Modern Authentication Schemes in Smartphones and IoT Devices

An Empirical Survey

Ahvanooey, Milad Taleby; Zhu, Mark Xuefang; Li, Qianmu; Mazurczyk, Wojciech; Choo, Kim Kwang Raymond; Gupta, Brij B.; Conti, Mauro

DOI
10.1109/JIOT.2021.3138073

Publication date
2022

Document Version
Final published version

Published in
IEEE Internet of Things Journal

Modern Authentication Schemes in Smartphones and IoT Devices: An Empirical Survey

Milad Taleby Ahvanooey, *Senior Member, IEEE*, Mark Xuefang Zhu, *Member, IEEE*, Qianmu Li, *Member, IEEE*, Wojciech Mazurczyk, *Senior Member, IEEE*, Kim-Kwang Raymond Choo, *Senior Member, IEEE*, Birij B. Gupta, *Senior Member, IEEE*, and Mauro Conti, *Fellow, IEEE*

***Abstract*—User authentication remains a challenging issue, despite the existence of a large number of proposed solutions, such as traditional text-based, graphical-based, biometrics-based, Web-based, and hardware-based schemes. For example, some of these schemes are not suitable for deployment in an Internet of Things (IoT) setting, partly due to the hardware and/or software constraints of IoT devices. The increasing popularity and pervasiveness of IoT equipment in a broad range of settings reinforces the importance of ensuring the security and privacy of IoT devices. Therefore, in this article, we conduct a comprehensive literature review and an empirical study to gain an in-depth understanding of the different authentication schemes as well as their vulnerabilities and deficits against various types of cyberattacks when applied in IoT-based systems. Based on the identified limitations, we recommend several mitigation strategies and discuss the practical implications of our findings.**

Manuscript received July 2, 2021; revised September 13, 2021 and November 5, 2021; accepted December 20, 2021. Date of publication December 24, 2021; date of current version May 9, 2022. This work was supported in part by the Post-Doctorate Fellowship at Nanjing University, China, under Grant 2007606; in part by the "Research on Key Technology of Endogenous Security Switches" of the National Key Research and Development Program under Grant 2020YFB1804604; in part by the "New Network Equipment Based on Independent Programmable Chips" under Grant 2020YFB1804600; in part by the 2020 Industrial Internet Innovation and Development Project from Ministry of Industry and Information Technology of China; in part by the Fundamental Research Fund for the Central Universities under Grant 30920041112. The work of Kim-Kwang Raymond Choo was supported by the Cloud Technology Endowed Professorship. *(Corresponding authors: Milad Taleby Ahvanooey; Mark Xuefang Zhu; Qianmu Li.)*

Milad Taleby Ahvanooey and Mark Xuefang Zhu are with the School of Information Management, Nanjing University, Nanjing 210023, China (e-mail: m.taleby@ieee.org; xfzhu@nju.edu.cn).

Qianmu Li is with the School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the School of Intelligent Manufacturing, Wuyi University, Jiangmen 529020, China (e-mail: qianmu@njust.edu.cn).

Wojciech Mazurczyk is with the Institute of Computer Science, Warsaw University of Technology, 00-661 Warsaw, Poland (e-mail: wojciech.mazurczyk@pw.edu.pl).

Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: raymond.choo@fulbrightmail.org).

Birij B. Gupta is with the Department of Computer Engineering, National Institute of Technology, Kurukshetra, Kurukshetra 136119, India, also with the Department of Computer Science and Information Engineering, Asia University, Taichung City 413, Taiwan, and also with the School of Digital, Technologies and Arts, Staffordshire University, Stoke-on-Trent ST4 2DE, U.K. (e-mail: gupta.brij@ieee.org).

Mauro Conti is with the Department of Mathematics, University of Padua, 35131 Padua, Italy, and also with the Faculty of Electrical Engineering, Mathematics and Computer Science, TU Delft, 2628 CD Delft, The Netherlands (e-mail: conti@math.unipd.it).

This article has supplementary downloadable material available at https://doi.org/10.1109/JIOT.2021.3138073, provided by the authors.

Digital Object Identifier 10.1109/JIOT.2021.3138073

***Index Terms*—Authentication schemes (ASs), biometrics, cracking attacks, graphical passwords, Internet of Things (IoT), password security.**

I. Introduction

WHILE there are hundreds, if not thousands, of authentication schemes (ASs) proposed in the literature, text-based (e.g., numeric PIN and alphanumeric passwords) approaches remain (one of) the most commonly used mechanism(s) for verifying end-user identity. For example, these text-based ASs are deployed in online services and modern Internet of Things (IoT) systems [1] to gain access to their accounts and contents (e.g., Bitcoin wallets and cloud services). There are also other AS [3], [4], such as based on graphical patterns [5], [11], device location [6], [12], biometric traits [7], [32], Web services [2], and hardware-based secret keys (or tokens) [125], [127]. In general, text-based ASs are easy to implement and use on consumer devices (e.g., smartphones and IoT devices such as ring cameras) and do not incur additional costs on these devices such as tokens (unlike hardware-based AS) [2], [8]. Hence, it is not surprising that the text-based AS will remain popular for the foreseeable future.

However, there are known limitations in traditional text-based AS approaches. For example, an efficient text-based password is typically "easy to remember and hard to guess," which is typically short and vulnerable to the dictionary or brute-force attacks. This is particularly the case when the attackers are close to the user and can use social engineering techniques to learn the password combinations [9]. Moreover, end users usually have a habit of reusing their credentials [13]. Thereby, if an attacker obtains the user credentials for one account (e.g., during a database intrusion), (s)he may try to use the same credentials for different accounts. Nowadays, the deployment of a weak or default password-reset strategy is one of the most popular attack vectors targeting IoT devices [10]–[14]. The primary reason behind is that users typically do not change the default passwords of IoT devices and share or reuse the previously utilized credentials [4].

There exist several state-of-the-art studies that focus on improving password strength policies, such as creating strong combinations (e.g., with a mandatory usage of uppercase/lowercase letters, punctuation marks, numbers, and other characters) as well as limiting the combination length to at least 6 or 8 [16], [17]. Also, cybersecurity experts introduced strength estimation approaches [18], [19] to help end

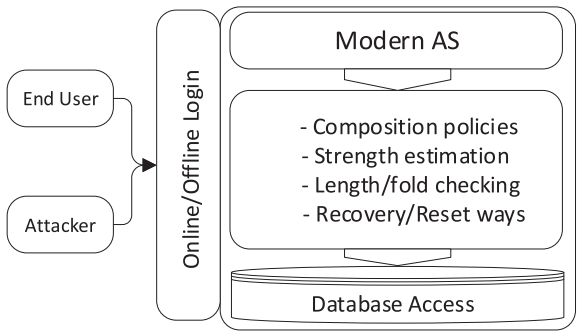

Fig. 1. Exemplary AS in the presence of an attacker.

users in picking efficient strong passwords (or patterns) by preventing the use of words related to themselves and evaluating/estimating the degree of strength. However, how to strike a balance between selecting strong combinations (e.g., those that are hard to guess and immune to brute-force attacks) and remembering different passwords for different services or devices, is still a challenge for most end users, and remains a topic of ongoing research.

A general type of AS deployed in online and IoT-based systems that includes various smart devices (e.g., wireless plugs [21], window shades [22], thermostat [23], lighting [24], door lock [25], and multiple other sensors [27]), involves two parties (end-user and online/offline system as depicted in Fig. 1). In an online scheme, end users must have the username and password (or other credentials) in which a hash value is generated and saved according to such credentials on the server during the registration phase. When the end user enters the combination along with an e-mail or username on the login page, a ciphertext message or hash value is constructed on the user's device. Then, this hash value is sent to the server for granting access to the database. On the server side, the modern scheme verifies the received hash value with the one already saved in the database. If both these two hash values are equal, the user is assumed to be legitimate and access is granted [3]. Once authenticated, these gadgets can be remotely controlled by applications (apps) installed on other smart devices (e.g., Android and iOS devices). Also, the contemporary scheme includes some safety policies, such as strength estimation, recovery/reset strategy, and length/fold checking, if a developer does not consider efficient policies during the design phase of an online scheme, it is susceptible to various types of cyberattacks. For example, there may be an attacker trying to correctly guess or obtain user passwords, e.g., by performing Man-in-the-Middle (MitM) [28] or Phishing attacks [29], using malware (e.g., keyloggers) [30], or exploiting vulnerabilities in the underpinning AS.

In this article, we first survey the literature on different IoT services, their potential risks/attack targets, and the vulnerabilities of IoT devices to potential cyberattacks due to weak AS-related security policies. Then, we categorize existing materials (e.g., 205 articles, chapters, and websites) published after 2010 based on the underpinning schemes and their contributions (e.g., new identification policies or prevention strategies) as well as various types of discovered attacks. Next, we conduct a computational complexity analysis of the existing methods to show their constrains in the IoT environment. Eventually, we discuss the mitigation solutions by addressing the limitations and vulnerabilities of the AS in IoT devices against the most serious attacks and recommend some prevention countermeasures (see Fig. 2). Clearly, throughout the years, researchers have published several related surveys. When we compare these articles with our article, the following limitations emerge.

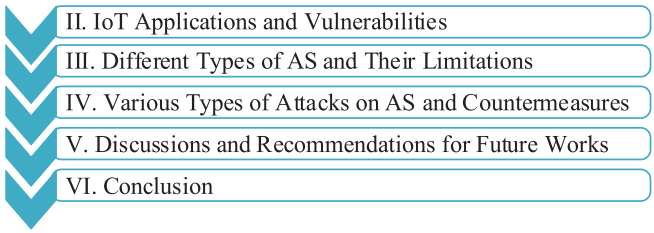

Fig. 2. Overall structure of sections for the rest of article.

1) El-hajj *et al.* [31] conducted the only most relevant survey paper on IoT-based AS, which summarizes the literature on a limited number of attacks and types of solutions, but excluding some significant threats, such as IP spoofing, cross-site scripting (XSS), and peeping. They also did not provide any discussion concerning the prevention solutions or future directions.
2) Some other studies, such as [26] and [32]–[34] comprised one type of AS by focusing on user behavioral biometrics and graphical patterns [118], without incorporating the applications of AS in IoT devices and possible vulnerabilities that can be exploited by attackers.
3) Barkadehi *et al.* [103] provided a brief review on various types of ASs, classifying a limited number of references (42) and they also did not mention the vulnerabilities of AS on smartphones and IoT devices, as well as different types of possible attacks in detail.

