

Excavating Vulnerabilities Lurking in Multi-Factor Authentication Protocols: A Systematic Security Analysis

Ang Kok Wee*
Singapore University of Technology
and Design
Singapore, Singapore
kok_ang@alumni.sutd.edu.sg

Eyasu Getahun Chekole*
Singapore University of Technology
and Design
Singapore, Singapore
eyasu_chekole@sutd.edu.sg

Jianying Zhou
Singapore University of Technology
and Design
Singapore, Singapore
jianying_zhou@sutd.edu.sg

ABSTRACT

Nowadays, cyberattacks are growing exponentially, causing havoc to Internet users. In particular, authentication attacks constitute the major attack vector where intruders impersonate legitimate users to maliciously access systems or resources. Traditional single-factor authentication (SFA) protocols are often bypassed by side-channel and other attack techniques, hence they are no longer sufficient to effectively address the current authentication requirements. To alleviate this problem, multi-factor authentication (MFA) protocols have been widely adopted recently, which helps to raise the security bar against imposters. Although MFA is generally considered more robust and secure than SFA, it may not always guarantee enhanced security and efficiency. This is because, critical security vulnerabilities and performance problems may still arise due to design or implementation flaws of the protocols. Such vulnerabilities are often left unnoticed by the designers or users until they are exploited by attackers. Therefore, the main objective of this work is identifying such vulnerabilities in existing MFA protocols by systematically analysing their designs and constructions. To this end, we first form a set of security evaluation criteria, encompassing both existing and newly introduced ones, which we believe are very critical for the security of MFA protocols. Then, we thoroughly review several MFA protocols across different domains. Subsequently, we revisit and thoroughly analyze the design and construction of the protocols to identify potential vulnerabilities. Consequently, we manage to identify critical vulnerabilities in ten of the MFA protocols investigated. We thoroughly discuss the identified vulnerabilities in each protocol and devise relevant mitigation strategies for each of the vulnerabilities identified. We also consolidate the performance information of the protocols. We believe that the consolidated security analysis and performance information would serve as a single reference point for researchers and practitioners to be aware of the potential security and performance issues when designing MFA protocols. This investigation also reinforces the fundamental need for an enhanced and secure design and implementation of MFA protocols.

KEYWORDS

Multi-Factor Authentication, Authentication Factors, MFA Vulnerabilities, Mutual Authentication, Key Leakage Resilience, Perfect Forward Secrecy, User Anonymity

1 INTRODUCTION

The importance of information security is increasing exponentially due to the dynamic cyber threats posed by malicious adversaries [15, 93]. They typically exploit various types of security weaknesses and flaws to achieve their malicious intent [17, 46]. In particular, authentication-related attacks constitute the major cyber-attacks in the cybersecurity landscape. In this regard, attackers exploit weaknesses in authentication protocols and impersonate legitimate users to gain unauthorized access to a system or service [58]. Therefore, employing effective authentication mechanisms is crucial to alleviate the prevalence of cyber risks in various domains.

Traditionally, authentication protocols are based on a single authentication factor, e.g., passwords, PINs, preshared keys, and biometrics. However, such authentication protocols are no longer sufficient due to various reasons. Weak passwords and password-related vulnerabilities are prevalent concerns. Users frequently choose easily guessable passwords or reuse them across multiple accounts, providing opportunities for attackers to crack passwords and gain unauthorized access [96]. Phishing attacks are also other significant threats in which attackers exploit human vulnerabilities by using fraudulent emails or fake login pages to trick users into revealing their authentication credentials. These attacks can be phenomenally successful even against users with strong passwords [12]. Various side-channel techniques, brute-force attacks, or a combination of the two could allow attackers to leak passwords/PINs or preshared keys. Adversaries can also maliciously copy biometric features or hack into a victim's database in numerous ways. The need for more robust security measures has led to the emergence of multi-factor authentication (MFA) [19].

MFA is often considered to be more robust and secure than SFA. However, it is crucial to recognize that only enforcing MFA may not always guarantee enhanced security. This is because, security weakness and flaws (vulnerabilities) may still arise due to an incorrect design or implementation of the MFA protocols. Since the construction of MFA protocols involves different authentication factors, various secret and public parameters and complex technical issues, it remains challenging to properly detect security flaws in such protocols. In fact, even rigorous formal security proofs usually fail to detect such flaws [89], which is also proven in our analysis discussed below. Consequently, such vulnerabilities are often left unnoticed by the protocol designers or end-users until they are exploited by attackers. If exploited, they may result in disastrous impacts on users. Furthermore, some MFA protocols employ three or more authentication factors to enhance their security guarantee [13, 41, 48, 52]. However, increasing the number of factors or using heavyweight approaches may also impose heavy

*Both authors contributed equally to this work and they share the first authorship.

performance penalties, which might not be tolerable in certain resource-constrained systems.

Therefore, this work aims to detect potential vulnerabilities in existing MFA protocols by systematically analyzing its design and construction information that is publicly available in the respective paper. To this end, we first perform a background study on MFA protocols. Specifically, we first discuss about authentication factors, which are the building blocks of MFA protocols. The design and construction of MFA protocols involves a systematic synergy of different authentication factors. In fact, the security and robustness of MFA protocols is heavily dependent on the synergy and the types of authentication factors used. To better comprehend our study and analysis, we revamp the taxonomy of authentication factors in MFA (cf. Figure 1). In brief, we classify them into two main categories: *conventional* and *emerging* authentication factors. The former involves the commonly used authentication factors and are further classified as *knowledge factors* (e.g., passwords, PINs and security questions), *possession factors* (e.g., physical tokens, security keys, smart cards and mobile authenticator apps), *inherent factors* [10] (e.g., fingerprints[59], facial recognition[83, 86], iris scans[47], and image recognition[84, 85]) and *location factors* [44, 49, 60, 75]. The latter includes recently introduced authentication factors that show high effectiveness, especially in the context of machine-to-machine (M2M) authentications. These includes *historical data* [24, 48], *physically unclonable functions (PUF)* [14, 39?] and *firmware integrity* [26, 27, 30]. Such a systematic classification of the authentication factors is imperative as the use of distinct and independent authentication factors plays a crucial role in improving the security and robustness of MFA protocols. The introduction of various machine-learning techniques [79–82] also enhanced the seamless integration and robustness of different authentication factors in MFA.

Then, we thoroughly conduct a literature review on various MFA protocols. To be more comprehensive, we cover several relevant protocols in different domains, which we classify them as generic client-server systems [28, 31, 88], cloud computing [22, 54, 97], finance [8, 9, 38, 43], healthcare [13, 37, 72, 76], generic IoT [30, 35, 41, 73, 98], healthcare IoT [11, 18, 29, 77, 95], industrial IoT (IIoT) [34, 42, 48, 50, 56, 57, 69, 99], smart cities/home [63, 65, 90, 94], and wireless sensor networks (WSN) [52, 55]. We also highlight the key security requirements and constraints under each domain. We also emphasise on IoT-based multi-factor authenticated key exchange (MAKE) protocols as they involve more critical security and efficiency requirements.

First, we identify several protocols based on its relevance and recency. Then, we revisit and thoroughly analyze the design and implementation of the MFA protocols to identify potential vulnerabilities. More specifically, we delve into the intricate details of their constructions to identify vulnerabilities that can potentially jeopardize the security of the authentication process and future key secrecy. The analysis is performed heuristically based on a set of evaluation criteria we employed for this purpose. While different users adopt different set of security evaluation criteria, we formed our own set of criteria. It encompasses both existing and our newly introduced ones, which we believe are very critical for the security of MFA protocols. Then, we thoroughly evaluate several MFA protocols based on the formed criteria. Consequently, we manage

to detect significant vulnerabilities in ten of the MFA protocols investigated, which could be readily exploited by an attacker. We thoroughly discuss the identified vulnerabilities and consolidate the performance information of the protocols.

The identified vulnerabilities are related to the lack of: explicit mutual authentication, independence of authentication factors, distinctiveness of authentication factors, leakage resilience, perfect-forward secrecy, user anonymity, resilience against known attacks, and realistic adversarial assumptions. Finally, we propose relevant mitigation strategies for the identified vulnerabilities. We believe that the consolidated information provided would serve as a single reference point for researchers and practitioners to be aware of the potential security issues when designing MFA protocols. It also helps to apply the necessary mitigation strategies to the vulnerable ones.

We believe that this work can provide valuable insights to security researchers and practitioners to better understand potential vulnerabilities that may exist in the design of MFA protocols and the attack vectors that could be utilized by adversaries. Remarkably, most of these vulnerable protocols we identified are published in top cybersecurity journals and conferences. However, those vulnerabilities went unnoticed during the design, implementation, testing, and peer-review processes. Although most of the authors provided rigorous formal proofs on their protocols, they failed to detect those flaws. This implies that heuristic analysis could sometimes be even more effective than formal proofs in certain contexts. Therefore, we believe that insights from this analysis could serve as the basis for proposing effective mitigation strategies to enhance the design of MFA protocols per the established security evaluation criteria. Outcomes of this work can also significantly contribute to the ongoing efforts in the research community to strengthen authentication protocols and mitigate the security risks of improper MFA designs.

While there are existing works that tried to detect vulnerabilities in MFA protocols, they only cover certain security issues on some protocols using different evaluation criteria [21, 55, 67, 96]. Some others provide a survey on the security of MFA protocols [19, 55, 66, 74, 92], but they did not perform any new security analysis to detect vulnerabilities in the design and construction of the MFA protocols. To the best of our knowledge, this work is the first to provide a comprehensive security analysis, especially on design- and construction-level vulnerabilities of MFA protocols, using a new set of security evaluation criteria.

Overall, the main objectives of this work are: 1) detecting critical security flaws in MFA protocols and report them before they are exploited; 2) showing that simply adding multiple authentication factors may not always guarantee enhanced security; 3) testifying the relevance of a heuristic analysis in such contexts (even performs better than formal analysis in some cases); 4) creating awareness on the critical design flaws in MFA protocols and provide mitigation strategies; and 5) providing insights and mitigation strategies towards the design of more secure MFA schemes.

In sum, this work makes the following main contributions.

- We perform a systematic review of several MFA protocols across different domains, which can serve as a single-point of reference about the state-of-the-art MFA protocols.

- Our work goes beyond the conventional survey work. Because, we also systematically analyze the security of several MFA protocols. To this end,
 - (1) We first form a set of security evaluation criteria (by introducing new ones and adopting some existing ones) that can be used to critically assess the security of MFA protocols.
 - (2) We thoroughly evaluate the protocols based on the formed criteria and managed to identify several critical vulnerabilities in ten of the protocols.
- We devise appropriate mitigation strategies for the vulnerabilities identified, based on our own perspectives and from certain existing sources.
- We believe that this work provides sufficient insights to the community about design-level security weaknesses and flaws in MFA protocols.

Organization. The remainder of this article is structured as follows. Section 2 provides the relevant background on MFA and authentication factors. Section 3 provides our reviews of existing MFA protocols across various domains. Section 4 highlights our adversarial assumptions used to assess the security of MFA protocols. In Section 5, we thoroughly analyze and discuss the flaws and vulnerabilities of the selected MFA protocols. In Section 6, we devise relevant mitigation strategies for the vulnerabilities identified. Section 7 concludes the article by outlining relevant future works.

2 BACKGROUND

2.1 Overview of MFA

MFA is employed in diverse contexts to ensure secure access to systems, applications, or sensitive information. It strengthens security measures by requiring individuals or entities to provide multiple authentication factors that prove their identity before they are allowed to access the system or application. This approach goes beyond traditional SFA and adds additional layers of security by combining multiple authentication factors.

As highlighted in the introduction, the reliance on a single authentication factor, such as passwords, PINs, secret keys, and biometrics, is no longer sufficient for the current security trends and requirements. So, by reducing the dependence on a single authentication factor, MFA can impose stronger verification mechanisms. Even if one of the factors is compromised, the inclusion of additional factors in MFA prevents attackers from advancing without presenting the complete set of authentication factors. It is worth noting that MFA is nowadays commonly required in regulatory standards [1, 45] and recommended in advisories [2, 3] to safeguard the security of systems and sensitive information.

Authentication factors are the building blocks of MFA protocols. They are combined systematically to form an MFA protocol with adequate security guarantees. In this work, we classify them into conventional and emerging categories. These categories and their respective subcategories are discussed in detail below and illustrated in Figure 1 with examples.

2.2 Conventional authentication factors

The four categories of authentication factors typically used in MFA protocols are knowledge factors, possession factors, inherent factors, and location factors [92].

2.2.1 Knowledge factors. Knowledge factors refer to something the individual knows, such as a password, PIN, or security questions. It is widely employed to verify an individual’s identity and grant access to systems. Knowledge factors are based on the assumption that the individual is solely the one who knows the information. However, knowledge factors may potentially be susceptible to known attacks. As highlighted in earlier, passwords can be weak or easily guessed. Security questions can sometimes have answers that can be easily obtained or guessed [70].

2.2.2 Possession factors. Possession factors involve something the individual possesses. A typical example of a possession factor is a physical token, such as a hardware security key or a smart card. Using such tokens, cryptographic algorithms are typically utilised to generate a unique code or response to verify the user’s identity. Mobile authenticators are another form of a possession factor. They are applications installed on an individual’s mobile device. Mobile authenticators employ algorithms such as HMAC-based One-Time Password (HOTP) or Time-based One-Time Password (TOTP) to generate a One-Time Password (OTP) at regular intervals. The individual is required to enter the generated OTP to verify their identity.

2.2.3 Inherent factors. Inherent factors rely on something the individual is or has, typically related to an individual’s biological traits or physical characteristics. These traits are difficult to replicate and provide an elevated level of security [10]. Fingerprints, facial recognition, and iris scans are examples of inherent factors.

These inherent factors offer several advantages in authentication. They provide high accuracy and security since they are difficult to counterfeit or manipulate. Additionally, they eliminate the need for users to remember passwords or carry physical tokens, enhancing convenience and user experience.

2.2.4 Location factors. The use of location as an additional authentication factor has been introduced recently [60]. This factor considers the individual’s physical presence and compares it to their expected or usual location. If the user’s location deviates significantly from their regular pattern or appears suspicious, it can trigger additional verification steps. The prevalence of mobile devices and technological advancements like Global Positioning System (GPS) [44], IP geolocation [75], and proximity authentication through ambient sounds [49] have led to the increased prominence of location-based authentication.

2.3 Emerging authentication factors

Traditional authentication factors possess certain limitations, especially when employed for machine-to-machine (M2M) authentications. Hence, they pose a challenge in ensuring secure M2M communications and transactions. Machines, unlike humans, cannot produce and provide authentication factors like password or

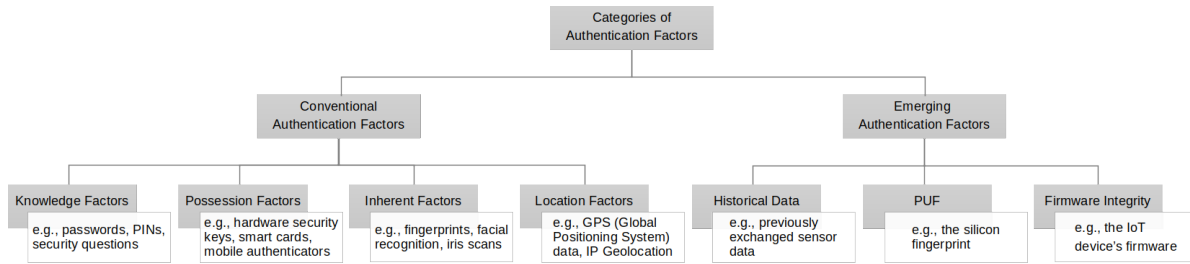


Figure 1: Taxonomy of authentication factors

biometric data. New types of authentication factors, such as historical data, PUF and firmware integrity, are emerged to address these issues.

2.3.1 Historical data. Historical data [24] (e.g., previously exchanged sensor data between a client and server) is recently introduced as a strong authentication factor due to its dynamic nature and high leakage resilience. It is incorporated by selecting a set of indices from the historical dataset that is continuously expanding and using the corresponding data and tags to compute the response as part of the authentication process. As historical data is constantly expanding (even if an adversary compromises the data), it will get outdated soon after, leaving limited room for exploitation.

