitz and M. Komu, "Host Identity Protocol Architecture," RFC 9063, Jul. 2021.
[27] Kaňuch, P., & Macko, D. (2019). E-HIP: An Energy-Efficient OpenHIP-Based Security in Internet of Things Networks. *Sensors (Basel, Switzerland)*, *19*(22), 4921.
[28] A. Koutsos, "The 5G-AKA Authentication Protocol Privacy," 2019 IEEE European Symposium on Security and Privacy (EuroS&P), 2019, pp. 464-479
[29] Writer, Carsten Gregersen. 'Passwords Aren't Enough – Rethinking IoT Access with Public Key Cryptography'. *IoT For All*, 3 Mar. 2022, https://www.iotforall.com/passwords-arent-enough-its-time-to-rethink-iot-access.