II. IoT Applications and Vulnerabilities

IoT encompasses a broad range of applications starting from smart medical devices (e.g., wireless pacemakers), industrial control systems, or battlefield equipment, just to name a few. There are various types of IoT devices, which can range from low-cost smart gadgets (e.g., room sensors) to expensive ones and computationally powerful devices, such as the Internet of Vehicles and unmanned aerial vehicle (UAV). These IoT-enabled gadgets typically sense, collect, and transmit data (e.g., remote sensing data) to other equipment and systems over the Internet or (a private network), without requiring human-to-computer or human-to-human interaction [35]. All of these devices utilize at least one or more types of ASs for authenticating their communications and granting control policies. Therefore, if an attacker bypasses one of the smart gadgets (e.g., a smartphone, or a sensor), (s)he can launch a breach, and get control over the other IoT-connected devices.

Considering the popularity of IoT devices (e.g., see Fig. 3, in the supplementary material [37]) and taking into account that several architectural models have been introduced in the literature, such as International Organizations for Standardization

TABLE I
VARIOUS IoT SERVICES, DEVICES, ATTACK TARGETS, AND VULNERABILITIES

IoT Services	Conventional IoT Devices	Attack Targets and Vulnerabilities	Policies / Objectives
Smart Buildings [39]-[43]	Home WI-FI/wireless routers; Motion / Occupancy sensors; Window / Door (contact sensors); Gas / Air quality/ temperature sensors; Optical / Light Sensors; Accelerometers; Asset Tag / Beacon; Security cameras;	Home routers (default passwords); Security cameras (default passwords); IP spoofing on wireless sensors; BAS programmable logic controller;	Integrity; Stability; Energy-Depletion; Reliability; Safety; Renewable energy;
Smart Government [44]-[46]	Wireless access points / routers; Emergency Alert System (EAS); Smart transportation systems; Online voting systems; Surveillance cameras;	Forensics apps and servers; Wireless routers (default passwords); Surveillance cameras; Web browsers (Sessions & Cookies);	Energy-Depletion; Reliability; Safety; Integrity; Sustainability;
Smart Economy [47]-[49]	Cloud Services Brokerage (CSB); Digital currency and E-money; Web-based banking payments; Online markets & data centers;	Web-based payment apps, and servers; Stock trading & Investments apps; Web browsers (Sessions & Cookies); E-money wallets;	Sustainability; Energy efficiency; Security; Productivity; Flexibility;
Smart Society [50]	Biomedical devices, such as Ultrasonic, CT scan, and MRI; Wearable devices, such as Bracelets, watches, Motiv Ring, Smart shirts;	Health monitoring apps; Wearable devices control apps;	Healthcare; Connected society; Sustainability; Security; Energy efficiency; Renewable energy;
Smart Environment [52]-[54]	Low-Power Wide-Area Networks; Motion sensors; Bluetooth and satellite communications; ZigBee; Earthquake or seismic sensors;	Programmable and switchable sensors; Web-based monitoring apps;	Energy management; Flexibility; Sustainability; Scalability; Reliability/Security;
Smart Mobility [55]-[57]	Online vehicle monitoring system; Crash avoidance system; Vehicle-to-everything (Satellite); communication; Navigation device; Electronic Control Unit (ECU);	Web browsers (Sessions & Cookies); Vehicle monitoring apps; Online ticket booking apps;	Flexibility; Time/Cost efficiency; Clean technology; Sustainability; Integration;

(ISO) and other private sector organizations (e.g., ISO, ETSI, 3GPP, IEEE, IETF, and one M2M) as well as academia [36]. Fig. 4, in the supplementary material, depicts an overview of a typical five-layer IoT architecture. Since the terms in this architecture are relatively generic, we skip their definitions and refer the interested reader to [35] and [36].

A. IoT Services, Devices, and Attack Targets

In this section, we classify the IoT applications into six different services, namely: 1) smart buildings; 2) smart government; 3) smart economy; 4) smart society; 5) smart environment; and 6) smart mobility [37], [38]. Also, as listed in Table I, we summarize conventional devices related to each category of IoT services along with possible attack targets and vulnerabilities.

Smart Buildings or building automation systems (BASs) include structures that exploit automated processes for controlling the building's operations automatically. They utilize microchips, sensors, and actuators for collecting data and processing them according to particular services and functions, such as air conditioning, ventilation, lighting, heating, and security. They also improve the reliability and efficiency of equipment, which minimizes energy costs, and optimizes the environmental impact of buildings [39]–[43]. Although the IoT network provides many facilities through the BAS, there exist some devices, such as wireless routers and cameras, that are vulnerable to botnet attacks, i.e., they are compromised to join the botnet as a zombie/bot as well as can be remotely controlled for nefarious purposes, e.g., becoming a part of Distributed Denial-of-Service (DDoS) attacks (like the Mirai).

Smart Government is the way of governance in which governments employ IoT technology as core-integrated systems in their operations and functions at all levels of decision-making and management. In general, the two terms: 1) smart government and 2) smart city are different as the former utilizes the latter as an area of practice [44]. In other words, smart government utilizes a range of IoT services, such as smart economy, mobility, and environment, to formulate administrative policies and business models. Moreover, it exploits such information for optimizing the public sectors' capability to meet the citizens' needs, expand employment opportunities, achieve economic growth, and support the development of the digital economy [44]–[46]. For example, digital forensics servers contain the identity information of citizens and are located in law enforcement agencies. Particularly, special agents can only access such servers via password credentials on smartphones, which are targeted by attackers through the IoT network via the MitM attack or by running malware on cameras for various purposes, such as identity theft and human trafficking.

Smart Economy or smart business refers to the primary basis of municipality development in a Smart City. This service utilizes IoT technology and works on the basis of a number of strategies to advance the attractiveness and sustainability of economy development, such as e-commerce, enhancement of productivity, innovation in production, and opportunities for entrepreneurship. The main objective of such strategies is essentially a return-on-investment model. In other words, it

employs predictive models to estimate the economical influence of the business plans to be performed and whether or not they can bring resources to the economic community and be sustainable over time. In practice, the smart economy provides a wide range of online business and investment services using IoT networks. For instance, end users usually utilize apps (e.g., bank payments, online markets, stock trading) to make purchases or access their online accounts. Consequently, smartphones and Web-based apps are the potential targets for attackers [47]–[49].

Smart Society or Smart living is an ecosystem in which digital technologies and machines are utilized to provide new solutions for addressing the current and future challenges of human life [49]. In other words, the main objective of the smart society is to develop new digital platforms using IoT technology, proposing health management strategies that organically connect assets, processes, and people. Despite the complexity that such a rapid growth can cause, it is IoT technology that offers the potential to keep and productively manage such requirements. In practice, the smart society utilizes IoT technology to monitor the health of citizens' using wearable devices. Hence, these devices are potential targets for malicious third parties to launch some cyberattacks for nefarious purposes that lead to terror or demand ransom [50].

Smart Environment refers to an environment in which individuals employ computing devices, embedded sensors, and displays so that they could accurately monitor and control environmental events, such as climate, pollution, earthquakes, and floods [52]. In other words, a smart environment not only consists of monitoring pollution and climate events, but this platform also controls the effects of environmental changes on plants, animals, and humans. To meet these requirements, an IoT-enabled environmental monitoring system utilizes artificial intelligence (AI) and high-speed communication technologies (e.g., 5G, 4G, 3G) for feeding, processing, and broadcasting data from the distributed sensors in different geographical places to data centers in real time. These sensors capture/collect some parameters, such as daylight, pressure, temperature, and humidity [53], [54]. In some cyberattacks, such sensors are targeted by attackers for malicious purposes, such as making an empty environment for robberies by creating a global panic or engaging law enforcement officers with false environmental alarms.

Smart Mobility refers to the integration of various transportation systems to make traveling cleaner, safer, and more time and cost efficient. It applies IoT technology to enable communication between user interfaces and modes of transports using a wireless network. Smart mobility has various forms, such as on-demand transportation, ride-sharing, car-sharing, and biking. In other words, the main objective of smart mobility is to reduce traffic congestion and its side-effects, consisting of fatalities, pollution, fuel costs, and waste of time. Moreover, this service should include social benefit and availability aspects, which implies that it should be affordable for every person as well as provide a desirable quality of life [55]–[57]. While smart mobility offers the aforementioned facilities, there are potential targets for attackers via smartphones due to multitude of applications of online transportation systems (e.g., ticket booking apps) and money transactions.

As mentioned above, smartphones are commonly used to control IoT devices for different application domains. They also interact with sensory data and servers through various communication systems; hence, these interactions are under the control of smartphones. Therefore, security vulnerabilities through smartphones and IoT-connected devices could provide potential ways for attackers to take control of them. However, several studies have comprehensively assessed the vulnerabilities of IoT systems and recommended some countermeasures [31], [58], but there are still many security flaws that they have not considered in their analyses. Recently, the number of unpredictable attacks on IoT devices has increased dramatically and various vulnerabilities have been discovered [60], [64].

Recent studies from security companies, such as F-Secure [59], Kaspersky [93], and IBM X-Force [94], indicate that the cybersecurity incidents are on the rise. For example, according to F-Secure's latest attack analysis report, there were over 2.8 billion attack events [59] in the first half of 2020, and a similar number of incidents were discovered in the first and the second halves of 2019. Among these numbers, 497 million in the first half of 2020 and 760 million in the second half of 2019 were related to the Telnet port (23) and 611 million attacks were for the UPnP port, i.e., these ports are typically employed in IoT devices. Moreover, there has been a massive growth in the discovered cyberattacks on IoT devices over the last two years, i.e., more than 1.5 billion in the first half of 2020 and over 2 billion events on transmission control protocol (TCP) ports in 2019 [59]. In addition to cyberattacks toward the TCP protocol, the majority of the remaining discovered events were directed at the user datagram protocol (UDP) ports (e.g., SSDP, UPnP, and LLMNR), including over 85 million in the first half of 2020 and 611 million in 2019.