2.3.2 Physically unclonable functions (PUF). PUF [14, 39] utilise the physical distinction of each electronic device to generate a unique response when provided with a challenge. These distinctions are typically introduced during the manufacturing processes. Given the difficulty in replicating the same response, PUF is used as an authentication factor. While there may be slight variation in the response due to external factors, it can be rectified using fuzzy extractors.

2.3.3 Firmware integrity. Firmware integrity [30] utilise the IoT device's firmware to verify its integrity and authenticity. Unlike traditional computing devices, most IoT devices have firmware as their operating system and interact directly with the hardware to complete the required assignments. With technological advancement, some IoT devices are installed with embedded Operating Systems (OS) such as Windows CE, embedded Linux OS, etc., where the image files are typically stored together with other essential files in a flash or embedded multi-media card. Any unauthorised modification would affect the integrity of the stored content, simplifying the authenticity checks of the IoT device by verifying the firmware integrity.

3 LITERATURE REVIEW ON MFA PROTOCOLS

Before we dive into our security analysis of the vulnerable MFA protocols in Section 5, we would like to provide a summary of the literature reviews we conducted in several MFA protocols. Because, the literature review helps to better comprehend the current trends and advancements in MFA security. With the multitude of MFA protocols published, it is necessary to understand the approaches undertaken by researchers when designing the MFA protocols.

In recent years, extensive research and development efforts have resulted in the development of numerous MFA protocols tailored

to specific domains, including but not limited to cloud computing, finance, healthcare, and the rapidly expanding field of the internet of things (IoT). Numerous research papers have also been published to analyze and compare MFA protocols to identify and rectify security weaknesses and flaws, ultimately improving the overall security of MFA protocols. To identify vulnerabilities in MFA protocols, we reviewed several research papers in different domains. Due to space limitation, we discuss below only the selected ones that we believe are more relevant in each respective domain. Out of which, we identify vulnerabilities in ten of them, which are discussed in detail in Section 5.

3.1 Generic client-server architecture

Ertan et al. [88] proposed a two-factor authentication protocol (using TOTP and PUF as the first and second authentication factors) to alleviate the security posture of TOTP systems. The MFA protocol uses PUF as storage to securely store the client's secret. The authors assumed that Transport Layer Security (TLS) is in place to secure the communication channel between the entities.

Ivaylo Chenchev [31] proposed an MFA protocol secure communications in a client-server or peer-to-peer architecture. The protocol uses time-based onetime passwords (TOTP) and dynamically generated passwords (which is not stored anywhere) as its authentication factors. This protocol mainly intended to address the vulnerabilities of traditional password-based MFA protocols that potentially arise due to insecure generation of passwords.

Chen et al. [28] proposed a biometrics-based three-factor authentication and key agreement scheme for multi-server environments. The authors mainly focused on addressing the weakness of conventional MFA protocols when adopted in a multi-server environment. This protocol employs user id, password and digital signature as authentication factors. Using this protocol, the authors claimed to achieve several impersonation and replay attacks.

3.2 Cloud computing

As the adoption rate of cloud computing increases, adversaries targeting cloud services are also rising. MFA improves the robustness and efficiency of the authentication process, which directly enhances the overall security posture of cloud-based systems, safeguarding sensitive data from potential cyber threats.

Bouchaala et al. [22] proposed a cloud-based MFA protocol using user id and password, and additionally, smart card as authentication factors. The protocol is based on elliptic curve cryptography

(ECC) and consists of four phases that include registration, authentication, key update, and card revocation. The key update phase facilitates the renewal of cryptographic keys stored in the smart card to maintain security, while the card revocation feature allows for the immediate revocation of compromised or lost smart card to reduce the likelihood of security compromise. Through the formal and informal security analysis conducted, the authors claimed that the MFA protocol is able to prevent attacks posed by adversaries.

Lee et al. [54] proposed a three-factor MFA protocol explicitly tailored for the cloud environment. The protocol employs password, smart device, and biometric as authentication factors. It consists of four essential phases, including the registration phase, login and authentication phase, password change phase, and identity update phase. Users are allowed to update their password and modify their personal information as needed. The authors proved that protocol was able to perform mutual authentication and establish a secured channel between both parties.

Lee et al. [97] proposed an innovative and improved three-factor authentication protocol to mitigate known attacks such as replay attacks, offline guessing attacks and Denial-of-Service (DoS) attacks. Besides password and biometric as the first and second authentication factors, the protocol allows the flexibility of using a laptop and/or smart card as the third authentication factor. Users can still complete the authentication process if the laptop or smart card is lost. The protocol consists of a registration and authentication phases.

Otta et al. [66] systematically surveyed MFA protocols explicitly tailored for securing cloud infrastructure. The paper covered the following key aspects: threats related to cloud authentication, analysis of the different types of authentication factors used in MFA, and a comparative analysis of MFA protocols designed by various researchers. The findings presented in the paper contributed to a structured approach in selecting the authentication factor, which can thwart impersonation attacks based on its uniqueness.

Google developed Google Authenticator to provide its users with an additional layer of security [87]. With the enhancement, besides furnishing their username and password, users are required to furnish the OTP generated by the Google Authenticator registered to their account as the second authentication factor to verify their identity when accessing Google services. Any other services or applications that support TOTP algorithm can also utilise Google Authenticator as the second authentication factor.

3.3 Finance

MFA is also widely adopted by financial sectors, such as banks, insurance companies, and various payment systems, to securely authenticate individuals before allowing them to perform any financial transactions. Scam and fraud transactions, and privacy of clients are some of the critical security concerns in this domain.

Hassan and Shukur [43] proposed a three-factor authentication protocol to enhance the authentication process for an electronic payment system. It combines three authentication factors: password, biometric, and OTP. The protocol consists of three phases: registration, authentication and transaction phases, to safeguard the payment system against a wide range of attacks.

Durairaj and Ramachandran [38] proposed an innovative approach to their ECC-based authentication protocol. In addition to low entropy password, International Mobile Subscriber Identity (IMSI) of the user's device, and fingerprint, Durairaj and Ramachandran proposed to include the feature extracted from voice print, processed using Mel Frequency Cepstral Coefficient algorithm (MFCC) as an additional authentication factor. They proved that the MFA protocol is able to establish secure communication channels between two entities after successful mutual authentication.

Similarly, Abiew et al. [8] also adopted an innovative low-cost approach using another authentication factor, Keystroke Dynamics, to improve the authentication process. Through the experiment conducted, they proved that the protocol is able to mitigate the vulnerability of ATM PINS to dictionary attacks.

Aburbeian et al. [9] proposed a protocol that integrates MFA and machine learning to secure financial transactions. The protocol involves two stages security. In the first stage, the protocol employs fingerprint and OTP as authentication factors to authenticate users. In the second stage, the protocol involves a machine learning layer, which employs facial recognition as a decisive and third authentication factor. This is to further enhance the security and robustness of the authentication process. The authors claimed to have achieved several security features with high accuracy.

3.4 Healthcare

In healthcare, researchers have focused on developing MFA protocols particularly designed to protect lost of patient data, privacy breaches and unauthorised access to health-related information.

Sabeeh and Yassin [72] proposed a two-factor authentication protocol to authenticate the administrator in the healthcare system using password and SMS token as authentication factors. The components include an administrator, user data entry, and a healthcare centre and are based on typical authentication phases such as initialization, registration, and authentication phase. Scyther tool was utilised to assess the security properties of the MFA protocol and proved to be capable of resisting known attacks.

Shamshad et al. [76] proposed an improved ECC-based two-factor authentication protocol that employs password and smart card as authentication factors to enhance the authentication process. Besides the typical registration and authentication phases, the protocol includes a password change phase to facilitate the change of password when required. While the performance of the protocol was observed to be sub-optimal, they proved that it is capable of resisting known attacks that affect other more efficient MFA protocols.

Ali et al. [13] discovered that the protocol of Barman et al. [20] was vulnerable to various attacks, such as session key leakage, server impersonation, and user impersonation attacks. Then, Ali et al. [13] proposed an improved three-factor MFA protocol to address those security flaws in [20]. This protocol employs password, smart card, and biometrics as its authentication factors. The protocol also involves a user revocation feature, which allows to revoke users in case of loss of their smart card. The authors claimed to achieve several security properties, such as key resilience against key leakage attacks, impersonation attacks, and known attacks.

Dhillon et al. [37] proposed a three-factor authentication protocol (involving password, biometrics, and smart card as its authentication factors) to establish a secure communication channel between a medical professional and a cloud server. The authors' informal and formal security analysis demonstrated that the proposed MFA protocol is capable of resisting various known attacks.

3.5 IoT

The rapid proliferation of IoT devices has contributed to the surge in designed MFA protocols. The authentication process for IoT devices is unique due to resource-limited devices and M2M communication scenarios where human involvement is minimal or absent. That poses challenges for traditional authentication factors like PINs, passwords, and biometrics. Within the IoT domain, different subdomains such as Healthcare IoT, industrial IoT (IIoT), smart cities/homes, and wireless sensor networks (WSN) require specifically tailored MFA protocols to address their respective requirements and challenges. To provide a more comprehensive and thorough analysis, we further classify the MFA protocols of this domain into the following sub-categories.

3.5.1 Generic IoT. Cvetković et al. [35] presented on the application of MFA protocols in the context of IoT. With the complexity of IoT architecture taken into consideration, the paper briefly discussed the various IoT security techniques that stood out, the security objectives of IoT, selected MFA protocols relevant to IoT, and security considerations specific to MFA implementation in the IoT. The findings presented in the paper contribute to understanding MFA protocols suitable for IoT environments and provide insights into the challenges and considerations involved. With the limitations faced, they highlighted the need for efficient resource utilization, lightweight security authentication protocols, and cryptographic algorithms customised for the IoT environments.

Halvor Vada presented a comparative analysis of various MFA protocols tailored for IoT systems. Vada highlighted the security challenges inherent in IoT systems at each layer of the selected 3-layer architecture and emphasized the importance of a robust authentication protocol to enhance security. The paper offered valuable insights into the implementation of MFA protocols for IoT systems by comparing a range of MFA protocols based on authentication factors employed, performance, strengths, and weaknesses.

Chen et al. [30] introduced a novel authentication protocol designed specifically to enhance the security of IoT devices through MFA. This protocol incorporates a hierarchical architecture based on the traditional IoT system, comprising the perception, network, and application layers. In addition to a secret cryptographic key and PUF, the integrity of the device's firmware, which is used to ensure integrity, is also employed as one of the authentication factors. The protocol also supports firmware updates, enabling the deployment of patches and security enhancements.

Mirsaraei et al. [41] proposed an ECC-based MFA protocol on a blockchain platform for generic IoT devices. It employs password, biometric, and smart card as the authentication factors. In addition to the typical registration phase, login phase, and authentication phase, an update phase was included to facilitate password and biometric information updates. Through the informal and formal

security analysis, the authors proved that the proposed MFA protocol satisfies the typical security requirements and is capable of resisting known attacks. Automated Validation of Internet Security Protocols and Applications (AVISPA) tool was utilised to automate the formal security analysis of this protocol.

Zahednejad et al. [98] proposed a two-factor authentication protocol for generic IoT devices in an M2M context. The protocol employs a long-term cryptographic key and data items as authentication factors. The three phases include initialization, authentication, and a revocation phase to revoke compromised or lost devices. To simulate real-life scenarios, they assumed a strong adversary capable of compromising the server and retrieving all the information within and proved that the MFA protocol was able to ensure a secure authentication process and repel attacks.

Sadhukhan et al. [73] proposed an ECC-based lightweight three-factor authentication (involving user identity, password and biometric data as the authentication factors) protocol for remote users in IoT network. The authors performed formal and informal security analysis to demonstrate that their proposed MFA protocol achieves several security features, such as resilience against impersonation and DoS attacks, perfect forward secrecy (PFS), and user anonymity, among others.

Melki et al. [61] proposed a lightweight ECC-based two-factor authentication protocol to provide secure communication channels between an IoT device and a gateway. A secret session identifier ID_S (that is derived from a PUF output value) and a secret channel-based parameter σ_i are used as authentication factors. The authors claimed to have achieved several security features, such as PFS, session secrecy, user privacy, and resilience against reply, side-channel and man-in-the-middle attacks.

3.5.2 Healthcare IoT. Jia et al. [95] proposed a two-factor authentication protocol for a fog-driven IoT healthcare system using password and smart card as authentication factors. This protocol designed is specifically for healthcare applications in the IoT domain. It involves several phases, including system setup, registration of users and fog nodes, authentication, and key agreement. In addition, password updates, user revocation and re-registration, and fog node revocation were also included to ensure comprehensiveness. The bilinear pairing was utilized to compute cryptographic operations efficiently. Formal and informal security analyzes against known attacks were also performed to demonstrate the security of the MFA protocol.

Al-Saggaf et al. [11] introduced an innovative and improved two-factor authentication protocol specifically designed for the IoT-enabled healthcare ecosystem. The protocol combines smart card and biometrics as authentication factors. Given the challenges in post-quantum computing, the authors utilized a Post-Quantum Fuzzy Commitment scheme (PQFC) to protect the biometric template. It consists of several key phases, including setup, registration, login, authentication, and biometric revocation. By incorporating biometric revocation capabilities, the use of suspected compromised biometrics is prevented.

Azroul et al. [18] designed a two-factor authentication protocol to address the vulnerabilities observed in a prior work [77]. This protocol employs password and smart card as authentication factors to secure communications between healthcare systems in the

cloud IoT. It encompasses several phases, including system setup, registration of new sensors and users, login and authentication, and password update. Formal security analysis was performed using the Scyther tool and the ROM model, in addition to informal security analysis to verify the security of the MFA protocol.

Chen et al. [29] proposed a privacy-preserving three-factor MFA scheme to secure a cloud-assisted medical IoT. The protocol employs user id, password and biometric data as the authentication factors. The authors claimed to have achieved several security properties, such as post-quantum security, PFS, user anonymity, and resilience against reply and impersonation attacks.

3.5.3 IIoT. A wide range of industrial IIoT-based MFA protocols has been proposed over the years [34, 48, 50, 56, 57, 69, 99]. Sain et al. [74] provided an overview of security issues related to Cyber Physical Systems (CPS) and explored the use of MFA to improve security. The authors also discussed the evolution of authentication, elaborated on the importance of MFA, and highlighted the rapid adoption of biometrics as one of the authentication factors. MFA protocols designed for CPS were deliberated and compared against an established set of evaluation criteria.

Zhang et al. [56] proposed a blockchain-based MFA protocol designed specifically for cross-domain IIoT systems. This protocol combines a long-term cryptographic key and PUF as authentication factors to enhance security. The protocol encompasses various phases, including registration, intra-domain authentication, cross-domain authentication to ensure a secure authentication process. A formal security analysis using BAN logic was performed to prove the security of the MFA protocol.

Khalid et al. [50] introduced an MFA protocol designed for cross-platform IIoT systems. This protocol combines password, smartcard, and biometric authentication factors to enhance the authentication process. The protocol utilizes the AES-ECC algorithm to secure communications between the entities. The protocol encompasses various phases, including setup, user and fog node registration, login, and authentication, ensuring only access to authenticated users. BAN logic was employed to perform a formal security analysis to prove the security of the MFA protocol.

The three-factor authentication protocol proposed in [69] employs a combination of authentication factors, including password, smart card, and biometric, to ensure secure communication channels. The protocol consists of seven phases: offline sensing device and user registrations, login, authenticated key agreement, biometric and password update, and dynamically sensing device addition and revocation. The security of the MFA protocol was proven using Real-Or-Random model (ROR).

A newly designed low-interactivity Multi-Factor Authenticated Key Exchange (MFAKE) protocol named Secure Remote Multi-Factor protocol was introduced in [56] and aims to enhance the security of M2M communication within the IIoT. The protocol ensures a robust authentication process by leveraging authentication factors such as password, biometric, and OTP. The key exchange was proven to be secured using the Bellare-Pointcheval-Rogaway model.

Liu et al. [57] introduced an innovative two-factor authentication protocol to enhance security in M2M communication within the IIoT landscape. The protocol employs long-term private keys and

big data tags as authentication factors. The protocol begins with an initialization phase where the entities generate their respective private keys associated with the big data tags. The IoT device and the server need to provide evidence or proof that they possess knowledge of the data and its associated tag before establishing a secure connection between the two during the authentication phase. They demonstrated that the MFA protocol can achieve Key Compromise Impersonation (KCI) and Server Compromise Impersonation (SCI) Resilience, which are critical in IIoT.