B. IoT Vulnerabilities

In general, the security flaws discovered through the IoT-connected devices are similar to those found in computing systems, such as laptops, smartphones, and tablets. Because they utilize identical AS to verify access control policies for connecting to the IoT network via WiFi or other technologies (e.g., 3G/4G/5G, BLE, or LTE). In other words, each IoT device has an interconnection to a specific hotspot with the other machines which makes it easily susceptible to attacks. Note that the IoT data are captured by a single gateway port (e.g., Telnet, UPnP, or SSDP) port; thus, every device is linked through the same network. Therefore, the attackers can proceed in the same way; they only require to obtain access to an IoT device, and, hence, they can breach every other one which is connected to the same network [58], [61]. Since smartphones typically play a central role in the IoT software system, they are potential targets for attackers, i.e., if an attacker gains full access to the smartphone, (s)he can control all installed apps and connected devices. In what follows, we focus on the classification of IoT vulnerabilities

into five different groups according to the potential attack targets.

1) Hidden Backdoors: Hidden Backdoors are typically utilized as a set of covert channels in firmware or software systems that differ from conventional bugs through control-flow types, i.e., attackers can employ these flaws to gain unauthorized access to the victim's device. In particular, IoT devices may include both deliberately or unintentionally located backdoors [62], [63]. According to a report published by the CISOMAG magazine in January 2020 [64], cybersecurity experts have discovered some backdoors in IoT-connected devices, such as smart cameras [65]–[69], smart bulbs [70], fax machines [71], smart TVs [72], [73], smart speakers [74], smart coffee makers [75], [76], and smart watches [77], which attackers could successfully take control of these machines by performing malware (e.g., Mirai, WannaCry, Faxploit, Police virus, WastedLocker, Spy Bug, ADB Miner, and Wiretap). According to a large-scale empirical analysis of IoT devices in real-world homes in 2019 [78], Kumar *et al.* discovered that almost 40% of households worldwide contain at least one IoT-enabled device, and approximately 66% of these devices are located in North America. Consequently, with the increase in the use of smartphones and IoT devices, the number of cybersecurity risks is expected to increase dramatically in the near future.

2) Insecure Network Services: The IoT network utilizes the IPv6 addressing scheme, which suffers from the same security flaws as IPv4; hence, it is also vulnerable to cyberattacks, such as Sybil, spoofing, reconnaissance, MitM, Smurfing, fragmentation, and rogue attacks. Therefore, it needs similar security measures that were previously employed to quantify IPv4. Moreover, since the IoT is designed as an intersection between the Internet encounters and the physical things in the real world, it leads to the new security flaws, mostly caused by the AS through IoT devices (e.g., employing a weak AS or default password policy) [36]. In other words, it generates several critical cybersecurity implications, whereas unpredictable attacks could move from accessing information to taking control of compromised devices. Consequently, attackers can cause serious damages by launching attacks from the vulnerable devices via the security flaws in protocols and password policies. Note also that IoT-based systems are increasingly evolving from closed control systems (e.g., SCADA and Modbus) to networked systems (IP-based) [80]. As depicted in Fig. 4, in the supplementary material, there are two transport protocols above IP in the IoT architecture: 1) the UDP and 2) the TCP. In the IoT network, data packets are transmitted between devices over ports and specific IP addresses using the TCP (e.g., 23-Telnet and SMB) or UDP (e.g., 1900-SSDP and UPnP) protocols. This means that the TCP is utilized for most human interactions in Web applications, such as Web browser and e-mail as well as where preventing packet losses is preferred over latency of their arrival, while UDP is employed for online services [e.g., streaming video, Voice over IP (VoIP), and domain name system (DNS)]. The IETF Constrained RESTful Environments (CoRE) working group has presented a constrained application protocol (CoAP) at the application layer of the TCP/IP stack, specifically defined to meet the requirements of lossy and low-power networks, such as multicast support, low energy consumption, low overhead, and simplicity. In the CoAP, the UDP is used for transmitting data packets from one embedded device to another [81]. Those IoT devices that utilize the UDP protocol at the transport layer are also susceptible to a specific DDoS attack known as *UDP flooding*. Fig. 3 depicts a large-scale scenario of cyberattacks on IoT networks considering two serious ones, such as the MitM and DDoS. In practice, the botmaster can perform a network scan to find the IoT devices that are still using their factory-assigned (or default) passwords. Once such devices are compromised by malware, they can be integrated into a botnet that can later be employed for launching DDoS attacks [78]. MitM is also another serious attack, where an attacker covertly relays and potentially eavesdrops (or modifies) the data being transmitted over the communication channels between the victim user and the access point. In other words, it allows attackers to covertly obtain the login credentials, credit card data, identity theft, or other malicious actions [28], [82], [83]. For example, a botmaster can run a Mirai botnet (malware) for performing the brute-force attack to bypass the login module on the unsecured IoT devices, i.e., it utilizes a set of (62–68) default usernames and passwords. Note that the Mirai-based botnet was responsible for the largest DDoS attack that cyberspace have ever faced so far, consisting of more than 5 million compromised IoT devices (IP cameras and home routers) with a flooding speed of 1.5 Tbps, against a French cloud computing company in October 2016 [81]. Since the source codes of such attacks are openly available on the Internet (e.g., GitHub.com), attackers can easily modify them and perform their malicious tasks. Therefore, the impacts and opportunities to successfully breach in IoT devices via such cyberattacks are still increasing dramatically [81], [82].

Table II, in the supplementary material lists the most massive DDoS attacks that have targeted the IoT devices in the last four years. According to the latest Kaspersky Lab report [93], the number of DDoS attacks in 2019 was almost double this number in 2018. Besides, they classified the discovered DDoS attacks into five different groups, such as TCP SYN flooding, UDP flooding, ICMP flooding, TCP-ACK flooding, and HTTP flooding. In the following, we briefly describe these attacks based on their strategies.

1) *TCP-SYN Flooding:* In this attack, a new connection is quickly established by sending repeated SYN packets to each port of the victim machine, often employing a fake IP address known as spoofed IP packets, without completing the TCP connection. On the other side, the device must consume resources to wait for half-opened connections, which can cause the system to stop responding legitimate traffic requests [81], [83].
2) *UDP Flooding:* In this attack, a massive number of UDP packets in the form of spoofed IP packets are transmitted to a victim device (e.g., a server) to impair its capability to process network traffic and reply. As a result, the targeted machine consumes a lot of resources to respond and is quickly exhausted [81], [83].
3) *ICMP Flooding:* This type of attack is also known as ping-flooding, in which an attacker tries to overwhelm

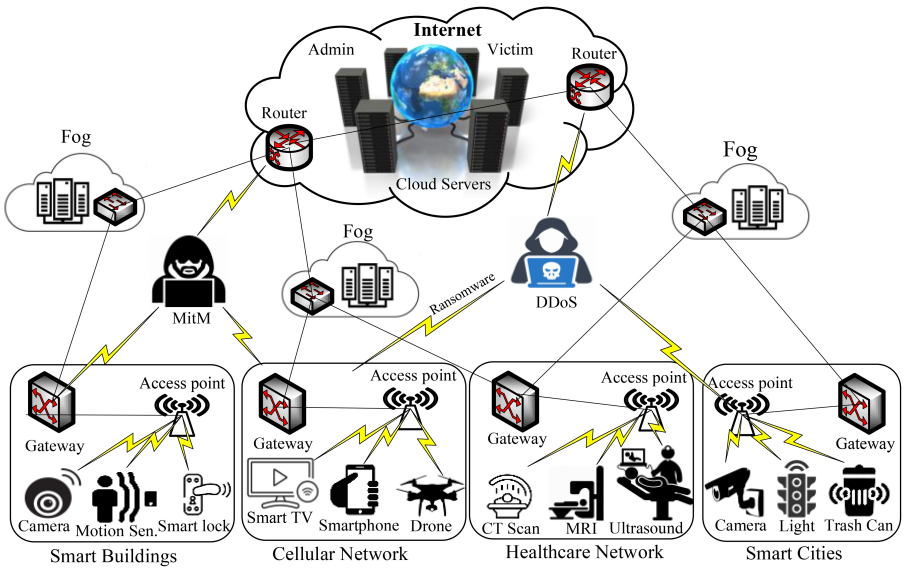

Fig. 3. Large-scale pictorial scenario of cyberattacks on the IoT network.

a victim device by transmitting a lot of ICMP echo-requests (or pings). In practice, the echo-request and echo-response messages are utilized to ping a device for identifying the state and connectivity of the host within the network. By flooding the victim machine with echo-request packets, it is forced to reply with the same number of response packets, which can overwhelm it [81], [83].

4) *TCP-ACK Flooding:* This attack attempts to overload a victim server by sending a large number of TCP-ACK packets. The primary goal of the ACK flooding is to hinder the process of providing services to other users by crashing the victim server with junk data. As a result, it must process every received ACK packet, which consumes so much computing resources that makes it unable to provide sufficient services to other legitimate requests [81], [83].
5) *HTTP Flooding:* This type of attack attempts to disable a target Web server or application by transmitting a massive number of HTTP Get/Post requests. This type of DDoS attacks is often launched via compromised devices that have been infected with malware (e.g., Trojan Horses) [81], [83].

Fig. 4 illustrates the distribution of the discovered DDoS attacks by type according to the latest Kaspersky Lab report in 2019, 2020, and 2021 [93].

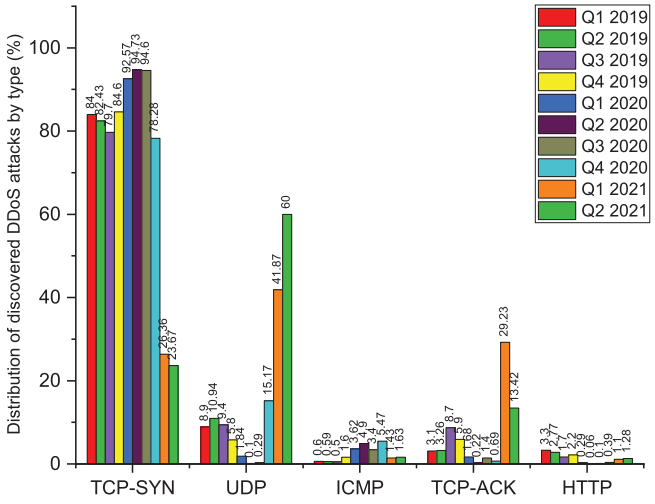

Fig. 4. Distribution of the discovered DDoS attacks by type considering the occurrence quarter (Q1–Q4) of year in 2019 and 2020, and (Q1 and Q2) in 2021 [93].

3) Weak Authentication: As we explained earlier, most Mirai-based DDoS attacks utilized a set of default usernames and passwords to by pass the login module of IoT devices. According to the monitoring analysis of thousands of IoT devices in [78], over 60% of the evaluated devices had default credentials (remained unchanged)—all default passwords of wireless routers are accessible in [79]. Hence, as depicted in Table II, in the supplementary material, we can observe that the use of default login credentials, such as in routers, CCTV cameras, and smart TVs, is still the main vulnerability of AS in IoT devices. Note that this allows an attacker to take control of millions of IoT devices on the Internet. Also, the usage of an insecure login interface (e.g., mobile, cloud, Web, and back-end API) outside the device is another vulnerability that leads to compromise it or its related elements [95]–[98]. However, manufacturers have introduced several possible countermeasures to address default password issue, such as MAC address-based, forced password change policy, and random combinations, but there are still many devices that utilize the default password policy (e.g., Wi-Fi routers, IP cameras, and printers) [82]. In [78], researchers have evaluated wireless routers from 4.8K vendors and analyzed the default login credentials of the FTP and Telnet-based routers. They also investigated the security profile of IoT devices in homes, specifically those that allow "weak authentication," i.e., the home security profile and routers which reveal their scanning

behaviors upon a large darknet. Moreover, they believed that IoT devices could act as embedded servers since 67.5% of evaluated machines offered at least a UDP or a TCP-based service.

4) Lack of Security Management: With the increasing number of IoT devices and services, security management of such devices is becoming an essential issue. Security management consists of all aspects of safety support for IoT devices, such as system monitoring, response capabilities, update policies, and asset management [99]. Current IoT stack architecture technology employs gateways (e.g., routers or hubs) to provide specific connectivity to IoT networks and other connected devices. In general, the gateways are assigned between various edge devices by the IoT configurations. Because of the limited processing and bandwidth capabilities of gateways, the vendors must adapt the quality of service in the edge IoT devices as much as possible to meet not only the network traffic requirements but also to fulfill the needs of service quality by the other edge devices while sharing the same gateway. Nevertheless, the existence of several gateways defined in an IoT network can cause an interdependent issue which is called the connection binding [100]. This problem is related to how IoT devices connect to the network as well as how they can communicate with other things, or how such things (e.g., machines, or sensors) can in turn communicate with the IoT devices. As indicated in Fig. 4, in the supplementary material, application-layer protocols, such as CoAP, AMQP, XMPP, and MQTT, are the most reliable and efficient standard protocols that provide real-time instant messaging approaches for IoT systems [36], [38]. However, these protocols are also susceptible to DDoS attacks. For example, CoAP, similar to all other UDP-based protocols, is inherently exposed to packet amplification (or reflection) and IP address spoofing. An attacker can exploit these two serious flaws for sending an amplified amount of traffic to a target device, breaching the victim's network infrastructure, and getting its resources offline. As shown in Fig. 5, an attacker can send a spoofed IP packet (request) to an IoT device (CoAP client) and the victim device responds with a "large response" packet. During this process, an attacker can substitute a randomized (fake) IP address with the IP address of the victim device. Then, (s)he can direct the DDoS attack toward the victim device due to the susceptibility of the CoAP protocol, which is sensitive to IP address spoofing. A typical amplification-and-reflection DDoS attack employs a generic scenario in which a group of compromised host packets with falsified source IP addresses (spoofed IP packets) are set to the victim's IP address and directed at a so-called reflector. In addition, a remote application replies to such packets directing traffic to the target device. As a result, the victim receives the blunt force of "amplified CoAP" traffic. Since CoAP designers have considered some security features to thwart such types of problems, CoAP is no longer lightweight when vendors include such safety features in their products. Note, however, that most available CoAP implementations do not have security measures that make them vulnerable to the DDoS attacks [101].

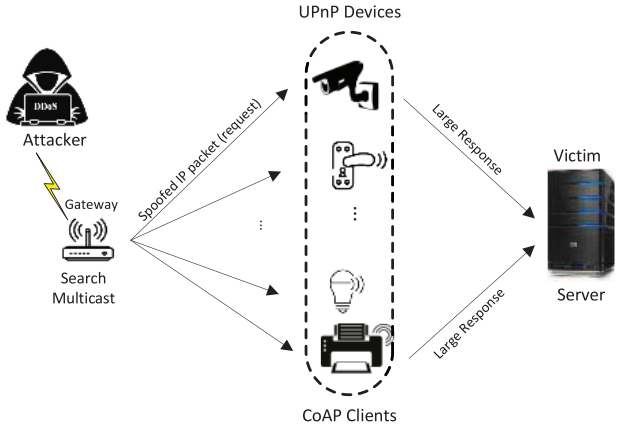

Fig. 5. DDoS (amplification) attack using IP spoofing.

5) Insufficient Privacy Protection: As IoT technology is gaining momentum in various domains, it is an essential concern to provide efficient privacy protection approaches for IoT devices that do not leave them susceptible to gain surveillance and intelligence by possible data breaches, is an essential concern. Nevertheless, in addition to the challenge of storing massive data, the preservation of user privacy and data security is one of the primary issues [184].

Since the IoT can still be considered as an immature technology, manufacturers are more interested in adding new features to attract the customers; thus, they do not focus on maintaining the privacy of "the sensitive massive data" in their products. In general, the manufacturers test their new IoT products on a limited population, then they release such devices and monitor how customers utilize them to identify which features are in demand [102]. As we have already summarized the compromised machines in Table II, in the supplementary material, they are some examples of privacy violations of the current IoT products discovered in recent years.

III. Different Types of ASs and Their Limitations

In this section, we explore different types of ASs and their associated limitations that may cause the aforementioned vulnerabilities. Technically, an AS is employed to verify either the user identity or the content authenticity of information on a website, software, or within any generic digital media or a device in an IT environment [103]. Here, we focus on the user authentication mechanism in which a system verifies the password credentials for accessing a device. In general, the information that end users employ to authenticate their identities in a device is retrieved from the following credentials: 1) a memorable answer or text—a knowledge-based combination that the user has already picked to remember as a username/password or a response to a reset/recovery question; 2) a gesture or sign—a graphical signature that the end user has already selected to remember as a pattern; 3) biometric traits, such as fingerprints [104], and iris—something end user has uniquely on his/her fingers, eyes, and face; 4) an online generated code—a verification code that end user has already requested to verify his/her credentials via

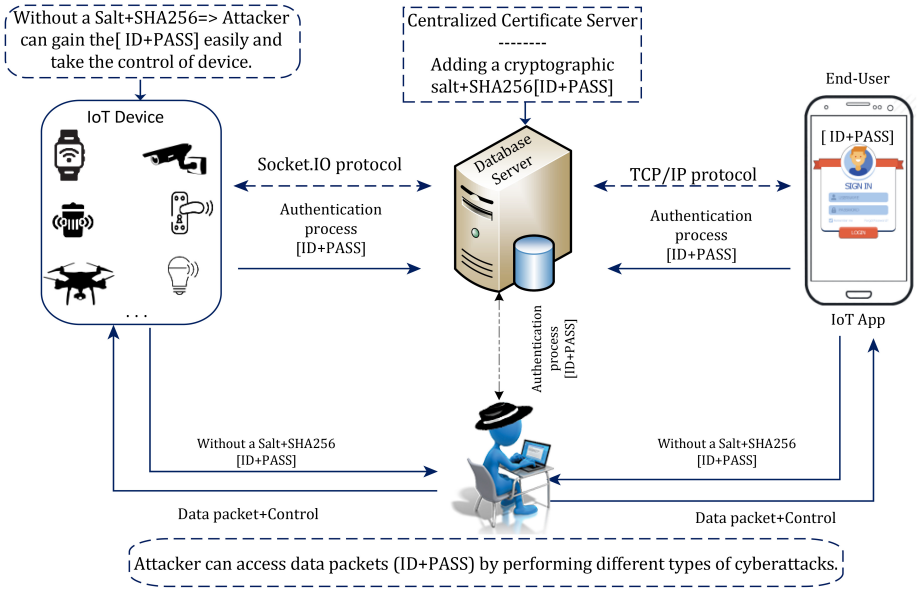

Fig. 6. Most common way of authentication process between a smartphone and other connected devices in an IoT environment.

online services; and 5) a smart card or a token—objects that end users commonly carry in their pockets, such as license, employment or student ID card. In some cases, the AS considers two or more of the above information simultaneously, i.e., an automatic teller machine (ATM) card and a pass-code (or Pin-code), which is called the multifactor AS. It implies that the multifactor AS processes one or more pieces of information when it verifies the user's credential and grants access permission [105]. Nevertheless, the concerns regarding identity theft and cracking methods have increased over the past few years, but cybersecurity experts are still attempting to enhance and introduce the use of multifactor AS to efficiently distinguish attackers from authentic users. In practice, however, multifactor ASs are more secure than the other mechanisms, but developers may need to assign more resources for development, such as hardware, software, and training [103]. Fig. 6 indicates the most common way of authentication process between a smartphone and other connected devices in an IoT system with/without applying a hash function (e.g., SHA256 and SHA512). Considering the above credentials, we classify various types of ASs into five different categories, namely: 1) text-based; 2) graphical-based; 3) biometrics-based; 4) Web-based; and 5) hardware-based approaches, as shown in Fig. 7. We describe each type of AS and its limitations in the following sections [103].

A. Text-Based Password

This type of AS involves verifying textual information as user's credentials, which is very common in computer systems and IoT devices [106], [107]. In addition to popular computer systems, such as PCs and laptops, end users should employ text-based AS within a wide variety of smart gadgets, such as smartphones, tablets, and smart door-locks. These devices provide different ways of entering text-based credentials when compared to computers (e.g., touch-screen and keypad) compared to computers. Moreover, smartphones are taking the place of computers in daily activities [108]. Technically, several safety policies affect the security level of the text-based AS, such as the length of the password, the variation of its symbols, and a limited number of attempts while the end user enters the username and password on a keypad or a touch-screen. When an attacker tries to crack a long password, it takes much longer if (s)he employs cracking approaches, such as brute-force and dictionary attacks. In a case, the modern AS utilize the above three security policies when granting access permission, they are much less likely to be cracked. Nevertheless, end users mostly prefer to exploit short-length passwords due to their simplicity and memorability [109]. To guarantee sufficient level of security, we summarize the robust safety policies as listed in Table II—developers should consider them while designing the text-based AS.