Jin et al. [48] proposed a historical data-based multi-factor authenticated and confidential channel establishment (HMACCE) protocol for the M2M communication within the IIoT. This protocol involves a client and a server as the entities, with the client possessing a long-term symmetric key and a secret key as authentication factors. In contrast, the server holds a long-term symmetric key, historical data, and data tags. The protocol encompasses three phases, including initialization, tag generation, and online authentication and key exchange. During the tag generation phase, authentication tags are generated using the historical data and data tags, enabling subsequent verifications. They also proposed another HMACCE protocol named π_{FS} to address the issue of adaptive bounded leakage.

Cui et al. [34] introduced a three-factor authentication protocol to satisfy the established security requirements of IIoT environments. The protocol incorporates three authentication factors, including password, biometric, and smart card. It encompasses various phases such as server initialization phase, smart device registration phase, user registration phase, user login phase, authentication and key agreement phase, password and biometric update phase, smart devices addition phase, and user revocation phase. All smart devices and users must be registered before communications are allowed. They demonstrated the security of the MFA protocol through informal and formal security analysis using ROR model.

Han et al. [42] proposed a three-factor MFA and key agreement protocol to secure the communication in IIoT that specifically involves three entities, namely user, gateway and sensing device. The protocol employs user id, password and biometric key (that is generated from the users biometric data using a fuzzy extractor) as its authentication factors. The protocol is mainly designed to address the vulnerabilities of a prior work [71], such as lack of forward security and vulnerability to insider and session specific temporary information (KSSTI) attacks.

3.5.4 Smart cities/homes. The two-factor authentication protocol for smart cities proposed in [65] incorporates password and smart card as authentication factors. The protocol consists of several phases, including setup, user and sensors registration phase, login phase, authentication and key exchange phase, update phase, and revocation phase to allow for the removal of user privileges if necessary. Besides informal security analysis using BAN logic, which was typically used for formal security analysis, AVISPA tool was employed to prove the security of the MFA protocols. A comparison with other related MFA protocols was also conducted to demonstrate its effectiveness.

Wang et al. [90] designed an improved ECC-based three-factor authentication protocol based on ECC for smart homes using password, smart card, and biometrics as authentication factors. The initialization, registration, login and authentication, and password update phase were involved in the authentication process. They demonstrated the security of the MFA protocol through formal security analysis using BAN logic and demonstrated its efficiency through performance comparison with other related MFA protocols.

The two-factor authentication protocol proposed in [63] was designed to address the security issues observed in [94]. Password and mobile device are employed as authentication factors in the MFA protocol. The protocol involves a series of phases, from initialization and device registration to authentication and key agreement processes. The protocol also supports password updates for enhanced security. Secure communication and interaction between the Mobile User and the Smart Device are established through the Home Gateway. A combination of tools, including BAN logic, ROR model, and AVISPA tool, was utilised to demonstrate the security of the MFA protocol.

3.5.5 WSN. Wang et al. [89] aims to investigate and understand the failures observed in the security proofs of MFA protocols specifically designed for mobile devices. The paper provided an overview of MFA protocols designed for mobile devices, an understanding of security proofs failures in MFA protocols, the development of an enhanced set of evaluation criteria, and an analysis of a selection of ten MFA protocols. They also proved that protocols with formal security proofs were able to better satisfy the established evaluation criteria which aid in the design of a more secure MFA protocol for mobile devices.

The three-factor authentication protocol designed by authors in [55] for WSNs consists of three phases, including registration of users and sensors phase, login and authentication phase, and password change phase. It employs password, biometric, and smart card as authentication factors. All users and sensors must register with the gateway before establishing any connection. The protocol utilizes "honey list" and "fuzzy extractor" techniques to enhance security. ProVerif tool was employed to prove the security of the MFA protocol. The protocol's performance also proved to be exceptionally better compared to other related MFA protocols.

Kumar et al. [52] proposed a three-factor authentication protocol incorporating password, smart card, and biometric as authentication factors. They incorporated several phases, including the initialization base station phase, user and sensor node registration phase, login and authentication phase, and password and session key updating phase, into the protocol design where users can update their password and generate new session keys for enhanced security. The protocol also allowed the inclusion of new sensor nodes into the network with proper authentication procedures through the adding new nodes phase. The ROR model and the AVISPA tool were employed to demonstrate the security of the MFA protocol, which proved to satisfy the security requirements.

3.6 Mobile authenticators

Recently, mobile applications are widely used as authentication factors in most of the domains mentioned above. For example, Google developed Google Authenticator to provide its users with

an additional layer of security [87]. With the enhancement, besides furnishing their username and password, users are required to furnish the OTP generated by the Google Authenticator registered to their account as the second authentication factor to verify their identity when accessing Google services. Any other services or applications that support the TOTP algorithm can also use Google Authenticator as the second authentication factor.

However, such authenticators are also not without security risks. In a recent publication [67], researchers discovered vulnerabilities in several mobile authenticators that exposed the unique secret key. The affected authenticators (as illustrated in Figure 2) includes Epic Authenticator, Google Authenticator, Microsoft Authenticator, Sophos Authenticator, Red Hat Free OTP, and Twilio Authy Authenticator. These vulnerabilities allowed an adversary to access the unique secret key stored plainly at specific repository locations, such as directories or database files. In addition, the unique secret key can also be retrieved from memory during specific periods.



Figure 2: Affected authenticator applications

4 ADVERSARIAL MODEL

In order to assess the security of MFA protocols, it is essential to assume a realistic and concrete adversary model. In fact, one of the limitations of most MFA protocols is failing to assume strong and realistic adversaries. Therefore, it is crucial to consider a realistic adversary model. In our model, we consider the following capabilities of the adversary in an MFA context:

- The adversary has full control of messages transmitted over the public channel, i.e., it can intercept, eavesdrop, and redirect it.
- The adversary can acquire design of the proposed MFA protocol.
- The adversary can acquire the first authentication factor, e.g., password.
- A strong adversary can obtain all the data from the device if he gets access to the device (cf. Figure 3 for the comparison between weak and strong adversaries).
- In case of PFS attack, we assume that the adversary can obtain the long-term secrets of both parties.

5 SECURITY ANALYSIS OF MFA PROTOCOLS

As highlighted in the introduction, we perform critical security analysis on several MFA protocols to identify potential security flaws. Of these, we identify serious security flaws in ten of the protocols. In this section, we provide a detailed discussion of the vulnerabilities identified in these protocols. In addition, since the runtime performance of the protocols is critically important, especially in resource-constrained devices and/or hard real-time constrained systems, we also present a comparative performance analysis of the protocols to assess their efficiency.

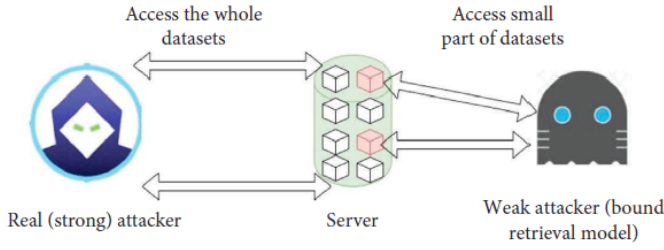


Figure 3: Snapshot of the comparison between a strong and weak attacker [98]

5.1 Evaluation criteria

To assess the security of an MFA protocol, it is essential to establish a set of evaluation criteria. Different researchers adopted different sets of security evaluation criteria considering various domains and contexts. After analyzing the criteria in the existing frameworks, we introduce some new criteria, such as distinctiveness of authentication factors, independence of authentication factors, and leakage resilience, to form our set of evaluation criteria. In sum, it comprise 8 criteria, which are briefly discussed and summarized in Table 1. We believe these criteria are very essential to evaluate MFA protocols in different domains, including the multi-factor authenticated key exchange (MAKE) domain in IoT settings.

5.2 Security analysis

As discussed in the preceding sections, we identified vulnerabilities in ten of the MFA protocols we analyzed. In this section, we discuss a detailed account of the vulnerabilities identified.

5.2.1 Vulnerable MFA Protocol 1 [48]. As discussed in Section 3.5.3, Jin et al. [48] proposed a historical data-based multi-factor authenticated and confidential channel establishment (HMACCE) protocol for the M2M communication within the IIoT, as illustrated in Figure 4. This protocol involves a client and a server as the entities. The client possesses a long-term symmetric key and a tag generation secret key as first and second authentication factors while the server possesses a long-term symmetric key, historical data and data tags as first, second and third authentication factors, respectively. The protocol encompasses three phases, including initialization, tag generation, and online authentication and key exchange. During the tag generation phase, authentication tags are generated using the historical data and data tags, enabling subsequent verifications. The authors claimed to address the key leakage issues of a prior historical data-based two-factor authentication protocol [25]. This protocol comprises two versions, namely π_{woFS} and π_{FS} , that are designed without and with forward secrecy, respectively. The latter is also claimed to address the issue of adaptive bounded leakage.

Based on our analysis, this protocol exhibits several vulnerabilities: 1) The authors claimed that adding an additional authentication message will ensure mutual authentication between the client and server. Based on the authentication steps provided, it is true that the client can verify the authentication message, $M = h(mk||Y||sid||Auth')$, sent by the server. However, further checks on the MFA protocol revealed that the client does not send an explicit authentication message to the server. Hence, the server

cannot verify authenticity of the client, therefore failing the mutual authentication criteria *C1*. 2) Both of the client’s authentication factors, i.e., the symmetric authentication key mk and the tag generation secret key K , are in the same authentication factor category, i.e., possession factors. Given that it does not fully satisfy the intent of MFA, where multiple authentication factors from different categories should be used to prove identities [62], it fails the authentication factors distinctiveness criteria *C2*. 3) The second authentication factor t_i of this protocol is protected using the first authentication factor $sk_{ids, idc}^1$ during transmission, hence failing the authentication factors independence criteria *C3*. 4) During the tag generation phase, in which the client transfers a piece of data d_i to the server, the server computes an authentication tag t_i using K , which is the second authentication factor of the client. The equation used to derive t_i is:

$$t_i = K.h(d_i||i) + k_i(modp)$$

As a result, the computation of the tag t_i depends on the value of the second authentication factors of the client and the server. Any compromise to the second authentication factor could potentially lead to the compromise of the third authentication factor, hence failing the criteria *C3* again. 5) This protocol is also susceptible to data- and tag-stealing attacks in which adversaries could retrieve all the historical data and tags. Since data and tags are used as authentication factors, such leakage can also compromise the authentication factors and the session keys. Therefore, it fails the leakage resilience criteria *C4*. 6) The entropy of its sensor data (which is used as authentication factor) is between 4.52 and 7.80, which is low and could easily be predicted. Hence, it fails the criteria *C4* again. 7) This protocol does not employ any client anonymity mechanism, thus failing criteria *C6*. 8) The authors assume a weak adversary (called a bounded-retrieval model) who can access only a fraction of the historical data after he compromises the server. This is an unrealistic assumption and fails the criteria *C8*. On the other hand, the second version (i.e., π_{FS}) of this protocol satisfies PFS while the first version (i.e., π_{woFS}) does not.

5.2.2 Vulnerable MFA Protocol 2 [41]. As discussed in Section 3.5.1, Mirsaraei et al. [41] proposed an ECC-based MFA protocol on a blockchain platform for generic IoT devices. It employs password, biometric, and smart card as the authentication factors. The authors conducted an informal and formal (e.g., using the AVISPA tool) analysis and claimed that their protocol is capable of satisfying a set of established security requirements and resisting attacks including key leakage, MITM attacks, DoS attacks, etc.

We perform further analysis in this protocol to identify potential security flaws. In their security analysis, it was assumed in “Assumption 5” that “the malicious attacker can obtain only one parameter in an equation at a time.” It is crucial to note that adversaries can employ various sophisticated techniques and strategies to acquire multiple parameters simultaneously. Given that a strong adversary could potentially acquire the information discussed in Section 4, there is a risk in this protocol that the session key can be computed and compromised. The session key is computed using the following equation:

$$SK = M_2 \oplus V_1$$

Table 1: Evaluation criteria

S/No.	Evaluation criteria	Description
C1	Mutual authentication	Both parties must verify each other's identities before advancing the authentication process and establishing a session key.
C2	Distinctiveness of factors	Employ distinct authentication factors each selected from different categories, such as knowledge factors, possession factors, inherent factors, historical data, etc.
C3	Independence of factors	Ensure each authentication factor is independent from other. For example, the generation of one factor must not depend on any other factor. Similarly, one factor must not be protected (e.g., encrypted) by using any other factor.
C4	Leakage resilience	Ensure that any data leakage cannot compromise keys or authentication factors. In addition, keys and authentication factors must be computationally infeasible to be predicted or guessed.
C5	Perfect-forward secrecy (PFS)	Ensure that the leakage of long-term keys (client or server) cannot compromise the security of previous sessions.
C6	User anonymity	Preserve user's identity during the authentication process.
C7	Resilience against known attacks	Protect the authentication process against known attacks, such as MITM, replay, password-guessing, impersonation, insider, and DoS attacks.
C8	Adversary assumption	Assume strong and realistic adversaries who possess adequate skills and resources to perform sophisticated attacks.

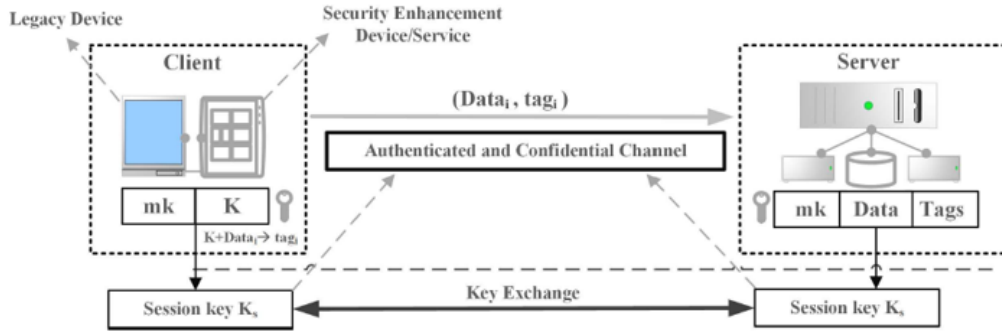


Figure 4: Overview of the HMACCE Protocol [48]

Using the information acquired discussed in Section 4, the adversary can easily obtain the values of M_2 and V_1 to compute SK . This is because, M_2 , which is transmitted through the insecure channel from the server to the client, can be easily retrieved by eavesdropping. V_1 can also be computed using the following equation:

$$GID_i = ID_i \oplus V_1$$

Similarly, GID_i is transmitted through the insecure channel from the client to the server and can be retrieved by the adversary through eavesdropping. As for ID_i , the adversary could acquire the information through other forms of attacks such as phishing, keylogging, malware, etc. With the computed session key, the adversary would be able to decipher the information transmitted via the secured communication channel using the compromised session key. Therefore, this protocol fails to achieve several evaluation criteria, including C4, C5, C7 and C8.

5.2.3 Vulnerable MFA Protocol 3 [98]. As highlighted in Section 3.5.1, Zahednejad et al. [98] proposed a two-factor authentication protocol for generic IoT devices in an M2M context. The protocol employs a long-term cryptographic key and historical data items as

authentication factors. To simulate real-life scenarios, they assumed a strong adversary capable of compromising the server and retrieving all the information within (as illustrated in Figure 3) and proved that the MFA protocol was able to ensure a secure authentication process and repel attacks. They also employed ROR model to formally prove how their protocol achieves perfect forward secrecy and key compromise resilience.