TABLE II
MULTIFACTOR SAFETY POLICIES FOR TEXT-BASED AS

Policiy	Notes
† Knowledge factor [4]	† This is the information known by end-users that they must remember when entering their credentials on the login page. Such knowledge may comprise PINs, passwords, patterns, and answers to confidential questions.
• Strength estimation [18], [19]	• It utilizes machine learning algorithms to provide a predictive and reliable feedback to users concerning the complexity of a chosen password in real-time.
◇ Geographical location [12]	◇ It employs the current location of end-user as an initial condition of authentication process. For example, smart gadgets that have GPS sensors, can be used to identify the user's geographical location.
⋆ Password hint [110]	⋆ A message for reminding the combination of a password. To remind the user's memory, the login modules permit a hint to be set, which is shown whenever the password hint is needed.
‡ Password checking limitation [109]	‡ It is often utilized as a logical restriction to other authentication policies. For instance, it counts the number of entry attempts, and considers the geographical locations of user during the last few logins – in result malicious logins can be identified if during authentication process a distant location is discovered in a short period of time.
∗ One-time password reset strategy [111]	∗ This strategy creates a randomized combination using machine learning based permutation functions which can be verified only once after creation. In some cases, it can be employed to reset the password via defined services on the login page such as SMS, email, and call.
○ Auditing policy [112], [113]	○ It tracks any configuration modifications, errors, service disruptions, and assess compromises that have occurred during the authentication of login credentials. Also, it can estimate the abnormal behaviors on login modules using AI-based predictive models. Auditing logs are essential to afford a trail of evidence where the AS is compromised.

As we pointed out in Table II, in the supplementary material, the majority of IoT devices, such as Wi-Fi routers and IP cameras, have default text-based credentials that make them vulnerable to Mirai-based DDoS attacks and other similar

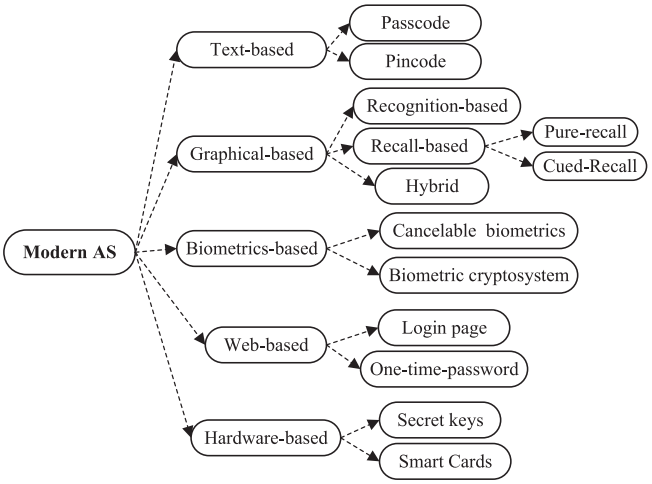

Fig. 7. Various types of ASs.

threats. However, the password reset strategies of IoT devices do not support the multifactor authentication policies mentioned in Table II. Thus, even if the user changes the default ID and password after the initial configuration of the device (e.g., wireless routers), an attacker with physical access and enough knowledge of the victim device can press a button to reset the login credential to the default one [114].

According to a technical report presented in [78], researchers evaluated 4800 routers from different vendors in 2019. Besides, they identified that TP-Link is the top vendor (15% of routers) and the most popular supplier of such devices in five regions: 1) Central Asia; 2) South Asia; 3) Eastern Europe; 4) Southeast Asia; and 5) South America. Arris is the most popular manufacturer in North America (16.4%), and Huawei is the most popular one in North Africa and the region Sub-Saharan, with an average of 19.8% and 25.6% of all home routers, respectively. They also reported that an average of 27.07% of FTP-based routers and 26.73% of Telnet-based routers among the evaluated ones have open default credentials. Due to employing the default strategy in wireless routers, they are easily accessible for everyone and still vulnerable to becoming part of the botnet performing the DDoS attacks [79]. Recently, some governments, such as the U.K. and Singapore, have announced new rules for IoT device manufacturers regarding security measures [115]. The manufacturers must omit the reset password strategy and default credentials, as well as ensure that their gadgets can receive security update patches. Otherwise, they would not be able to sell their products in the aforementioned countries. These rules consist of requiring the manufacturers to generate unique passwords for each device separately that cannot be reset to default factory settings. Moreover, they must also provide a mandatory labeling scheme that shows consumers how to secure their gadgets, such as Wi-Fi routers, smart TVs, and toys. In other words, IoT vendors need to publicly commit to a vulnerability disclosure policy that specifies the minimum period for which the device can receive security update patches. However, such kind of policies would enhance the security measures for the next generation of IoT products, but most existing devices already contain the default credentials that make them susceptible to DDoS attacks and other potential threats.

B. Graphical-Based Schemes

This type of AS employs images rather than digits or letters. In practice, the pictures are used as a part of the authentication process where end user interacts with them to identify a knowledge-based pattern [116]. Below, we describe and group the graphical-based AS into four categories: recognition-based, recall-based, and hybrid mechanisms.

Recognition-based methods involve identifying images that end users set and memorize during the password generation phase. On the login page, end users must then reuse and recognize these pictures for gaining access to the system. During the password creation phase, the end user selects a sequence of pictures from her/his portfolio. On the other hand, during the authentication process on the login page, a set of images is exhibited; then, the end user must recognize the portfolio in the same order as they were determined in the password creation phase. To facilitate memorability, end users could make a story based on pictures to remember the original pattern of the password [116]. In practice, the recognition-based mechanisms aim to provide three policies: 1) the authentication process of credentials should be reliable and easy to remember; 2) it must not allow the end users to generate weak password patterns; and 3) the created pattern should be difficult to write or share with others. Moreover, they contain three phases, such as portfolio creation, training, and authentication. In the pattern generation phase, the end user must create a portfolio with images available in the system. Then, (s)he can perform a short training process which helps him/her to memorize and practice the defined password pattern [117]–[119].

Pure-recall-based mechanisms involve analyzing a drawmetric system as an authentication tool in which end users recall and draw a sign, such as a signature or a gesture [11]. In the password creation phase, end users must draw their predefined pattern as a password either on a grid or a blank area consisting of a one or multiple strokes. During the authentication process, the end user must redraw the same pattern, and if the strokes are in the same grid and in the identical order, then the end user is granted access to the system; otherwise, the permission is denied.

Cued-Recall-based mechanisms involve recognizing patterns on specific locations within images to authenticate the user. These schemes aim at enhancing usability and reducing the memorability of complex passwords. In the password creation phase, the end user must select an image from the predefined images or gallery and click on the exact positions to draw a password pattern by considering the sequence of click-points. During the authentication process, the end user has to draw the same pattern by considering the original locations and order of click-points to gain access to the system [117], [118].

Hybrid schemes consist of a combination of two or more aforementioned mechanisms. In practice, they aim to overcome the limitations of the such schemes by combining them (e.g., the text-based password and the story

graphical-based) together and proposing a new strategy that provides a protection tool by enhancing the simplicity and safety against several attacks, such as the shoulder-surfing and spyware [118], [119].

C. Biometrics-Based Schemes

This type of AS involves processing biological measurements that can verify a user using particular traits, such as irises, retinas, fingerprints, voices, and facial characteristics. They store such biometrics data to verify a user's identity during authentication registration. These characteristics are unique to each end user, and AS do not require the user to memorize any password/pattern. For this reason, biometrics-based ASs are more widely used compared to other types of ASs. However, they are vulnerable to several types of attacks, such as the availability of biometric traits for anyone. Unlike text-based AS, the templates of the biometric AS could not be substituted if they are compromised or lost [33]. Biometric traits can be behavioral or physical. Hence, biometrics-based ASs are utilized for authenticating the identity of individuals for several sensitive applications, such as access control, immigration, border control, and forensics. Any conventional biometrics-based AS consist of two main phases: 1) registration and 2) authentication. During the registration phase, the AS scans a user's biometric image (e.g., face, fingerprint, or voice), generates a template of the biometric characteristics extracted from the image, and saves it in a database. During the authentication phase, the biometrics-based scheme scans the user's biometric data and extracts new features using the pattern recognition methods and matches the new template with the stored template of the end user. Finally, it outputs a similarity rate, indicating whether the new template exactly matches the stored template or not [32], [34], [104]. To address the challenge of stealing biometric traits, the researchers have introduced the template protection mechanisms, which is supposed to meet the following criteria [33].

1) *Diversity:* Various types of transformed templates should be similar, so that cross-matching is not possible.
2) *Noninvertibility:* It should be unlikely to regenerate the original biometric data from the transformed one.
3) *Reusability:* It should be possible to generate different forms of protected biometric templates from the original template.
4) *Robustness Performance:* Utilizing the template protection mechanism must not decrease the total performance of the Biometric AS because of high computational complexity.

As depicted in Fig. 8, template protection mechanisms are mainly classified into two categories: 1) cancelable biometrics and 2) biometric cryptosystem.

Cancelable biometrics-based mechanisms work based on distorting the biometric features deliberately and systematically that substitutes a biometric template when the saved template is lost or stolen. In other words, it is a feature domain transformation where a distorted version of a biometric template is constructed by simply mixing it with a completely artificial pattern (random salt value) and matching it in the transformed domain. In such a case, if a cancelable mechanism is compromised, the distortion features are altered, and the identical traits are mapped to generate a new template that is employed subsequently [26].

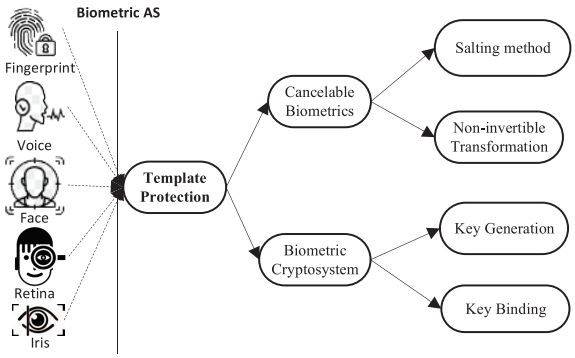

Fig. 8. Various categories of the template protection mechanisms.

Biometric cryptosystem creates an encoded token or key for encrypting the biometric data while saving it into a template. Therefore, during the authentication phase, this template is decoded to compare with the new one that is composed by extracting new biometric features. In other words, the original templates are substituted by the biometric-dependent information (known as helper data) that assists in retrieving the cryptographic secret keys. The process of matching is implemented indirectly by validating the authenticity of the recovered keys [121].

D. Web-Based Mechanisms

This type of AS involves employing online services (e.g., SMS, calls, e-mails, or patterns) to authenticate the user identity that could be exploited in any multifactor authentication system, i.e., they provide a wide range of secure strategies which are different from the conventional text-based AS [122], such as online login page, and one-time password (OTP).

Login Page: In the online-based AS, a developer may design a login page containing multiple optional AS, such as graphical pattern, text based, or biometrics based for authenticating the user's identity [123]. This option facilitates the memorability and simplicity of the authentication process, but at the same time, such ASs are susceptible to a wide range of cracking methods, such as reuses & shadow, brute-force, and dictionary attacks [124].

One-Time Password or Dynamic Password: This strategy is one of the well-known Web-based ASs, which constructs a random text-based combination exclusively valid for using one login transaction (or session) on a device. It utilizes an online service to send a randomly generated verification code to a reliable source previously defined by the end user. In practice, traditional text-based approaches are very popular for providing low-cost AS. However, such schemes are vulnerable to password-guessing, replay, phishing, cloning, eavesdropping, etc., as well as breakable against cyberattacks mostly because the end user often chooses a simple combination so that it can

be easily memorized. The OTP-based AS utilize a combination of a one-way hash function and a multifactor strategy to create a one-time valid verification code that overcomes the weaknesses of the conventional text-based AS against cracking attacks [125], [126].

E. Hardware-Based Schemes

This type of AS works based on identifying the user's credentials by relying on a dedicated physical device, such as a smart card or token. In addition to a static password, the end user provides a supplementary physical device to get access permission to the system (e.g., an ATM card). When it comes to protecting highly sensitive data, app-based or SMS-based two-factor authentication using a smartphone is more secure than relying only upon static passwords. On the contrary, they can also be time consuming and are vulnerable to network attacks [125], [127]. Hardware-based AS (or secret keys) enable a fast solution to facilitate two-factor authentication without adding complicated procedures (e.g., patterns) to the smartphone. In practice, these mechanisms function based on the FIDO U2F standard, a security protocol that is hard to intercept, i.e., this standard was developed by Google and the Yubico security company [128]. Several companies have utilized the FIDO U2F standard in their hardware-based tokens, such as YubiKey (5 NFC, 5C, and 5 Nano), and Google Titan Key (K9T and K13T), VerMark Fingerprint USB key, and Nitrokey (Pro2, HSB 2, and 3) that need connectivity ports (e.g., NFC, Bluetooth, or USB-C/A) to perform the two-factor authentication strategy. However, most of the IoT devices support at least one of these required connectivity ports, but some of them do not equip such ports; hence, the hardware-based AS cannot be applied to devices that do not have such ports.

IV. Various Types of Attacks on AS and Countermeasures

Several attacks have been presented in the literature to crack various types of ASs. Here, we explain different potential cyberattacks corresponding to each type of AS and potential countermeasures for solving security flaws (see Fig. 11), in the supplementary material.

A. Attacks on Text-Based Passwords

Guessing Attacks: In general, a text password is a knowledge-based combination of symbols, including digits, alphabetical letters, and punctuation marks, which is typically associated with a user ID. This combination allows a text-based AS to authenticate a specific end user. To break this obstacle, attackers utilize several venues for guessing passwords, such as the dictionary, social engineering, and brute-force (see Table III) [11]. The dictionary and brute-force attacks can be visualized as in Fig. 12, in the supplementary material. To mitigate the risk of being cracked by a dictionary attack or a rainbow table, the cybersecurity experts suggested hashing functions, also known as "salted hashes," to salt passwords—a salt is a random encrypted fixed-length value constructed by a secure function added to a password combination as input to some hash function, say SHA256,

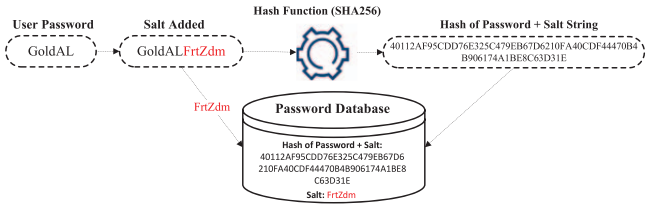

Fig. 9. Process of adding a cryptographic salt to each password combination as the input of hash function.

SHA512, or RipeMD [144]. Fig. 9 illustrates the process of adding a "salt" to a password combination in order to generate unique hashes while storing them in the database.

According to a statistical text-mining analysis of 70 million passwords in [129], scholars have reported that end users usually tend to utilize the combinations of "6–8" symbols and have a strong dislike of exploiting nonalphanumeric letters. Moreover, they suggested that an acceptable benchmark for an efficient password interval is equal to the predicted number of optimal guesses per account required to break 50% of accounts (e.g., for Yahoo! users, this number corresponded to 21.6 bits). To guess a correct text-based password on an unlimited login module, the attacker utilizes the brute-force attacks to crack it by simply attempting several combinations of symbols repeatedly until a match is discovered. Since the strategy of this attack is very simple and restricted to trying as many combinations as possible, this is also named the "exhaustive search." The intruder generally employs a high-performance server or computer for experimenting with a large number of possible combinations (NPCs) within a short period. Let us suppose that an end user could create a password with three combinations: 1) ten numbers $[0-9]$; 2) 52 English alphabet letters $[26(A\cdots Z)+26(a\cdots z)]$; and 3) 33 punctuation and special symbols which depends on the number of choices (NC) and length of password (LP). Therefore, the NPC can be calculated as follows:

$$\mathrm{NPC}=\mathrm{NC}^{\mathrm{LP}}. \tag{1}$$

In an experimental project called Projekt RC5-72 [135], cybersecurity researchers evaluated how fast text-based passwords could be cracked. In this study, they aimed to decode a message that was encrypted with a 72-bit key. To meet this requirement, they tried various possible keys until the proper key is identified. In the previous projects of this organization, researchers succeeded in cracking a 64-bit key within 1757 days and a 56-bit key within 250 days. Table IV depicts some examples of the possible combinations and the estimated time for cracking different passwords. Since the problem of cracking combinations depends on the factors, such as *simplicity* and *memorability*, developers need to consider the strength estimation policies/guidelines to ensure that end users choose stronger combinations [8], [18], [19].

Network-based attacks typically target a Web-based login page for discovering the password combinations to access a unique online application. In general, these attacks can be classified into four types: 1) Phishing; 2) Keyloggers; 3) Replay;

TABLE III
MOST COMMON CYBERATTACKS FOR CRACKING THE TEXT-BASED AS

Attack Name	Description	Open-source Codes / Tools
Dictionary or Rainbow tables [13]	This attack tries to break the AS of a device by systematically entering each word in a dictionary as a password. It can also be employed in an attempt to obtain the key required to decode an encoded document or message. A rainbow table is a dictionary which is utilized to break the AS by decoding the password hashes. In other words, it is a predefined dictionary containing plain-text passwords and their related hash values that could be exploited to discover which plain-text password generates a particular hash value.	CrackStation's Dataset 64 million passwords [131] MD5-Hash cracking with Rainbow tables (Java) [132]
Brute-force [8]	This is a cracking attempt also known as an exhaustive search that works based on estimating conceivable combinations of a password until the correct one is matched. A longer password leads to more combinations that require to be tested. This attack can be time-consuming, hard to accomplish if ways such as data obfuscation are employed and at times downright impossible. In a case that the length of combination is short (less than 6 letters), it may take a few seconds to be cracked, i.e., if the AS does not check the limitation of password tries.	Android app to Brute Force Wi-Fi passwords (Java) [133]
Social Engineering -Shoulder Surfing (Baiting) [13]	This is the act of obtaining secret information in which an individual spies the user's activities to catch the username and password. In other words, the malicious third party utilizes some invisible tricks, such as monitoring the victim device using a camera called Peeping attack, watching it undercover, or manipulating the victim into disclosing sensitive information that can lead to cracking his/her password credentials. In Baiting attacks, a malicious user leaves infected USB sticks or other things in the employer or public locations hoping that an employee may pick them up and connect to their devices.	A Collection of Social Engineering tricks [134] Toolkit [135]

TABLE IV
RESULTS OF THE NPCS GENERATED BY THE BRUTE-FORCE ATTACK BASED ON THE EXAMPLES [136]

Password Example	Possible Combinations	Time Required to Crack
Password-Length= 5 letters [2 numbers+3 lowercase letters]= $10 + 26 = 36$	$36^5 = 60,466,176$	$\frac{60,466,176}{2,000,000,000} \cong 0.03$ Seconds
Password-Length= 6 letters [2 uppercase +2 lowercase +2 numbers]	$62^6 = 56,800,235,584$	$\frac{56,800,235,584}{2,000,000,000} = \cong 28$ seconds
Password-Length= 7 letters [6 uppercase letters + 1 lowercase character]	$52^7 = 1,028,071,702,528$	$\frac{1,028,071,702,528}{2,000,000,000} = 514 \cong 9$ minutes
Password-Length=8 letters [2 punctuation +4 lowercase +2 numbers]	$68^8 = 457,163,239,653,376$	$\frac{457,163,239,653,376}{2,000,000,000} = 228,581$ Sec. $\cong 2.6$ days
Password-Length=9 letters [3 lowercase+2 uppercase+2 numbers+2 punctuation]	$95^9 = 630,249,409,724,609,375$	$\frac{630,249,409,724,609,375}{2,000,000,000} = 315,124,704$ Sec. $\cong 9.2$ years
Password-Length= 12 letters [3 uppercase + 4 lowercase + 3 punctuation+2 numbers]	$95^{12} = 540,360,087,662,636,962,890,625$	$\frac{540,360,087,662,636,962,890,625}{2,000,000,000} \cong 27,018,004,3831,318$ Sec.$\cong 8.5$ million years

TABLE V
MOST COMMON CYBER ATTEMPTS FOR PERFORMING NETWORK ATTACKS ON THE TEXT-BASED AS

Attack Name	Description	Open-source Codes / Tools
Phishing or Spear phishing [271]	This is a cyber trick which employs a carefully crafted email message. The aim is to trap the receiving party into believing that the email is something they need or expect. For instance, a message from someone in their office or a request from their bank. If end-user clicks on a link or downloads an infected file which is attached to email, then this leads to sending the credentials to the attacker.	A framework used for phishing session cookies and credentials of any Web service (Python). [139]
Keyloggers [140]	Keyloggers or keystroke loggers are malware programs that record the traces of keys pressed on a device to covertly monitor the victim user's action by logging keystrokes and transferring them to an attacker. While they are rarely utilized for authentic purposes (e.g., parental/surveillance monitoring tools), keystroke loggers are often maliciously employed by the attackers to leak sensitive information. Many passwords and credit card numbers have been stolen so far utilizing keyloggers, which makes them one of the most hazardous kinds of spyware identified so far.	Java-Keylogger utilizes a Native Hook library to add global listener for key presses. [141]
Traffic Interception or Replay [142]	This attack involves utilizing the encoded data to break the security protocols by stealing the message during the transmission from a particular sender and driving it into the victim device. In this case, the attackers try to lure the parties into believing that they have successfully accomplished the data transmission between two intended participants. Therefore, they resend the valid credentials (or messages) to misdirect the recipient into operating what the attacker requests. The effective strategy of replay attack is that an intruder does not require advanced skills to decode the credentials after catching from the network channel, i.e., (s)he simply needs to resend the same message.	An SMS-Based-Payment-System which generates the AES Key for all transactions to prevent the Replay attack (Java). [143]
MitM [28]	This attack involves employing a malware which does not only monitor information being transmitted, but also actively embeds itself in the middle of the connection by impersonating an app or a website. It allows malware to gain the user's credentials and other confidential information, such as social security numbers, or bank account numbers. The MitM attacks are often involved by social engineering attacks that delude the end-user to a spurious Website in order to steal the relevant cookies or sessions.	A LittleProxy-MitM is an extension which allows all the filter capabilities of LittleProxy in HTTPS Websites (Java). [144]

and 4) MitM attacks, which are briefly summarized in Table V and illustrated in Fig. 10.