We perform a further analysis of the protocol to assess its security. As per the threat model, the adversary can only obtain $N - 1$ authentication factors and is able to eavesdrop any information passed through the public communication channel. Based on that assumption, it is possible to acquire the values of $R1$, $R2$, Y and TID_c , which are transmitted in plain through the public channel. The adversary would also be able to acquire the value of one of the authentication factors mk . Given that spk_s is the server's public key, it is publicly available and can be easily acquired by the adversary. With the acquired information, the adversary would be able to compute $r1$, $r2$, and X using the following equations:

- (1) $R1 = mk \oplus r1$;
- (2) $R1 = mk \oplus r1$;

$$(3) X = Y - H(r1||r2);$$

Using the acquired and computed values, the adversary can compute the value of the session key SK_s and decipher the information that is transmitted via the communication channel secured by the session key SK_s using the equation:

$$SK_s = H(mk|r1|r2|X|TID_c|spk_s)$$

Moreover, the construction of the authentication process in this protocol is entirely dependent on the first authentication factor (i.e., mk). Meaning, if the adversary manages to get mk, he can simply forge the whole authentication process. Therefore, this protocol does not even satisfy multi-factor authentication, and it fails several evaluation criteria, such as C3, C4, C5, C7 and C8.

5.2.4 Vulnerable MFA Protocol 4 [52]. As discussed in Section 3.5.5, Kumar et al. [52] proposed a three-factor authentication protocol for wireless sensor networks. It employs password, smart card, and biometric as authentication factors.

However, based on our analysis, this protocol also has several limitations. First of all, a strong adversary was not considered in their threat model. In the scenario whereby a strong adversary is able to intercept the information that passes through the open communication channel, i.e. ID_{sn}^i , N_{ur} , $h(RN_{sc})$, TS_1 and TS_5 , and obtain $N - 1$ authentication factors, i.e., user id and password, and smart card, the session key used by the entities can be computed. Using the data acquired, the adversary can compute the value of the session using the equation:

$$K_{ss} = h(ID_{ur}||ID_{sn}^i||U_{rg}^i|h(ID_{ur}||N_{ur})||h(RN_{sc})||TS_1||TS_5)$$

It is possible to infer that the value TS_5 is meant for a particular session key by checking the value of l_{10} , which is equal to $(h(K_{sh} \oplus U_{rg}^i)||ID_{ur})$. Based on Kerckhoffs's principle [68] and that hash values have a fixed length, the adversary is able to derive the length of $h(K_{sh} \oplus U_{rg}^i)$. Using this information, the adversary can extract the value of ID_{ur} by removing the bits belonging to the hash value and determine if TS_5 is meant for the affected user that is attempting to establish a secure communication channel, which results in security and anonymity breaches. Overall, this protocol fails several evaluation criteria, such as C4, C5, C6, C7 and C8.

5.2.5 Vulnerable MFA Protocol 5 [18]. As discussed in Section 3.5.2, Azrou et al. [18] designed a two-factor authentication protocol for healthcare IoT systems. The protocol was designed to address vulnerabilities of a prior work [77]. This protocol employs password and smart card as authentication factors to secure communications between healthcare systems in the cloud IoT. In addition to the informal security analysis, the authors claimed that they performed a formal security analysis using the Scyther tool and the ROM model to verify the security of their MFA protocol.

However, upon performing a security analysis, we discovered several flaws in this protocol. First of all, the authors did not provide a clear threat model nor indicated capabilities of the assumed adversaries. In addition, based on their formal and informal analysis, the authors claimed that they achieved perfect forward secrecy assuming that the values of x_s and MID are kept secret. However, our analysis of this protocol proved the other way round. MID is transmitted plainly through the public channel; therefore, it is not secret. In addition, a strong adversary would be capable of acquiring

the value of x_s from the cloud server through alternative means. Then, the value of w_i , which is used to compute the session key S_{key} , can be obtained using the following equation:

$$w_i = h(MID||x_s)$$

ID_{SN} is also transmitted plainly through the public channel, therefore it can be eavesdropped. Using the values w_i , MID and ID_{SN} , the adversary can compute the session key S_{key} using the equation:

$$S_{key} = h(w_i||MID||ID_{SN})$$

Therefore, this protocol fails several evaluation criteria, including C4, C5, C6, C7, and C8.

5.2.6 Vulnerable MFA Protocol 6 [13]. As discussed in Section 3.4, Ali et al. [13] proposed a three-factor symmetric key-based secure AKA protocol for Telecare Medicine Information Systems (TMIS). It was designed to address vulnerabilities of a prior work [20]. The protocol involves password, smart card and biometrics as its authentication factors. A revocation/re-register phase was also incorporated to enable the revocation of users in case of loss of a smart card. Through the informal and formal analysis, the authors demonstrated that the MFA protocol is resilient against key leakage, impersonation and known attacks, unlike several other authentication protocols.

However, our analysis proves that the proposed protocol fails to achieve several security properties, as discussed follows. The proposed MFA protocol computes its session key SK_{ij} as follows:

$$SK_{ij} = h(Y_{RC}||SID_j||T_3) == SK'_{ij} = h(Y_i||SID_j||T_3)$$

Through our analysis, the values of SID_j and T_3 can be intercepted from the open communication channel. It is also possible to compute the value Y_i using the following equation. $Y_i = h(SID_j||HID'_i||R_{rand2}||T_1)$, where:

- $HID'_i = h(PID_i)$ (acquire PID_i by compromising the first authentication factor);
- $R_{rand2} = R'_{rand2} \oplus ID_i$ (acquire R'_{rand2} by intercepting the message from the open communication channel);
- $T_1 = T'_1 \oplus HID'_i$ (acquire T'_1 by intercepting the message from the open communication channel);

Upon acquiring the required information as shown above, the adversary can compute the value of the session key SK_{ij} and decipher the information that is secured using the session key. The authors also assumed that the adversary cannot steal the private key of the registration center, which is a weak assumption as such keys can be leaked using side-channel or other techniques. Therefore, this protocol does not also achieve key leakage resilience (C4), perfect forward secrecy (C5), resilience against known attacks (C7), and strong adversarial assumption (C8).

5.2.7 Vulnerable MFA Protocol 7 [88]. As discussed in Section 3.1, Ertan et al. [88] proposed a two-factor authentication protocol for a generic client-server architecture. It aimed to alleviate the security posture of TOTP systems. It employs TOTP and PUF as its first and second authentication factors. The MFA protocol uses PUF as storage to securely store the client's secret. The authors assumed that the communication channel between the entities is secured using TLS.

Based on our analysis, it is observed that the paper did not explicitly demonstrate the complete authentication process, specifically mutual authentication, which is an important requirement in MFA protocols. The client sends the following messages to the server for the authentication:

- a) $M_1 = H(k_1, (t - t_0)/I) \oplus r$;
- b) $M_2 = H(k_2 \oplus r, (t - t_0)/I)$;
- c) c_1 , and;
- d) c_2 .

The server computes the value of $H(k_2 \oplus r, (t - t_0)/I)$ and verify if the computed value is the same as M_2 to authenticate the client. There is, however, no clear indication of verification performed by the client to determine the identity and authenticity of the server. Hence, it does not provide mutual authentication. Furthermore, the authors did not consider other critical security requirements in MFA, such as key leakage resilience, PFS, user anonymity, resilience against known attacks, and strong adversarial assumption. Therefore, it fails the evaluation criteria C1, C4, C5, C6, C7, and C8.

5.2.8 Vulnerable MFA Protocol 8 [73]. As discussed in Section 3.5.1, Sadhukhan et al. [73] proposed an ECC-based lightweight three-factor authentication protocol for remote users in a generic IoT network. It employs user identity, password and biometric data as its authentication factors. Through their informal security analysis, the authors claimed to have achieved several security goals, including resilience against impersonation and DoS attacks, PFS, and user anonymity, among others.

However, our further analysis on this protocol reveals that most of the claimed security goals are not properly achieved. For example, the authors claim of achieving PFS is just by assuming that the secret random number R_U used in the session key generation cannot be compromised by an attacker even if the pre-shared encryption/decryption symmetric key K_X is compromised by the attacker. This is a wrong assumption. If the attacker gets K_X (which is the underlying assumption in PFS), he can intercept the packet sent over the public channel and decrypt it to obtain R_U (see Fig.6 in [73]). Because, R_U is protected by only K_X . That is also how the gateway obtains R_U in their protocol. Therefore, this protocol does not achieve PFS.

In addition, the authors do not consider proper authentication factors as the user ID and biometric can be easily eavesdropped over the public communication channel. Moreover, the mutual authentication is based on the hash of the three factors, i.e., $H_U = h(ID_U || PW_U || B_U)$. This value (alongside R_U) is also encrypted using K_X , i.e., $E_{K_X}(H_U || R_U)$, and send to other parties over the public channel (see Figure A.1 or Fig.6 in [73]). That means, it is completely dependent on the pre-shared encryption/decryption key K_X (see the discussion about R_U above). In other words, the security of the protocol is entirely dependent on K_X , hence the authentication factors become meaningless. Therefore, this protocol does not achieve proper mutual authentication, distinctiveness of factors, and independence of factors.

Furthermore, the authors claim of user anonymity is by assuming that the user's identity, i.e., ID_u , is never communicated in plain. However, the user sends a message, consisting of the variables $ID_u, E_{k_x}(H_u || R_u), T_1$, to the IoT node over the public channel (as highlighted in Figure A.1). Since ID_u is sent in plain over the

open communication channel, an adversary can obtain it by eavesdropping the open communication channel. Hence, the protocol does not preserve the user's anonymity.

In addition, this protocol does not properly achieve resilience against key leakage and known attacks nor provided strong threat model. Overall, this protocol fails all of our security evaluation criteria, i.e., C1 through C8.

5.2.9 Vulnerable MFA Protocol 9 [37]. As highlighted in Section 3.4, Dhillon et al. [37] proposed a three-factor MFA and key exchange protocol for a healthcare system. In particular, it is designed to establish a secure communication channel between a medical professional and a remote patient monitoring in Cloud-IoT environments. The protocol involves a password, biometrics, and smart card as its authentication factors. The authors claimed to have achieved several security goals, such as mutual authentication, user anonymity, forward secrecy and many other known attacks.

However, our further analysis on the proposed protocol revealed several security flaws. For example, even though the protocol is supposed to use three authentication factors, credentials computed based on the user's password (1st factor) and biometric feature (2nd factor) are made to be stored in the smart card (3rd factor). This means, the two factors are dependent on the safety and security of the 3rd factor, failing our factors independence criteria C3.

In addition, the authors claim of achieving forward secrecy is infeasible. To achieve this, they are relying on the MP's identity (ID_{MP}), its private key X , and other parameters c and u . First of all, u is transmitted over a public channel (as seen in Fig.5 of [37]), which can be easily intercepted. Secondly, the assumption in forward secrecy is that the adversary can compromise long-term credentials of the client and server, hence the authors cannot rely on X . Third, c is computed based on the private key of the cloud server, which is the same issue as above. Fourth, the ID_{MP} is also stored both in the client and cloud server, which can be obtained in the same way or in several other means. In fact, ID_{MP} should not also be used for such purpose since it is not a secret factor. Therefore, this protocol fails the forward secrecy criteria C5.

Furthermore, there is no resilience against session key or long-term key leakage attacks or other known attacks. The authors also did not put a strong adversarial assumption that compromise identities and credentials of the user and cloud server. Overall, this protocol fails several security criteria, including C3, C4, C5, C7 and C8.

5.2.10 Vulnerable MFA Protocol 10 [61]. As discussed in Section 3.5.1, Melki et al. [61] proposed a lightweight ECC-based two-factor authentication protocol to provide secure communication channels between an IoT device and a gateway (see Figure A.6). The protocol uses a secret session identifier ID_s (that is derived from a PUF output value) and a secret channel-based parameter σ_i are used as authentication factors. The authors claimed to have achieved several security properties, such as PFS, session secrecy, user privacy, and resilience against reply, side-channel and man-in-the-middle attacks.

However, our analysis of the protocol reveals several flaws of the protocol. First of all, the assumed authentication factors, i.e., ID_s and σ_i , are not appropriate factors since such features are not reliable and prone to several security problems. These factors

are not also distinct as both belong to the family of possession factors. In addition, the authors' assumption that "the adversary's probability of obtaining ID_s is exceptionally low" is unrealistic and it diminishes the principal threat assumption of MFA, in which the attacker can achieve $N - 1$ authentication factors. In fact, there are several ways to obtain ID_s , e.g., via side-channel techniques as it is stored in the client and server devices.

Furthermore, the proposed protocol is susceptible to both client and server impersonation attacks. Hence, an adversary can potentially obtain the list of ID_s from the gateway. In addition, the adversary can plant itself in between the IoT device and the gateway and establish separate communication channels with either of them. In that case, the IoT device would send $Message_1 :< M_1, M_2, TS_A >$ to the adversary instead of the gateway. The adversary will attempt to retrieve the values of R_A and τ_i by computing $M_3^a = h(ID_s || TS_A)$, $R_A = M_3^a \oplus M_1$, $\tau_i = M_2 \oplus R_A$. The adversary will also compute $\sigma_i' = Rep(N_{0,A^a}, \tau_i)$ where N_{0,A^a} is the channel-based nonce between the IoT device and adversary. The adversary then computes $Message_1^a$ containing $< M_1, M_2^a, TS_A >$ where $M_2^a = R_A \oplus \tau_i^a$. Here, τ_i^a is computed using the equation $Gen(N_{0,B^a}) = (\sigma_i^a, \tau_i^a)$ where N_{0,B^a} is the channel-based nonce between the adversary and gateway. Upon receiving $Message_1^a$, the gateway can compute the following values and send them back to the adversary as $Message_2 :< M_4, M_5, TS_B >$ where:

- a) $SK = h(ID_s || TS_A || TS_B || R_A || R_B || \sigma_i^a)$, where $\sigma_i^a = Rep(N_{0,B^a}, \tau_i^a)$;
- b) $M_4 = h(ID_s || TS_B || TS_A || R_A) \oplus R_B$, and;
- c) $M_5 = h(SK \oplus R_A \oplus R_B)$.

The adversary will then send $Message_2^a :< M_4, M_5^a, TS_B >$ to the IoT device where $M_5^a = h(SK_a \oplus R_A \oplus R_B)$ and $SK_a = h(ID_s || TS_A || TS_B || R_A || R_B || \sigma_i^a)$. Upon receiving $Message_2^a$, the IoT device will compute the following values and sent back to the adversary as $Message_3 :< M_8, TS_{A'} >$ where $M_8 = h(SK_a' || ID_s)$. Lastly, the adversary will send $Message_3^a :< M_8^a, TS_{A'} >$ where $M_8^a = h(SK' || ID_s)$ to the gateway. With the above actions performed, the adversary can successfully plant itself in between the IoT device and gateway to intercept all the transmitted information.

Furthermore, the authors claimed to have achieved PFS just by considering the one-way property of a hash function and the secrecy of the session identifier ID_s . This is also an unrealistic claim. Because, the ID_s is not properly secure (as discussed above) and the other parameters (e.g., timestamp) are transmitted over the public channel (see Figure A.6). More importantly, the authors' assumption of PFS is different from the actual definition. In reality, PFS is a critical security requirement in key exchange which requires that past session keys remain secure even if the long-term credentials of both client and server are compromised. Therefore, this protocol does not achieve PFS in many ways.

In addition, the authors excluded a malware embedded in one of the communicating devices and an adversary present in the same subnet (as it can obtain channel-based parameters) from their threat assumption. These are unrealistic assumptions, which makes their adversarial model weak. Overall, this protocol fails to achieve any of our evaluation criteria, i.e., C1 through C8.

5.3 Summary of the security analysis

The analysis performed demonstrated that not all MFA protocols are without weaknesses or flaws. Table 2 summarizes results of our evaluation on the ten protocols using our evaluation criteria. One of the key insights drawn from the analysis is that having more authentication factors does not necessarily translate to better security. As shown in the above analysis, most of the MFA protocols do not consider the critical security requirements discussed in our evaluation criteria.

Some MFA protocols were not designed with mutual authentication in mind [16]. With the increase in scams [33], it is no longer surprising that individuals require assurance that the entities they are interacting with are trusted and legitimate. Organizations verifying the authenticity of individuals alone are no longer sufficient, as they would need to prove their identity to the individuals interacting with them too.

As shown in the above analysis, employing interdependent multiple authentication factors will negatively affect the security of the MFA protocol it is originally supposed to provide. Compromise in one of the authentication factors may result in a cascading effect leading to the compromise of the other authentication factors.