Recently, researchers have proposed several techniques to release warnings concerning the predicted websites in the early phases of phishing activities. To detect such attacks, most commonly utilized methods employ fixed blacklists or machine learning techniques [145], [146] to eliminate redundant and irrelevant characteristics as well as to extract the most related minimal set of features that can efficiently estimate phishing attempts. In general, attackers have multiple ways to mislead the end users to launch a Web-based phishing attack. Therefore, the existing techniques do not provide sufficient performance in accurately recognizing such new variants of phishing tricks, which are under the guise of legitimate websites [146]. Consequently, efficient phishing detection techniques need adaptive and intelligent solutions that can accurately predict the new forms of phishing activities. Since the Keyloggers and Phishing attacks are unpredictable as they depend on the user interactions, it is a very complicated task to defend against new forms of such threats [145], [147]. To protect the AS from the MitM attacks, it is essential to employ efficient encryption and authentication mechanisms between the two parties (the user's device and server). In such a scenario, the server has to verify the device's request by assigning a digital certificate, and only the communication is established only when a valid certificate is identified. For instance, HTTPS utilizes the secure sockets layer (SSL) protocol to cover Web traffic. To decode the HTTPS, a MitM attacker would need to gain the keys employed to encode the network traffic, which is a very challenging task [28].

In a replay attack, since the attacker who intercepts the original credentials (or message) does not need to read or decode the secret key, he/she must obtain the entire credentials and resend them to the victim device. To prevent this vulnerability, developers should establish an entirely random session key between both the sender and the receiver, which is a kind of randomized code similar to OTP, i.e., it is only valid for one-time verification. Another countermeasure against this type of attack is to apply timestamps to all credentials. It prevents attackers from resending network packets transmitted previously; hence, it reduces the opportunity of being intercepted or eavesdropped on by the attacker. Another technique to prevent the AS from becoming a victim of the replay attack, is to assign a unique randomized password for encoding

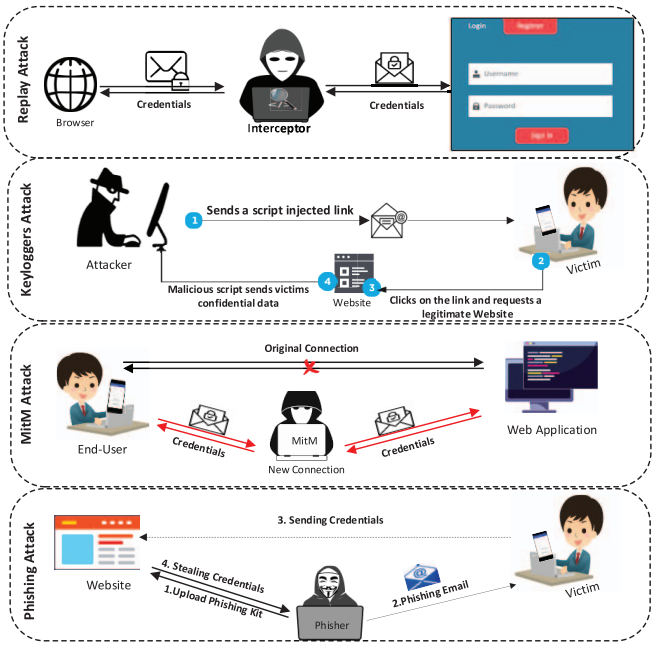

Fig. 10. Illustration of Replay, Key-Logger, Phishing, and MitM Attacks.

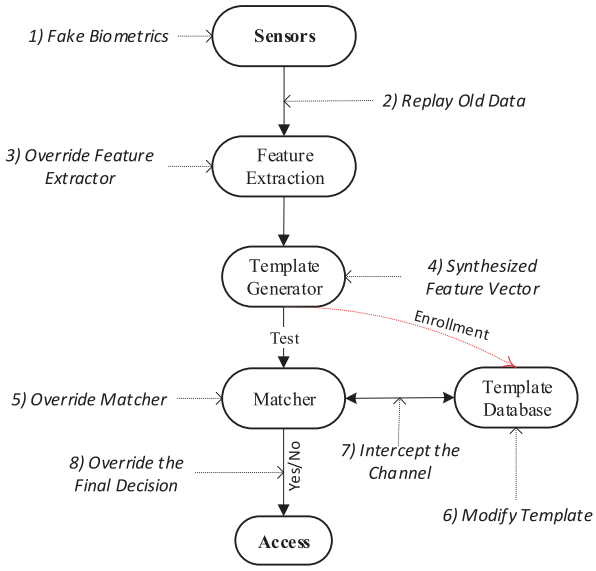

Fig. 11. Possible attacks on various phase of the biometric AS.

each transaction between two parties that is valid only once. Therefore, it guarantees that even if the credential is saved and resent by the attacker, then encrypted code is already consumed or invalid [147], [148], [183].

B. Attacks on Graphical-Based Passwords

Peeping attack is a threat to user authentication known as a "Shoulder Surfing" in which the adversary monitors the screen of the target device over the end user's shoulder to discover the password or a pattern. This usually occurs in public places where someone observes the screen of the targeted smartphone or records a video via cameras (e.g., drones). This type of attack can be classified into two different types [149].

1) *Active Attack:* A malicious user has a short distance to the victim and directly watches the screen of the target device as well as (s)he may ask some questions or send a message for driving the victim to use his credentials. In such a scenario, the intruder may discover a partial view of the target screen since some part is expected to be guarded by fingers or hands.
2) *Passive Attack:* An attacker attempts to record password credentials via surveillance cameras, drones, or malware to capture the full view of the target device's screen where end users enter their patterns or combinations.

Purely automated dictionary attack involves estimating a set of pass-point graphical patterns that end users are "more likely" to pick; these sets genuinely produce a collection of data that can be employed in a dictionary attack. An enforceable attack must be able to efficiently create a dictionary consisting of highly possible patterns [118]. These cyberattacks function based on the assumption that end users are more likely to pick click points associated with predictable paths by clustering the points within a pattern of click order such as four points in a direct line, or selecting such points on the parts of the picture where their attention is typically drawn toward. Van Oorschot *et al.* [150] proposed and evaluated some purely automated attacks against the pass-point style graphical-based AS. They also suggested an approach to effectively generate dictionaries based on heuristics such as click-order patterns (e.g., four points all on a straight line) for cracking graphical patterns. This strategy combines click-order heuristics with the focus-of-attention scanpaths by considering the model of visual attention and rating patterns. In practice, such patterns actually could not predict the correct choices since a background picture considerably changes the range of user's choices. To guess a correct gesture-based pattern at least three basic requirements should be met [118]: 1) learning click patterns from harvested gestures, even if they are not obtained from the target image; 2) constructing dictionaries for formerly unseen images; and 3) functioning on simple paths and complex gestures. The automated attack approaches have studied the possibility of cracking the pass-point graphical-based AS on the limited number of images. However, where the user chooses a unique picture as the default background, then guessing a correct pattern with the precise gesture order, is a challenging task for such attacks.

C. Attacks on Biometrics-Based AS

The primary drawback of the biometrics-based AS is that traits are openly accessible for anyone (e.g., fingerprints, facial photos, or voices). Because of the openness of such traits, attackers could easily exploit them and create new ways to break the security of biometrics-based AS [33]. As shown in Fig. 11, there are eight possible attacks on various phases of biometrics-based AS.

Attacks on sensors involve providing fake biometric traits in front of sensors to bypass the biometrics-based AS. In practice, attackers could create a spurious object with legitimate features, such as fake fingerprints, facial pictures of the genuine user, wearing some lenses with the original iris features [120]. There are possible countermeasures for these

kinds of attacks, including multibiometrics, liveness detection, and soft biometrics [151], [197]. Multibiometric ASs are used to prevent such attacks by considering two or more traits (e.g., a combination of iris, voice, fingerprint, or face) for authenticating the user identity. When several biometric features are employed, it is a very complex task for the attacker to gain access to all of them simultaneously. Thus, multibiometrics can significantly limit or even prevent spoofing attacks. In multibiometrics, various types of traits are combined either by blending the matching scores or by fusing the features. Consequently, the possibility of creating fake biometric patterns by attackers is very low [151]. *Liveness* detection technique identifies whether the biometric data are captured from a live user or a fake object. Hence, it is possible to detect different live gestures, such as face temperature, blood pressure, sweating pores, eye movement, facial movement, or pulse rate, by employing an additional program or hardware. Nevertheless, using an extra real-time identification program or hardware makes the biometrics-based AS more complicated than the conventional ones [197]. Soft biometrics are the traits that are not uniquely sufficient to verify a person (e.g., eye color, gender, age, skin color, and height) [152]. During the enrolment phase, the application of additional soft-biometrics factors may increase the security level of biometrics-based AS to some extent. Hence, it could contribute to measure the exact matching score threshold in both multimodal and unimodal biometrics, which can protect the AS from spoofing attack [153].