Privacy and anonymity are two features that people increasingly demand as they become more aware of the importance of safeguarding their personal information and identity. Moreover, there are privacy regulations [36, 40, 78] that organizations must comply with. Secure generation of session keys that are only accessible by authorized parties, therefore, becomes one of the key requirements [52], which seems to be lacking in some of the investigated MFA protocols. A combined attack such as MTIM attack to obtain information transmitted through the open communication channels and other attacks e.g., smart card loss attack, side-channel attack, or malware infection resulting in the loss of one authentication factors, a strong adversary is able to repeatedly compute the session key generated for each session and use it to decipher the sensitive information protected by the session key.

Lastly, it is imperative to accurately depict real-life threat scenarios. The capabilities of adversaries and all possible entry points that could be targeted by them must be considered in the design of a robust, efficient, and secure MFA protocol. The weaknesses and flaws identified through this analysis is a clear indication of fundamental but critical security requirements that should have been considered when designing MFA protocols.

5.4 Comparative performance analysis

Although most MFA protocols mainly focus on their security guarantee, their performance overhead (or the security-efficiency trade-off) should not be neglected. In fact, some protocols employ too many authentication factors without considering the penalty in performance. However, performance is usually equally critical as security, especially in resource and real-time constrained systems, such as IoT and CPS. Therefore, it is essential to assess the performance of MFA protocols as well. To do such assessment, researchers use different types of performance metrics. The most commonly used ones are computation cost (i.e., the amount of resources required to execute the authentication protocol), communication bits (i.e., the amount of data required to be transmitted between entities),

Table 2: Security evaluation results of the 10 MFA protocols

Protocol	Domain	Authentication Factors	C1	C2	C3	C4	C5	C6	C7	C8
Protocol 1 [48] (π_{woFS})	IIoT	LSK + TGK ^C + HD ^S + HDT ^S	×	×	×	×	×	×	✓	WA
Protocol 1 [48] (π_{FS})	IIoT	LSK + TGK ^C + HD ^S + HDT ^S	×	×	×	×	✓	×	✓	WA
Protocol 2 [41]	Generic IoT	PW + SC + BD	✓	✓	✓	×	×	✓	×	WA
Protocol 3 [98]	Generic IoT	LSK + HD	✓	✓	×	×	×	✓	×	SA
Protocol 4 [52]	WSN	PW + SC + BD	✓	✓	✓	×	×	×	×	WA
Protocol 5 [18]	Healthcare IoT	PW + SC	✓	✓	✓	×	×	✓	×	WA
Protocol 6 [13]	Healthcare	PW + SC + BD	✓	✓	✓	×	×	✓	×	WA
Protocol 7 [88]	Client-Server	TOTP + PUF	×	✓	✓	×	×	×	×	WA
Protocol 8 [73]	Generic IoT	UID + PW + BD	×	×	×	×	×	×	×	WA
Protocol 9 [37]	Healthcare	PW + SC + BD	✓	✓	×	×	×	✓	×	WA
Protocol 10 [61]	Generic IoT	SSID + SCP	×	×	×	×	×	×	×	WA

Description of notations: LSK: Long-term Shared Key, TGK^C: Tag Generation Key, HD: Historical Data, HDT: Historical Data Tags, X^C : factor X is only for Client, X^S : factor X is only for Server, PW: Password, SC: Smart Card, BD: Biometric Data, TOTP: Time-based Ontime Password, PUF: Physical Unclonable Function, UID: User Identity, SSID: Secret Session Identifier, SCP: Secret Channel-based Parameter, WA: Weak Adversary, SA: Strong Adversary

communication passes (i.e., the number of messages exchanged between the client and the server to complete the authentication process), authentication time (i.e., the time taken to complete the authentication process), and the storage cost (i.e., the storage size required to complete the authentication process). Therefore, we also used these metrics to perform a comparative performance analysis of the ten protocols investigated. The overall performance analysis is summarized in Table 3. We believe that this helps future researchers to easily obtain the performance information of the protocols in a single reference point.

6 RECOMMENDED MITIGATION STRATEGIES

In Section 5, we demonstrated that most MFA protocols are not without weaknesses and flaws. Adversaries can potentially target such weaknesses to gain unauthorised access to systems and applications. In this section, we discuss a wide range of mitigation strategies that can potentially address the identified weaknesses and flaws and minimize the risk exposure of the systems and applications. Some of the strategies are our suggestions based on our perspectives while some others are best practices that we compiled from existing works.

6.1 Mutual authentication

Mutual authentication can be achieved in many ways. One of the most commonly adopted approaches is using digital certificates in which users authenticate each other through their respective CA-issued public keys. However, due to its high performance overhead, it is not the ideal solution for most resource constrained systems, such as IoT. An ideal approach to address this problem is a cryptographic construction where both parties can present provable credentials or factors to prove their identity. One such approach is properly utilizing shared cryptographic keys in a challenge-response manner. For example, the server can compute the response to a

random nonce with the shared cryptographic key and compare it with the response computed by the client. If the response matches, the client is authenticated. Similarly, the client will compute the response to another random nonce with the shared cryptographic key and compare it with the response computed by the server. The server is authenticated if the responses match. Kim et al. [51] (as demonstrated in Figures A.4 and A.2) used such approach to ensure mutual authentication between entities. It was achieved by verifying messages that were computed using shared secret parameters distributed during the registration phase. Using the shared secret parameter S_i^1 , the user and the gateway will mutually authenticate each other by verifying the correctness of messages GM_5 and U_iM_8 , respectively.

Oh et al. [63] also adopted a similar approach (as demonstrated in Figures A.5 and A.9) to achieve mutual authentication. The shared secret keys K_{MUG} and K_{GSD} stored in the Home Gateway (HGW) are used to verify the identity of the user and smart device, respectively. The user and smart device will also utilise the same keys to verify the identity of the HGW to achieve mutual authentication.

Another interesting approach to achieve mutual authentication is by proving a possession of a piece of data that allows both parties to authenticate each other. For example, Zahednejad et al. [98] utilized historical data that were shared between the parties, which allows them to prove each other’s identity in a challenge-response manner.

For higher assurance, a more rigorous approach can be adopted using formal methods such as BAN logic [23, 91] that several researchers used to evaluate the security of their MFA protocols [30, 50, 63, 65, 90, 99], to further prove mutual authentication.

6.2 Distinctiveness of authentication factors

For an MFA protocol to be fully effective, one of the key design criteria should be distinctiveness of its authentication factors. That means, the protocol should involve distinct authentication factors

Table 3: Performance comparison of the 10 MFA protocols

Protocol	Computation Cost	Communication Bits	Communication Passes	Time Taken (ms)	Storage Required
Protocol 1 [48] (π_{woFS})	$326T_h + 1T_{aed}$	3992	2	22.39	-
Protocol 1 [48] (π_{FS})	$328T_h + 4T_{ecc} + 1T_{aed}$	4748	3	115.549 [#]	-
Protocol 2 [41]	$18T_h + 14T_x + 2T_{fe} + 2T_{ecc}$	1024	1	198.21	-
Protocol 3 [98]	$4T_{me} + (2z+3)T_m + (2z)T_a + (2z+26)T_h$	2720	4	11.28	3.4GB + 252B
Protocol 4 [52]	$26T_h + 2T_{ed}$	2000	3	-	-
Protocol 5 [18]	$17T_h\#$	1312*	3	-	-
Protocol 6 [13]	$15T_h + 1T_{fe} + 3T_{ed}$	2144	2	8.9385	-
Protocol 7 [88]	$4T_h + 2T_p$	640	1	-	404 GB
Protocol 8 [73]	$5T_h + 2T_{ecc} + 8T_{ed}$	1600	3	80.6	480 bits
Protocol 9 [37]	$1T_{me} + 7T_h$	448*	1	-	-
Protocol 10 [61]	$10T_h + 10T_x + 1T_{fe}$	896	2	-	-

Description of notations: T_h : hash operation, T_{ecc} : ECC multiplication, T_{ed} : symmetric encryption decryption, T_{aed} : SLHAE encryption and decryption, T_x : XOR operation, T_{fe} : fuzzy extraction operation, T_p : PUF computation, T_{me} : modular exponential operation, T_a : addition operation, T_m : multiplication operation, # : XOR operations ignored, * : Estimated value, -: no information provided.

that are drawn from different categories, such as knowledge, possession, inherent, and location factors. This is because, the authentication factors of the same category often have a relatively similar security level. Hence, if the adversary manages to break one authentication factor, he can do so on others by applying a similar technique, as seen in [64]. For example, Jin et al. [64] (see the discussion in Section 5.2.1), used a preshared symmetric key and a tag generation symmetric key as first and second authentication factors for the client, respectively. Here, both are possession factors, and an adversary may apply a similar technique to break them. Therefore, ensuring the distinctiveness of the authentication factors is an essential strategy that MFA protocol designers should follow.

6.3 Independence of authentication factors

The generation of an authentication factor must be unique and independent from others to ensure that the compromise of one authentication factor does not result in the compromise of other authentication factors. For example, an authentication factor must not be derived from other authentication factors. In addition, an authentication factor must not be used to protect the confidentiality of another authentication factor.

By combining multiple independent authentication factors, an adversary must acquire the different authentication factors in order to gain access to the targeted system. For example, a three-factor authentication protocol should utilise three independent authentication factors where the compromise of one should not affect other. This improves the overall security of the MFA protocol and the underlying systems. Lee et al. [54] demonstrated the independence of authentication factors in its MFA protocol and compromise to any of the authentication factors will not result in further compromise to other authentication factors. Vinoth et al. [69] also demonstrated in their MFA protocol that in the event whereby $N - 1$ authentication factors are compromised, the remaining authentication factor will stay secret, and the overall authentication process will not be

compromised. A snapshot of the approach and authentication steps of Vinoth et al. [69] is provided in Figure A.8.

6.4 Key leakage resilience

Ensuring key leakage resilience is a sophisticated and critical task in the design of an MFA protocol and has to be extensively deliberated to address every possible key leakage scenario. There is no one-size-fits-all design to achieve key leakage resilience and every MFA protocol would require a tailored approach based on the requirements of the systems and the potential threats the system is exposed to. However, there are some guiding principles that can be adhered to achieve the objective.

Given that the initial authentication process is typically conducted through open communication channels, researchers should ensure that not all parameters utilised in the session key generation are transmitted in clear. In addition, the parameters used in the generation of session keys should not be easily computed using information obtained through the open communication channels and compromise of the authentication factors. One example is the MFA protocol designed by Kwon et al. [53], which is illustrated in Figure A.7. In this protocol, the equation used to compute the session key is $SK = h(h(N_2||HID_i)||N_3||N_1)$. As shown in Figure A.7, none of the parameters used in the computation of the session were transmitted in plain through the open communication channel. To obtain the random nonces N_1 , N_2 and N_3 , the adversary would either need to compromise all three components i.e., User, Gateway, and Sensor Node, or compromise other secret keys or both authentication factors to obtain parameters that are used to protect the random nonces.

Although not specifically discussed in the papers relating to MFA protocols, attaining a high entropy of the cryptographic keys and random nonces, and proper key management is equally important in achieving the objective the MFA protocol aims to deliver. High entropy reduces the likelihood of adversaries from acquiring the sensitive information through guessing or brute-force attacks that

directly or indirectly results in the compromise of the session keys [4]. Key security is not only about generation, but comprises of other aspects such as distribution, storage, usage, rotation, and disposal. Therefore, proper key management adhering to guidelines by reputable organisations such as NIST [4–6] would provide a higher assurance in the overall security of the authentication process. With technological evolution, cryptographic key lengths once thought to be secure may become vulnerable. Therefore, it is important to keep current with the latest cryptographic standards recommended [32], and regularly review the security of the MFA protocols.

6.5 User Anonymity

Ensuring anonymity can be achieved through the use of anonymous credentials, such as issuing the user with a pseudo identity that will not reveal their actual identity. For example, in the registration phase of the MFA protocol by Kim et al. [51] shown in Figure A.4, the Gateway generates a pseudo identity TID_i using the values of the user’s actual identity ID_i , the Gateway’s secret key K_{GW} , and a randomly generated nonce R_{GW}^1 . The pseudo identity TID_i instead of the actual identity ID_i is then used in the authentication process. In addition, the pseudo identity TID_i is refreshed every time with a new pseudo identity TID_i^{new} during the authentication process. Alternatively, cryptographic techniques can be employed to protect the anonymity of individuals during the authentication process. These include encryption, hashing, and other mathematical operations such as concatenation, XOR, etc. In the proposed MFA protocol by Bouchaala et al. [22], the actual identity of user ID_u is never transmitted in plain. Instead, cryptographic techniques were employed to generate PID_u , which is used in the authentication process, with the following equation:

$$PID_u = (ID_u || C_1) \oplus h(B_1 || B_3)$$

As the value of B_3 changes every time, the PID_u value will also be different for every authentication session, making it hard to determine the actual identity of the user. Similarly, in the design by Lee et al. [54], cryptographic techniques were employed to achieve anonymity. The actual identity ID_u , as shown in Figure A.3, is not transmitted in plain through the open communication channel. Instead, ID_u , together with PWB_i , were used to derive UID_u and used for the authentication process. UID_u is obfuscated further using cryptographic techniques before it is transmitted to the Cloud Server.

6.6 Resistance to known attacks

There are several possible approaches to resist known attacks and actual security mitigation strategies are dependent to specific requirements and constraints present. The following paragraphs will focus on the weaknesses and flaws identified in the protocols analysed in Section 5 and provide examples on the mitigating strategies that can be applied. It is typically not feasible to communicate via encrypted channel during the initial authentication process. The client and server are required to have prior trust established and the required information to establish the secure communication channel which is one of the key objectives of the authentication process. Therefore, appropriate assumption had to be accorded that the MFA protocol is unavoidably susceptible to interception

during the initial authentication process, regardless if the adversaries are weak or strong. Mitigation strategies discussed in Section 6.3 are applicable in mitigating the potential impacts of client and server impersonation attacks. Researchers have to ensure that only non-sensitive information that does not directly or indirectly affects the security of the MFA protocols is transmitted through the open communication channel. Oh et al. [63] demonstrated how impersonation attack is mitigated in their design. Even if the adversary can successfully compromise the first authentication factor i.e. mobile device to obtain the credentials $A_1, A_2, A_3, A_4, PID_{MU}$ and intercept the information i.e., $PID_{MU}, M_1, C_1, V_{MU}$ transmitted through the open communication channel, the adversary will not be succeed in impersonating the user as the adversary does not possess information relating to the second authentication factor that is necessary to generate a spoofed authentication request that is valid. Similarly, although the adversary can intercept the information $PID_{MU}, M_3, C_2, V_{MUG}$ and M_5, V_{GSD} , the adversary is unable to generate a spoofed response that is valid without information relating to the second authentication factor. In addition, in both scenarios, the adversary is unable to obtain the random nonces i.e., RN_{MU} and RN_G which are required in the computation of a valid request and response. From the analysis, time-based nonces i.e., timestamps and/or randomly generated nonces were commonly employed to resist replay attacks. However, it is worthy to note that the utilisation of time-based nonces may potentially result in other issue such as DoS attack. The entities involve in the authentication process must have their clocks synchronised in order for authentication process to work effectively. In the event of a significant time difference or clock drift between the entities, it can lead to authentication failures, resulting in DoS. It is therefore recommended to adopt the use of randomly generated nonces to eliminate the need to depend on external factors, in this case time synchronisation. As demonstrated in [63] and shown in Figure A.5, randomly generated nonces are employed to ensure the “freshness” of the authentication request and is able to resist replay attacks. However, if time-based nonces must be adopted for any reasons, additional control measures shall be implemented to ensure accuracy in time synchronisation to minimize the occurrence of time discrepancies.

6.7 Adversary assumption

Assumption of the adversary’s capability is vital in determining the necessary security mitigation strategies required to satisfy the security requirements and reduce the likelihood of the risks occurring to the lowest possible. As indicated by Zahednejad et al. [98], assuming a weak adversary that is only capable of obtaining a small portion of information may not truly reflect the actual reality. It is therefore important to adopt a more realistic approach and accord the adversary with the necessary respect so that security mitigation strategies can designed and implemented to successfully thwart the attacks.

6.8 Summary

Table 4 provides a summary of the possible mitigation strategies that can be applied to mitigate the weaknesses and flaws identified in Section 5. As shown, the mitigation strategies are nothing out

of the ordinary and have been implemented in the design of MFA protocols by other researchers [22, 51, 53, 54, 63, 69, 98].

7 CONCLUSION

With the focus on authentication, several MFA protocols specifically tailored for the various domains have been developed in recent years. However, common but yet critical security criteria that should have been considered and applied were observed to be omitted in some of the proposed MFA protocols. In some cases, claims of the MFA protocols were capable of satisfying certain security criteria were proved otherwise. In this work, we reviewed several MFA protocols and analyzed potential vulnerabilities heuristically. In particular, we systematically analyzed security flaws in the construction of the protocols using a set of security evaluation criteria we employed.

Consequently, we managed to identify several vulnerabilities in ten of the MFA protocols. We provided a detailed discussion of the vulnerabilities identified. We also highlighted relevant mitigation strategies for those vulnerabilities. We believe that the consolidated information would provide a single reference point for researchers to be aware of the potential security issues that require attention and apply the necessary mitigation strategies when designing MFA protocols. It is also worth noting the importance of performance alongside its security. The performance of the MFA protocol should not be overlooked. Complementing it with an efficient performance will improve the adoption rate of the secure MFA protocol, thus enhancing the overall security.

To further strengthen the design and implementation of an MFA protocol, a security-by-design approach [7] should be considered. In the future, this work can be further extended by employing a formal analysis of the protocols. Emerging security concerns, such as the security risk posed by the advent of quantum computing, can also be included as additional evaluation criteria.

ACKNOWLEDGMENT

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative (DTC-T2FI-CFP-0002). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority.

REFERENCES

- [1] 2019. NOTICE ON CYBER HYGIENE. <https://www.mas.gov.sg/-/media/mas/notices/pdf/psn06-notice-on-cyber-hygiene.pdf>
- [2] 2022. Multi-factor Authentication Fact Sheet. <https://www.cisa.gov/sites/default/files/publications/MFA-Fact-Sheet-Jan22-508.pdf>
- [3] 2023. Importance of Using Secure Multi-Factor Authentication Methods — csa.gov.sg. <https://www.csa.gov.sg/alerts-advisories/Advisories/2023/ad-2023-006>. [Accessed 22-08-2023].
- [4] n.d.. Recommendation for key management Part 1 - NIST. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf>.
- [5] n.d.. Recommendation for key management Part 2. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt2r1.pdf>.
- [6] n.d.. Recommendation for key management Part 3 - NIST. <https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-57pt3r1.pdf>.
- [7] n.d.. Security by Design Framework - Cyber Security Agency of Singapore. https://www.csa.gov.sg/docs/default-source/csa/documents/legislation_supplementary_references/security_by_design_framework.pdf?sfvrsn=560b9f3_0.
- [8] Nuku Abiew, Maxwell Dorgbefu Jnr, and Samuel Banning. 2020. Design and Implementation of Cost Effective Multi-factor Authentication Framework for ATM Systems. *Asian Journal of Research in Computer Science* (04 2020), 7–20. <https://doi.org/10.9734/ajrcos/2020/v5i330135>
- [9] AlsharifHasan Mohamad Aburbeian and Manuel Fernández-Veiga. 2024. Secure Internet Financial Transactions: A Framework Integrating Multi-Factor Authentication and Machine Learning. *AI* 5, 1 (2024), 177–194.
- [10] Fardin Ahmadi, Sonia Garg, Gaurav Gupta, Preety Baglat, Puja Thakur, and Syed Zahra. 2021. Multi-factor Biometric Authentication Approach for Fog Computing to ensure Security Perspective. <https://doi.org/10.1109/INDIACom51348.2021.00031>
- [11] Alawi Al-Saggaf, Tarek Sheltami, Hoda Alkhzaimi, and Gamil Ahmed. 2022. Lightweight Two-Factor-Based User Authentication Protocol for IoT-Enabled Healthcare Ecosystem in Quantum Computing. *Arabian Journal for Science and Engineering* 48 (09 2022), 1–11. <https://doi.org/10.1007/s13369-022-07235-0>
- [12] Ahmed Aleroud and Lina Zhou. 2017. Phishing Environments, Techniques, and Countermeasures: A Survey. *Computers & Security* 68 (04 2017). <https://doi.org/10.1016/j.cose.2017.04.006>
- [13] Zeeshan Ali, Sajid Hussain, Rana Rehman, Asmaa Munshi, Misbah Liaqat, Neeraj Kumar, and Shehzad Chaudhry. 2020. ITSSAKA-MS: An improved three-factor symmetric-key based secure AKA scheme for multi-server environments. *IEEE Access* PP (06 2020), 1–1. <https://doi.org/10.1109/ACCESS.2020.3000716>
- [14] Mohammed Alkathairi, Sajid Saleem, Mohammed Alqarni, Ahmad Aseeri, Sajjad Chaudhary, and Yu Zhuang. 2022. A Lightweight Authentication Scheme for a Network of Unmanned Aerial Vehicles (UAVs) by Using Physical Unclonable Functions. *Electronics* 11 (09 2022), 2921. <https://doi.org/10.3390/electronics11182921>
- [15] Esra Altulaihan, Mohammed Amin Almaiah, and Ahmed Aljughaiman. 2022. Cybersecurity threats, countermeasures and mitigation techniques on the IoT: Future research directions. *Electronics* 11, 20 (2022), 3330.
- [16] Ruhul Amin, SK Islam, P. Vijayakumar, Khurram Khan, and Victor Chang. 2018. A robust and efficient bilinear pairing based mutual authentication and session key verification over insecure communication. *Multimedia Tools and Applications* 77 (2018). <https://doi.org/10.1007/s11042-017-4996-z>
- [17] Wei Heng Ang, Huaqun Guo, and Eyasu Getahun Chekole. 2023. VulnGen: Vulnerable Virtual Machine Generator. In *2023 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. IEEE, 1–8.
- [18] Mourade Azroul, Jamal Mabrouki, and Rajasekhar Chaganti. 2021. New Efficient and Secured Authentication Protocol for Remote Healthcare Systems in Cloud-IoT. *Security and Communication Networks* 2021 (05 2021), 1–12. <https://doi.org/10.1155/2021/5546334>
- [19] Mohammadreza Barkadehi, Mehrbaksh Nilashi, Othman Ibrahim, Ali Fardi, and Sarminah Samad. 2018. Authentication Systems: A Literature Review and Classification. *Telematics and Informatics* 35 (03 2018). <https://doi.org/10.1016/j.tele.2018.03.018>
- [20] S. Barman, H. P. H. Shum, S. Chattopadhyay, and D. Samanta. 2019. A secure authentication protocol for multi-server-based E-healthcare using a fuzzy commitment scheme. *IEEE Access* 7 (2019), 12557–12574.
- [21] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. 1998. Modular approach to the design and analysis of authentication and key exchange protocols. In *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*.
- [22] Mariem Bouchaala, Cherif Ghazel, and Leila Saidane. 2022. Enhancing security and efficiency in cloud computing authentication and key agreement scheme based on smart card. *The Journal of Supercomputing* 78 (01 2022), 1–26. <https://doi.org/10.1007/s11227-021-03857-7>
- [23] Michael Burrows, Martin Abadi, and Roger Needham. 1990. A logic of authentication. *ACM Trans. Comput. Syst.* 8, 1 (February 1990), 18–36. <https://doi.org/10.1145/77648.77649>
- [24] Aldar Chan, Jun Wen, Jianying Zhou, and Joseph Teo. 2016. Scalable Two-Factor Authentication Using Historical Data, Vol. 9878. 91–110. https://doi.org/10.1007/978-3-319-45744-4_5
- [25] A. C.-F. Chan, J. W. Wong, J. Zhou, and J. Teo. 2016. Scalable two-factor authentication using historical data. In *Proc. ESORICS*, Vol. 9878. Springer, 91–110.
- [26] Eyasu Getahun Chekole, Sudipta Chattopadhyay, Martín Ochoa, Huaqun Guo, and Unnikrishnan Cheramangalath. 2020. Cima: Compiler-enforced resilience against memory safety attacks in cyber-physical systems. *Computers & Security* 94 (2020), 101832.
- [27] Eyasu Getahun Chekole, Martín Ochoa, and Sudipta Chattopadhyay. 2021. Scope: Secure compiling of plcs in cyber-physical systems. *International Journal of Critical Infrastructure Protection* 33 (2021), 100431.
- [28] Rui Chen, Yongcong Mou, and Min Zhang. 2022. A novel three-factor authentication scheme with high security for multi-server environments. *Wireless Personal Communications* 124, 1 (2022), 763–781.
- [29] Xiao Chen, BaoCheng Wang, and Haibin Li. 2024. A privacy-preserving multi-factor authentication scheme for cloud-assisted IoMT with post-quantum security. *Journal of Information Security and Applications* 81 (2024), 103708.
- [30] Zigang Chen, Zhiquan Cheng, Wenjun Luo, Jin Ao, Yuhong Liu, Kai Sheng, and Long Chen. 2023. FSMFA: Efficient firmware-secure multi-factor authentication

Table 4: Summary of possible mitigation strategies

Security Evaluation Criteria	Possible Mitigation Strategies
Mutual Authentication	<ul style="list-style-type: none"> – Both parties presenting appropriately provable credentials or factors to each other to prove their identity. – Cryptographic computations and verifications using shared secret keys. – Perform security analysis using methods such as BAN logic to formally prove the security properties.
Distinctiveness of factors	<ul style="list-style-type: none"> – Ensure that the authentication factors are from distinct categories.
Independence of factors	<ul style="list-style-type: none"> – Generation of authentication factor must be unique and cannot be derived from the knowledge of another authentication factor. – One authentication factor must not be used to protect the confidentiality of another authentication factor.
Key Leak Resilience	<ul style="list-style-type: none"> – Do not transmit all parameters utilized in the session key generation in clear. – Parameters utilized in the generation of session keys should not be easily derived from the information transmitted in clear. – Achieve high entropy of the cryptographic keys and random nonces. – Perform proper key management in line with guidelines by reputable organizations such as NIST.
Perfect-forward secrecy	<ul style="list-style-type: none"> – Construct the MFA protocol (including its authentication factors) in such a way that ensures the leakage of long-term keys of both client and server cannot compromise the security of previous sessions. In other words, previous session keys should not be computed from long-term keys of the client and the server.
User Anonymity	<ul style="list-style-type: none"> – Employ strong user privacy-preserving approaches (even beyond the use of pseudonymous identities). – Utilize cryptographic techniques to protect the anonymity of individuals.
Resistance to Known Attacks i.e., impersonation and DoS attacks	<ul style="list-style-type: none"> – Do not transmit sensitive information in clear that directly or indirectly affects the security of MFA protocols. – Protect parameters such as random nonces using cryptographic techniques and ensure that compromise to $N - 1$ authentication factors would not affect the security of the MFA protocols.
Adversary Assumption	<ul style="list-style-type: none"> – Adopt a realistic approach and accord the adversary with appropriate capabilities.

protocol for IoT devices. *Internet of Things* 21 (04 2023), 100685. <https://doi.org/10.1016/j.iot.2023.100685>

[31] Iwaylo Chenchev. 2023. Framework for Multi-factor Authentication with Dynamically Generated Passwords. In *Future of Information and Communication Conference*. Springer, 563–576.

[32] Computer Security Division, I. T. L. n.d.. Cryptographic standards and guidelines: CSRC. <https://csrc.nist.gov/Projects/Cryptographic-Standards-and-Guidelines>.

[33] Cue. 2023. Scams to watch out for in 2023. *The Straits Times*. <https://www.straitstimes.com/business/invest/scams-to-watch-out-for-in-2023>.

[34] Jie Cui, Fangzheng Cheng, Hong Zhong, Qingyang Zhang, Chengjie Gu, and Lu Liu. 2022. Multi-factor based session secret key agreement for the Industrial Internet of Things. *Ad Hoc Networks* 138 (09 2022), 102997. <https://doi.org/10.1016/j.adhoc.2022.102997>

[35] Aleksandar Sandro Cvetković, Vesna Radojčić, and Saša Adamović. 2021. Multi-factor Authentication for the Internet of Things. *ZBORNIK RADOVA UNIVERZITETA SINERGIIJA* 22. <https://doi.org/10.7251/ZRSNG2101013C>

[36] Default. 2023. Importance of using secure multi-factor authentication methods. <https://www.csa.gov.sg/alerts-advisories/Advisories/2023/ad-2023-006>. Accessed on: [insert date].

[37] Parwinder Dhillon and Sheetal Kalra. 2018. Multi-factor user authentication scheme for IoT-based healthcare services. *Journal of Reliable Intelligent Environments* 4 (2018). <https://doi.org/10.1007/s40860-018-0062-5>

[38] Prabakaran Durairaj and Shyamala Ramachandran. 2021. Multi-Factor Authentication for Secured Financial Transactions in Cloud Environment. *Computers, Materials and Continua* 70 (09 2021), 1781–1798. <https://doi.org/10.32604/cm.2022.019591>

[39] Blaise Gassend, Dwaine Clarke, Marten van Dijk, and Srinivas Devadas. 2002. Silicon physical random functions. In *Proceedings of the 9th ACM conference on Computer and communications security (CCS '02)*. Association for Computing Machinery, New York, NY, USA, 148–160. <https://doi.org/10.1145/586110.586132>

[40] GDPR.eu. n.d.. General Data Protection Regulation (GDPR) compliance guidelines. <https://gdpr.eu/>.

[41] Amir Hossein Ghafouri Mirsarai, Ali Barati, and Hamid Barati. 2022. A secure three-factor authentication scheme for IoT environments. *J. Parallel and Distrib. Comput.* 169 (11 2022), 87–105. <https://doi.org/10.1016/j.jpdc.2022.06.011>

[42] Yiran Han, Hua Guo, Jianwei Liu, Brou Bernard Ehui, Yapeng Wu, and Sijia Li. 2024. An Enhanced Multifactor Authentication and Key Agreement Protocol in Industrial Internet of Things. *IEEE Internet of Things Journal* 11, 9 (2024), 16243–16254. <https://doi.org/10.1109/JIOT.2024.3355228>

[43] Md Hassan and Zarina Shukur. 2021. A Secure Multi Factor User Authentication Framework for Electronic Payment System. 1–6. <https://doi.org/10.1109/CRC50527.2021.9392564>

[44] Christopher J Hegarty. 2017. The global positioning system (GPS). *Springer handbook of global navigation satellite systems* (2017), 197–218.

[45] The White House. 2021. Executive Order on Improving the Nation’s Cybersecurity. <https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/>

[46] Mamoona Humayun, Mahmood Niazi, NZ Jhanjhi, Mohammad Alshayeb, and Sajjad Mahmood. 2020. Cyber security threats and vulnerabilities: a systematic mapping study. *Arabian Journal for Science and Engineering* 45 (2020), 3171–3189.

[47] A Jardine, TD Ramotsoela, and Gerhard P Hancke. 2021. Biometric authentication system for industrial applications using iris recognition. In *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*. IEEE, 01–06.

[48] Chenglu Jin, Zheng Yang, Tao Xiang, Sridhar Adepu, and Jianying Zhou. 2023. HMACCE: Establishing Authenticated and Confidential Channel From Historical Data for Industrial Internet of Things. *IEEE Transactions on Information Forensics and Security* 18 (2023), 1080–1094. <https://doi.org/10.1109/TIFS.2023.3234873>

[49] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. 2015. {Sound-Proof}: Usable {Two-Factor} authentication based on ambient sound. In *24th USENIX security symposium (USENIX security 15)*. 483–498.

[50] Haqi Khalid, Shaiful Hashim, Sharifah Mumtazah, Fazirulhisyam Hashim, and Muhammad Chaudhary. 2021. SELAMAT: A New Secure and Lightweight Multi-Factor Authentication Scheme for Cross-Platform Industrial IoT Systems. *Sensors* 21 (02 2021), 1428. <https://doi.org/10.3390/s21041428>

[51] Keunok Kim, Jihyeon Ryu, Youngsook Lee, and Dongho Won. 2023. An Improved Lightweight User Authentication Scheme for the Internet of Medical Things. *Sensors* 23 (2023), 1122. <https://doi.org/10.3390/s23031122>

[52] Ravi Kumar, Samayveer Singh, and Pradeep Singh. 2023. A secure and efficient computation based multifactor authentication scheme for Intelligent IoT-enabled WSNs. *Computers and Electrical Engineering* 105 (01 2023), 108495. <https://doi.org/10.1016/j.compeleceng.2022.108495>

[53] DeokKyu Kwon, Sungjin Yu, JoonYoung Lee, SeungHwan Son, and YoungHo Park. 2021. WSN-SLAP: Secure and Lightweight Mutual Authentication Protocol for Wireless Sensor Networks. *Sensors* 21 (2021), 936. <https://doi.org/10.3390/s21030936>

[54] HakJun Lee, Dongwoo Kang, Youngsook Lee, and Dongho Won. 2021. Secure Three-Factor Anonymous User Authentication Scheme for Cloud Computing Environment. *Wireless Communications and Mobile Computing* 2021 (07 2021). <https://doi.org/10.1155/2021/2098530>

- [55] Joonyoung Lee, Sungjin Yu, Myeonghyun Kim, Youngho Park, and Ashok Kumar Das. 2020. On the Design of Secure and Efficient Three-Factor Authentication Protocol Using Honey List for Wireless Sensor Networks. *IEEE Access* PP (06 2020), 1–1. <https://doi.org/10.1109/ACCESS.2020.3000790>
- [56] Zengpeng Li, Zheng Yang, Pawel Szalachowski, and Jianying Zhou. 2020. Building Low-Interactivity Multi-Factor Authenticated Key Exchange for Industrial Internet-of-Things. *IEEE Internet of Things Journal* PP (07 2020), 1–1. <https://doi.org/10.1109/JIOT.2020.3008773>
- [57] Bowen Liu, Qiang Tang, and Jianying Zhou. 2021. *Bigdata-Facilitated Two-Party Authenticated Key Exchange for IoT*. 95–116. https://doi.org/10.1007/978-3-030-91356-4_6
- [58] Zhiqiang Ma and Jun He. 2022. Outsider key compromise impersonation attack on a multi-factor authenticated key exchange protocol. In *International Conference on Applied Cryptography and Network Security*. Springer, 320–337.
- [59] Davide Maltoni, Dario Maio, Anil K Jain, Salil Prabhakar, et al. 2009. *Handbook of fingerprint recognition*. Vol. 2. Springer.
- [60] Bimal Meher and Ruhul Amin. 2022. A location-based multi-factor authentication scheme for mobile devices. *International Journal of Ad Hoc and Ubiquitous Computing* 41 (04 2022). <https://doi.org/10.1504/IJAHUC.2022.10050034>
- [61] Reem Melki, Hassan Noura, and Ali Chehab. 2020. Lightweight Multi-Factor Mutual Authentication Protocol for IoT Devices. *International Journal of Information Security* 19 (2020). <https://doi.org/10.1007/s10207-019-00484-5>
- [62] NIST. 2022. Multi-factor authentication. <https://www.nist.gov/itl/smallbusinesscyber/guidance-topic/multi-factor-authentication>.
- [63] JiHyeon Oh, Sungjin Yu, JoonYoung Lee, SeungHwan Son, MyeongHyun Kim, and YoungHo Park. 2021. A Secure and Lightweight Authentication Protocol for IoT-Based Smart Homes. *Sensors* 21 (02 2021), 1488. <https://doi.org/10.3390/s21041488>
- [64] Aleksandr Ometov, Sergey Bezzateev, Niko Mäkitalo, Sergey Andreev, Tommi Mikkonen, and Yevgeni Koucheryavy. 2018. Multi-factor authentication: A survey. *Cryptography* 2, 1 (2018), 1.
- [65] Soufiene Othman. 2023. Secure and Lightweight Authentication Protocol for Privacy Preserving Communications in Smart City Applications. (03 2023). <https://doi.org/10.3390/su15065346>
- [66] Soumya Otta, Subhrakanta Panda, Maanak Gupta, and Chittaranjan Hota. 2023. A Systematic Survey of Multi-Factor Authentication for Cloud Infrastructure. *Future Internet* 15 (04 2023), 146. <https://doi.org/10.3390/fi15040146>
- [67] Can Ozkan and Kemal Bicakci. 2020. Security Analysis of Mobile Authenticator Applications. 18–30. <https://doi.org/10.1109/ISCATURKEY51113.2020.9308020>
- [68] Fabien A. P. Petitcolas. 2011. *Kerckhoffs' Principle*. Springer US, Boston, MA, 675–675. https://doi.org/10.1007/978-1-4419-5906-5_487
- [69] Vinoth R., Lazarus Deborah, P. Vijayakumar, and Neeraj Kumar. 2020. Secure Multi-factor Authenticated Key Agreement Scheme for Industrial IoT. *IEEE Internet of Things Journal* PP (12 2020). <https://doi.org/10.1109/JIOT.2020.3024703>
- [70] Ariel Rabkin. 2008. Personal Knowledge Questions for Fallback Authentication: Security Questions in the Era of Facebook. In *SOUPS 2008: Proceedings of the 4th Symposium on Usable Privacy and Security*, vol 23, 13–23. <https://doi.org/10.1145/1408664.1408667>
- [71] Farva Rafique, Mohammad S Obaidat, Khalid Mahmood, Muhammad Faizan Ayub, Javed Ferzund, and Shehzad Ashraf Chaudhry. 2022. An efficient and provably secure certificateless protocol for industrial Internet of Things. *IEEE Transactions on Industrial Informatics* 18, 11 (2022), 8039–8046.
- [72] Mohammad Sabeeh and Ali adil Yassin. 2022. Privacy-preserving multi-factor authentication and role-based access control scheme for the E-healthcare system. *Bulletin of Electrical Engineering and Informatics* 11 (08 2022), 2131–2141. <https://doi.org/10.11591/eei.v11i4.3658>
- [73] D. Sadhukhan, S. Ray, G.P. Biswas, M.K. Khan, and M. Dasgupta. 2020. A lightweight remote user authentication scheme for IoT communication using elliptic curve cryptography. *The Journal of Supercomputing* 77 (2020), 1114–1151.
- [74] Mangal Sain, Oloviddin Normurodov, Chen Hong, and Kueh Hui. 2022. A Survey on the Security in Cyber Physical System with Multi-Factor Authentication. 1–8. <https://doi.org/10.23919/ICACT53585.2022.9728892>
- [75] James Saxon and Nick Feamster. 2022. GPS-based geolocation of consumer IP addresses. In *International Conference on Passive and Active Network Measurement*. Springer, 122–151.
- [76] Salman Shamshad, Muhammad Faizan Ayub, Khalid Mahmood, Saru Kumari, Shehzad Chaudhry, and Chien-Ming Chen. 2021. An enhanced scheme for mutual authentication for healthcare services. *Digital Communications and Networks* 8 (07 2021). <https://doi.org/10.1016/j.dcan.2021.07.002>
- [77] Geeta Sharma and Sheetal Kalra. 2018. A Lightweight User Authentication Scheme for Cloud-IoT Based Healthcare Services. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* 43 (10 2018), 1–18. <https://doi.org/10.1007/s40998-018-0146-5>
- [78] Singapore Statutes Online. n.d.. PERSONAL DATA PROTECTION ACT 2012 (2020 REVISED EDITION). <https://sso.agc.gov.sg/Act/PDPA2012>.
- [79] Enoch Solomon. 2023. Face anti-spoofing and deep learning based unsupervised image recognition systems. (2023).
- [80] Enoch Solomon and Krzysztof J Cios. 2023. FASS: Face anti-spoofing system using image quality features and deep learning. *Electronics* 12, 10 (2023), 2199.
- [81] Enoch Solomon and Krzysztof J Cios. 2023. Hdlhc: Hybrid face anti-spoofing method concatenating deep learning and hand-crafted features. In *2023 IEEE 6th International Conference On Electronic Information And Communication Technology (ICEICT)*. IEEE, 470–474.
- [82] Enoch Solomon and Abraham Woubie. 2024. Federated Learning Method for Preserving Privacy in Face Recognition System. *arXiv preprint arXiv:2403.05344* (2024).
- [83] Enoch Solomon, Abraham Woubie, and Krzysztof J Cios. 2022. UFace: An Unsupervised Deep Learning Face Verification System. *Electronics* 11, 23 (2022), 3909.
- [84] Enoch Solomon, Abraham Woubie, and Eyael Solomon Emiru. 2023. Face Image Recognition Using Deep learning Through Autoencoder Pre-Training. In *2023 International Conference on Modeling, Simulation & Intelligent Computing (MoSiCom)*. IEEE, 597–602.
- [85] Enoch Solomon, Abraham Woubie, and Eyael Solomon Emiru. 2023. Nearest Neighbor Based Unsupervised Deep Learning Image Recognition Method. In *2023 International Conference on Modeling, Simulation & Intelligent Computing (MoSiCom)*. IEEE, 592–596.
- [86] Enoch Solomon, Abraham Woubie, and Eyael Solomon Emiru. 2023. Self-supervised Deep Learning Based End-to-End Face Verification Method using Siamese Network. In *2023 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. IEEE, 1–6.
- [87] Google Team. [n. d.]. <https://support.google.com/accounts/answer/1066447>
- [88] Ertan Uysal and Mete Akgün. 2023. P/Key: PUF based second factor authentication. *PLoS one* 18 (2023), e0280181. <https://doi.org/10.1371/journal.pone.0280181>
- [89] Qingxuan Wang and Ding Wang. 2022. Understanding Failures in Security Proofs of Multi-Factor Authentication for Mobile Devices. *IEEE Transactions on Information Forensics and Security* 1 (11 2022), 1–15. <https://doi.org/10.1109/TIFS.2022.3227753>
- [90] Xiong Wang, Yuan Teng, Yaping Chi, and Hongbo Hu. 2022. A Robust and Anonymous Three-Factor Authentication Scheme Based ECC for Smart Home Environments. *Symmetry* 14 (11 2022), 2394. <https://doi.org/10.3390/sym14112394>
- [91] Wikimedia Foundation. 2023. Mutual authentication. https://en.wikipedia.org/wiki/Mutual_authentication.
- [92] Joseph Williamson and Kevin Curran. 2021. Best Practice in Multi-factor Authentication. *Semiconductor Science and Information Devices* 3 (05 2021). <https://doi.org/10.30564/ssid.v3i1.3152>
- [93] Ann Yi Wong, Eyasu Getahun Chekole, Martín Ochoa, and Jianying Zhou. 2023. On the security of containers: Threat modeling, attack analysis, and mitigation strategies. *Computers & Security* 128 (2023), 103140.
- [94] Anhao Xiang and Jun Zheng. 2020. A Situation-Aware Scheme for Efficient Device Authentication in Smart Grid-Enabled Home Area Networks. (06 2020). <https://doi.org/10.3390/electronics9060989>
- [95] Jia Xiaoying, Debiao He, Neeraj Kumar, and Kim-Kwang Raymond Choo. 2019. Authenticated key agreement scheme for fog-driven IoT healthcare system. *Wireless Networks* 25 (11 2019). <https://doi.org/10.1007/s11276-018-1759-3>
- [96] Bailin Xie, Qi Li, and Hao Qian. 2022. *Weak Password Scanning System for Penetration Testing*. 120–130. https://doi.org/10.1007/978-3-030-94029-4_9
- [97] Jinghai Yi and Yunhua Wen. 2023. An Improved Data Backup Scheme Based on Multi-Factor Authentication. In *2023 IEEE 9th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)*. 187–197. <https://doi.org/10.1109/BigDataSecurity-HPSC-IDS58521.2023.00041>
- [98] Behnam Zahednejad, Huang Teng, Saeed Kosari, and Ren Xiaojun. 2023. A Lightweight, Secure Big Data-Based Authentication and Key-Agreement Scheme for IoT with Revocability. *International Journal of Intelligent Systems* 2023 (04 2023), 1–19. <https://doi.org/10.1155/2023/9731239>
- [99] Yan Zhang, Bing Li, Jiaxin Wu, Bo Liu, Rui Chen, and Jinke Chang. 2022. Efficient and Privacy-Preserving Blockchain-Based Multifactor Device Authentication Protocol for Cross-Domain IIoT. *IEEE Internet of Things Journal* PP (05 2022), 1–1. <https://doi.org/10.1109/JIOT.2022.3176192>

A SUPPLEMENTARY FIGURES

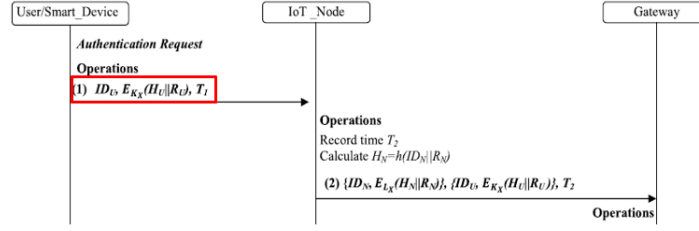


Figure A.1: Snapshot of the MFA protocol in [73] where ID_u is transmitted in plain

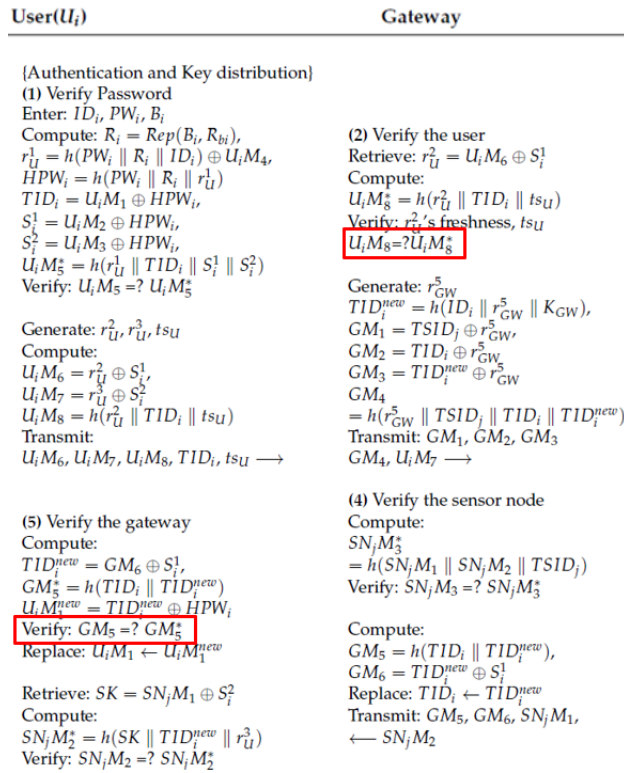


Figure A.2: Snapshot of the checks performed to mutually authenticate user and Gateway in [51]

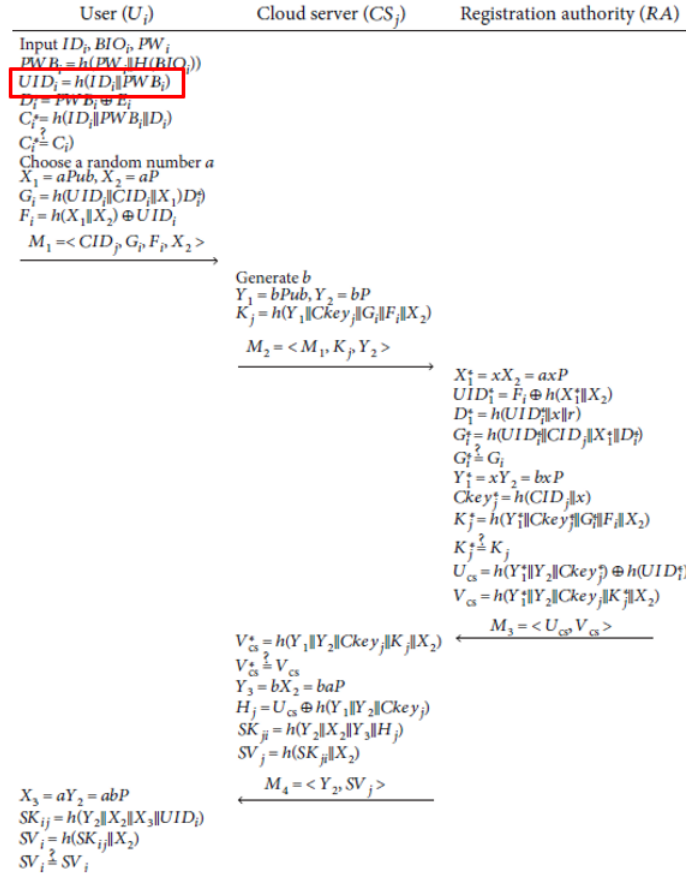


Figure A.3: Snapshot of the authentication phase in [54] whereby UID_u is used to protect the actual identity

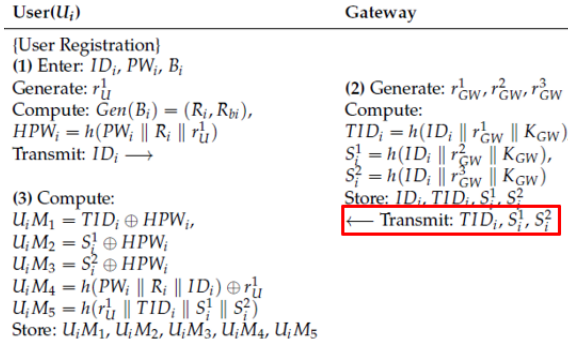


Figure A.4: Snapshot on the distribution of shared secret parameter S_i^1 in [51]

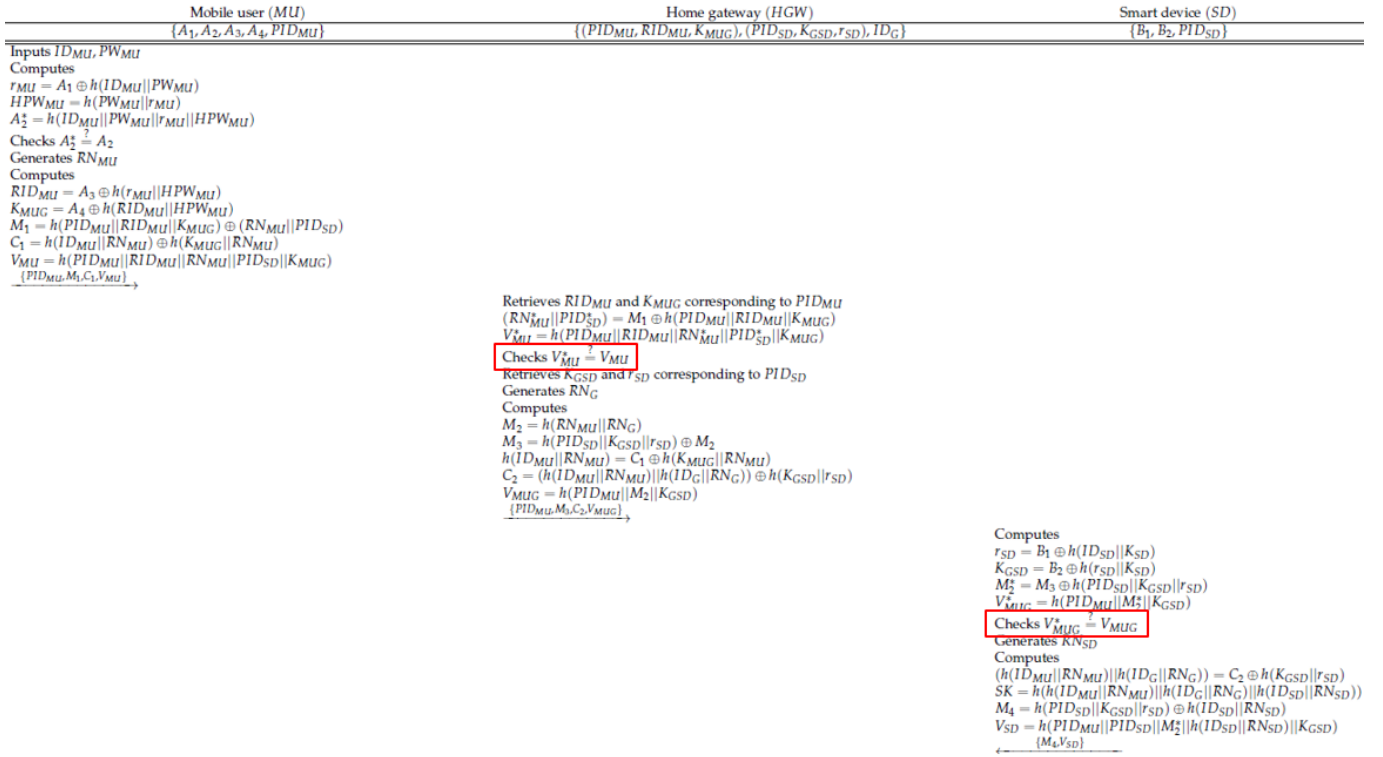


Figure A.5: Snapshot of the checks performed to mutually authenticate user, HGW and smart device in [63](part 1)

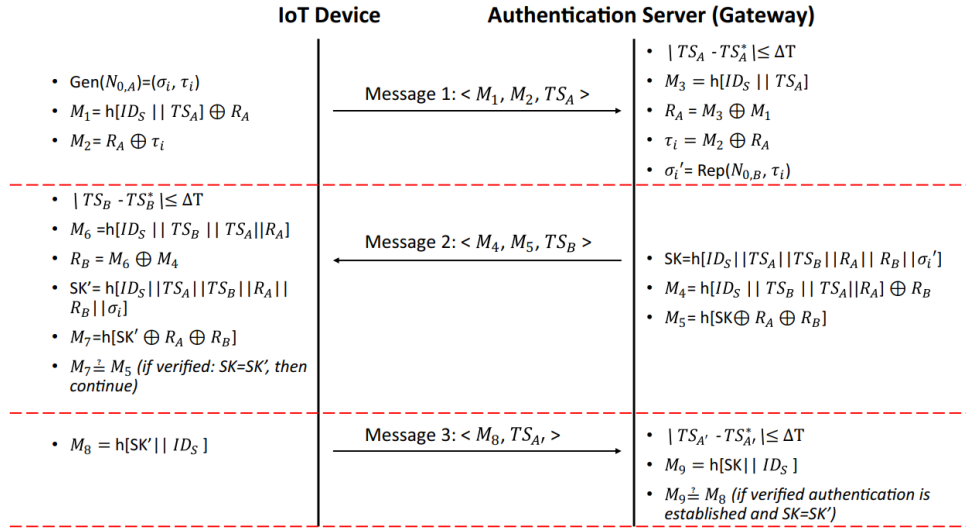


Figure A.6: Snapshot of the authentication and key agreement phase in [73]

User (U_i)	Gateway (GW)	Sensor Node (S_j)
Inserts the smart card Inputs ID_i, PW_i Computes $R_1^* = SR_1 \oplus h(ID_i PW_i)$ $APW_i^* = h(PW_i R_1^*)$ $V_1^* = h(APW_i ID_i R_1^*)$ Checks $V_1^* \stackrel{?}{=} V_1$ Generates a random nonce N_1 Computes $HID_i = SHID_i \oplus h(PW_i ID_i R_1^*)$ $S_i = SID_j \oplus h(PID_j HID_i)$ $M_1 = N_1 \oplus h(HID_i PID_j)$ $V_1 = h(SID_j PID_j N_1 HID_i)$ $\{PID_i, S_i, M_1, V_1\}$	Retrieves PID_i and the secret value x Computes $HID_i^* = PID_i \oplus h(x k_{GWN})$ $SID_j^* = S_i \oplus h(PID_j HID_i^*)$ $N_1^* = M_1 \oplus h(HID_i^* PID_j)$ $V_1^* = h(SID_j^* PID_j N_1^* HID_i^*)$ Checks $V_1^* \stackrel{?}{=} V_1$ Generates a random nonce N_2 Retrieves SID_j and $h(SID_j R_j)$ Computes $KS_j = h(h(SID_j R_j) k_{GWN})$ $M_2 = h(N_2 HID_i) \oplus h(KS_j PID_j)$ $M_3 = N_1 \oplus h(h(N_2 HID_i) KS_j)$ $V_2 = h(PID_j SID_j h(N_2 HID_i) N_1)$ $\{PID_i, M_2, M_3, V_2\}$	Computes $h(N_2 HID_i)^* = M_2 \oplus h(KS_j PID_j)$ $N_1^* = M_3 \oplus h(h(N_2 HID_i)^* PID_j)$ $V_2^* = h(PID_j SID_j h(N_2 HID_i) N_1^*)$ Checks $V_2^* \stackrel{?}{=} V_2$ Generates a random nonce N_3 Computes $SK = h(h(N_2 HID_i) N_3 N_1)$ $M_4 = N_3 \oplus h(KS_j N_2)$ $V_3 = h(SK N_3 SID_j)$ $\{M_4, V_3\}$
Computes $PID_i^{new} = P_i \oplus h(N_1 HID_i)$ $N_2^* = M_5 \oplus h(HID_i SID_j N_1)$ $N_3^* = M_6 \oplus h(N_2^* HID_i PID_i^{new})$ $SK^* = h(h(N_2^* HID_i) N_3^* N_1)$ $V_4^* = h(N_2^* N_3^* PID_i^{new} SK^*)$ Checks $V_4^* \stackrel{?}{=} V_4$ Replaces $\{PID_i\}$ to $\{PID_i^{new}\}$ in the smart card.	Computes $N_3^* = M_4 \oplus h(KS_j N_2)$ $SK^* = h(h(N_2 HID_i) N_3^* N_1)$ $V_3^* = h(SK^* N_3^* SID_j)$ Checks $V_3^* \stackrel{?}{=} V_3$ Computes $x^{new} = h(x N_2)$ $PID_i^{new} = HID_i \oplus h(x^{new} k_{GWN})$ $P_i = PID_i^{new} \oplus h(N_1 HID_i)$ $M_5 = N_2 \oplus h(HID_i SID_j N_1)$ $M_6 = N_3 \oplus h(N_2 HID_i PID_i^{new})$ $V_4 = h(N_2 N_3 PID_i^{new} SK)$ If the key agreement is successful, updates $\{PID_i, x\}$ to $\{PID_i^{new}, x^{new}\}$. $\{P_i, M_5, M_6, V_4\}$	

Figure A.7: Snapshot of the authentication phase demonstrating key leakage resiliency in [53]

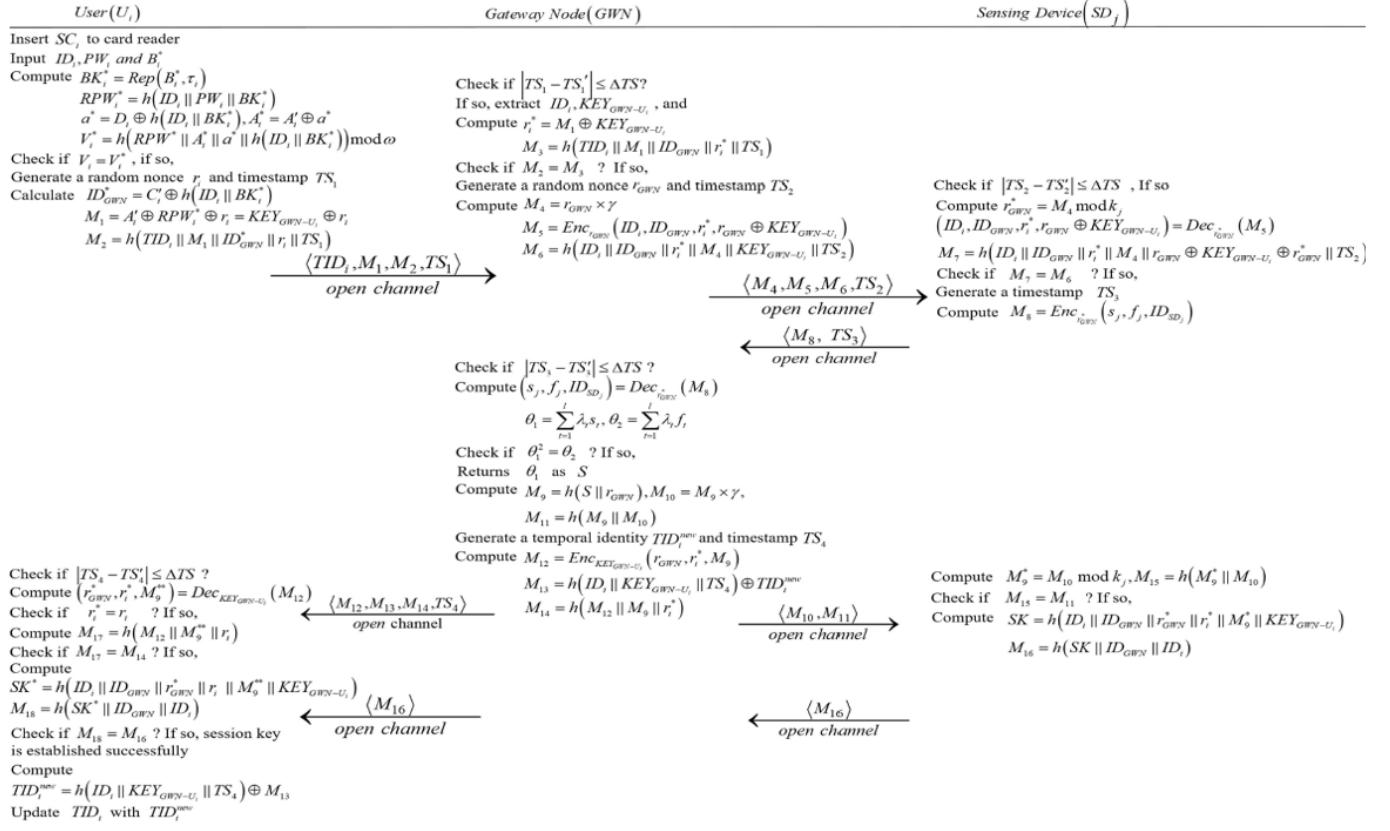


Figure A.8: Snapshot of the authentication phase in [69] showing authentication factors' independence

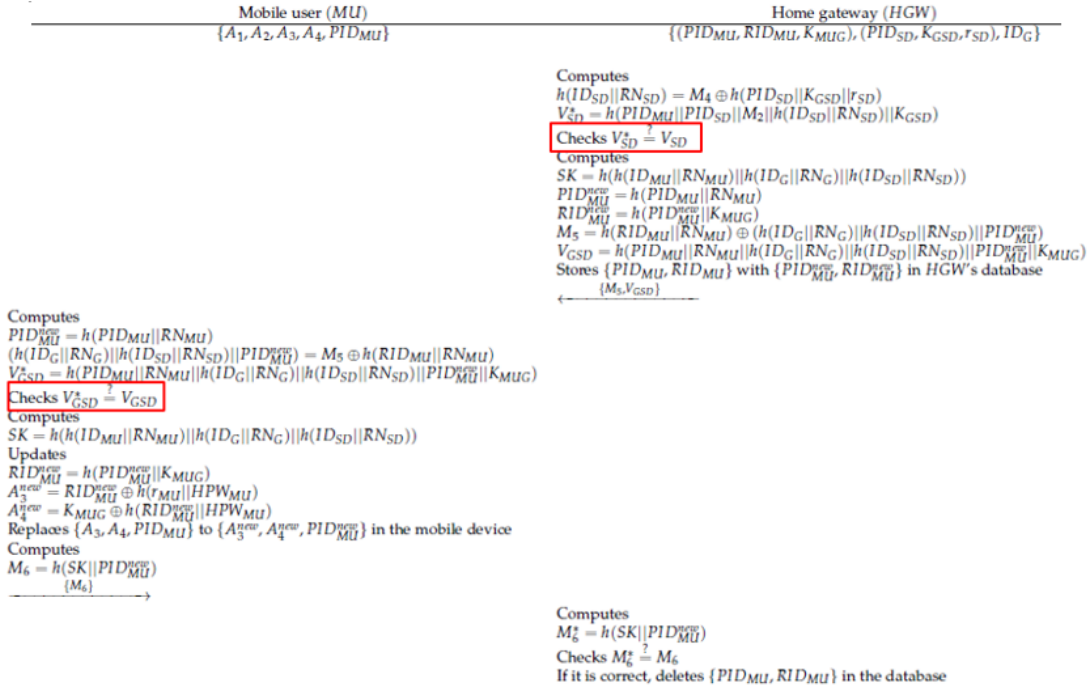


Figure A.9: Snapshot of the checks performed to mutually authenticate user, HGW and smart device in [63](part 2)