Replay Old Data: In this type of attack, an intruder could steal the biometric traits and later utilize the previously saved data to bypass the feature extraction procedure. Recently, researchers have introduced data hiding techniques to protect the biometrics-based AS against replay old data attacks. For instance, Khan *et al.* [154] suggested a steganography-based technique to hide fingerprint-biometric templates into audio signals that could securely transmit the biometric traits without giving any trace to the attackers. In practice, steganography is employed for providing covert communication; hence, the biometric data could be transferred to various modules through the biometric AS via innocent-looking cover media (e.g., text, image, audio, and video). In such a case, the presence of biometric traits is hidden under the cover of another object that is undetectable to attackers. Smith *et al.* [155] introduced a watermarking technique for protecting the biometrics-based AS using facial traits which could counter replay attacks by employing colors exhibited on the screen reflected from the end user's face. In this work, they exploited the color reflections to recognize that whether the facial pictures are captured in real time, i.e., the reflected colors are used to watermark the captured images. Their experiments demonstrated that the color reflections of the facial photos can be precisely classified under ideal conditions, which could be utilized to defend against replay attacks in the face recognition-based AS.

Overriding feature extractor is typically performed by attacking the software of the biometric AS. This involves substituting the feature extractor module with a feature set by performing a Trojan horse that contains malicious code, and can be remotely controlled by an attacker to catch the biometric traits and override the AS [156].

Spoofing Attack by Synthesized Feature Vector: In this attack, the path from the feature extraction to the matcher is monitored to steal the vector consisting of the traits of the legitimate user, i.e., the Matcher is an algorithm that processes the live features and compares them with the existing traits stored in the database. Then, the malicious party performs a replay attack by embedding the stolen vector to bypass the matcher module [157].

Overriding matcher involves infecting the biometrics-based AS by spyware (or a Trojan horse) in such a way that the attacker could manipulate the matching score by creating a high similarity rate to bypass the AS [33].

Attack on the Template Database: This attack modifies the database records while the templates are saved by replacing or embedding fake biometric traits. In other words, when an attacker compromises the security of the template database, (s)he can substitute the biometric traits of a legitimate user with another unauthorized person. To prevent this type of attack, researchers have proposed several template protection approaches in the literature, such as cancelable biometrics and biometric cryptosystem. As we already explained, in the cancelable protection mechanism, it is saved in the database instead of the legitimate biometric data. Hence, the imposter cannot gain access to the original biometric data in the database. In the cryptosystem protection strategies, since the biometric data are encoded before they are stored in the template database; thus, this is very complicated for the attackers to decode the encrypted biometric data. It is also impossible to identify certain data and/or replace it with a new data from the template database [151], [158].

Intercepting the Channel Between Database and Matcher: This attack involves interfering with the channel to change the existing biometric data or replay the stolen old data. To defend against this attack, data hiding-based response systems in the form of watermarking or steganographic techniques have been introduced in [33].

Hill-Climbing Attacks or Override the Final Decision: This threat involves evaluating the matching scores generated by the Matcher to construct synthetic biometrics data that can provide a false recognition in the final decision. In practice, the matcher is executed repeatedly whenever newly updated data are presented at a certain stage; then, it computes the matching output score until a successful acceptance is achieved. Surprisingly, in this cyberattack, adversary does not require any special previous knowledge of the victim user's biometric traits to perform this strategy. In other words, this attack only gathers the statistical information regarding the complete distributions of the biometrics traits that must be available while the intruder overrides the actual decision of the Matcher [159], [160]. In the literature, multibiometrics-based AS works based on both serial and parallel fusion mechanisms using soft biometric features have provided achievable recognition rates and sufficient security against multiple hill-climbing attacks [153].

D. Attacks on Web-Based AS

Web-based ASs are vulnerable to network threats, such as SQL injection attack (SQLIA), cross-site request forgery (CSRF), and XSS [161].

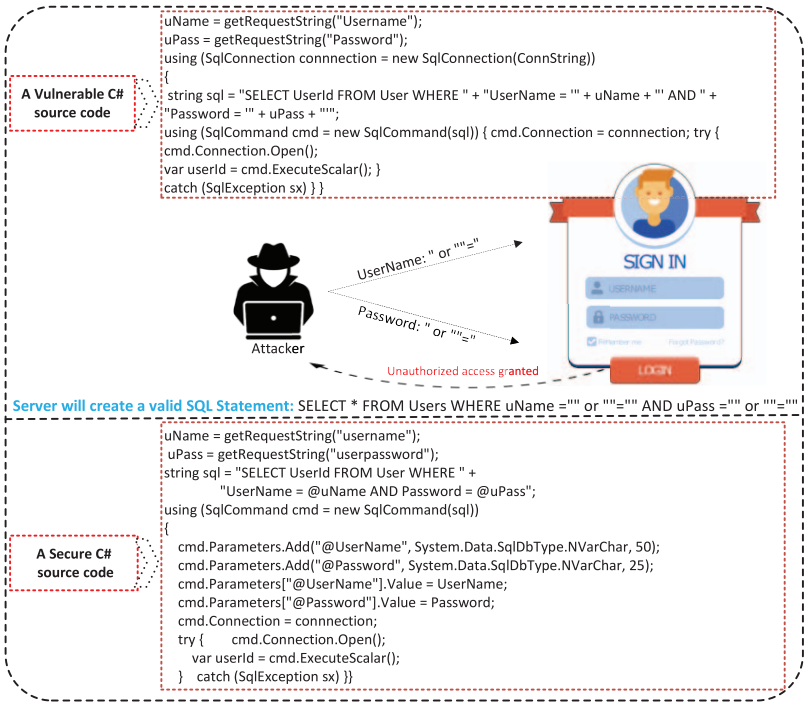

Fig. 12. Illustration of the SQLIA on a Web login page.

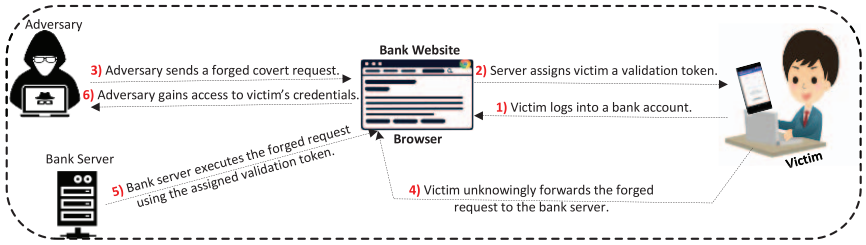

Fig. 13. Illustration of the CSRF attack on a bank website.

SQL Injection Attack: Since the Web-based AS utilizes a database as a back-end to save the user's credentials (e.g., username/password, graphical patterns, or biometric data), they must employ a back-end functionality which is provided by the database management systems (DBMSs), such as MySQL, SQL Server, and Oracle. In the case of using SQL server, the statements are dynamically generated from the end-user data. These statements involve processing direct interactions by the SQL database. As shown in Fig. 12, if the developer does not consider the SQL parameters such as "@" during the design of a Web-based AS, the attacker can inject a malicious code in SQL statements through the login page input values. Practically, it occurs when a user enters some input credentials on the login page (e.g., a username and password) and instead of such data, the attacker types a malicious data (e.g., "or" and "="") that the AS will execute on the database. In this attack, an intruder can forge SQL commands inside input fields by embedding malicious data into them. When the server processes such malicious data, then adversary can gain unauthorized access to the system [161]. To protect a Web-based AS from SQLIAs, developers must utilize the aforementioned SQL parameters. These parameters are text strings/symbols that are inserted into an SQL query at execution time to control the input data. Moreover, they should note that SQL parameters are added within the statement by a "@" letter. Hence, the SQL engine evaluates each parameter for ensuring whether it is not as part of the SQL query (or code), and if it is identified true for its corresponding column, then it could be performed [162].

Cross-Site Request Forgery is a serious security vulnerability in the Web 2.0 environment, where a website causes an end user's browser to execute malicious code on a legitimate page; thus, the saved passwords are exposed via Cookies. To initiate a CSRF attack, the malicious party sends unauthorized requests from the end user's device to the accessed websites that the end user trusts. The requests coming from a CSRF attack are "unauthorized," i.e., the end user did not initiate (or authorize) them and is not even aware of being transmitted from the device [161]. These requests could provide hidden access to the victim's credentials on a legitimate website for attackers. Fig. 13 demonstrates a flow diagram schematically showing an exemplary CSRF attack on a bank website. The adversary launches a CSRF attack by forging the HTTP request which utilizes the current session of the

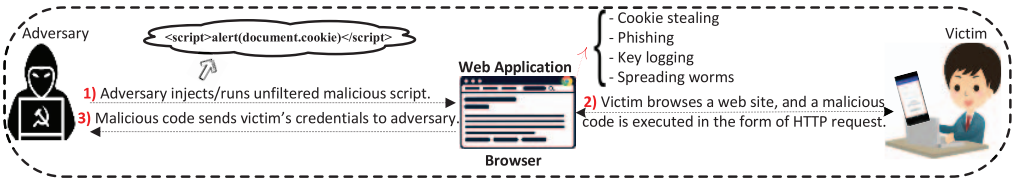

Fig. 14. Illustration of the XSS attack on a website.

end user from the browser, i.e., every user, after logging in to a Web-based AS, receives a Session ID (called SID) that is transmitted back to the server for confirmation either by a "GET" variable or by the cookie for temporary use until the browser is closed. This attack targets a legitimate website as a payload in the form of a JavaScript or an HTML file, which contains malicious code to activate and execute a covert action on another third-party Web page on behalf of the victim's device. To prevent this attack, there are two alternative techniques: 1) Double Submit Cookie Pattern and 2) Synchronizer Token Pattern [163]–[165].

1) Since saving the CSRF token through a session is problematic, a preventive way is to employ a *double-submit-cookie* for storing the token. In such a case, the CSRF token is transmitted to the server twice, i.e., once into the cookie and then inside the user's actual request. As the cookie is preserved by the "same-origin policy," only scripts of the same domain can access the unique cookie or modify the relevant value. Therefore, if a duplicate value is saved via the form and the cookie, the server authenticates that the request was managed by the website (or a resource) of the same domain. Consequently, cross-domain resources cannot generate a valid request; hence, the forged requests from the CSRF attacks will be removed [163].
2) *Synchronizer − Token − Pattern* utilizes an encryption technique, rather than a comparison for authenticating the token validation, i.e., a legitimate user has a "hidden token" that is created on the server side. In other words, after successfully validating a user token, the server creates a unique hidden token consisting of the SID, a nonce, and a timestamp value using a specific key accessible only by the server. The encrypted token is embedded in a covert field and sent back to the client's Web browser. Later, the AJAX requests contain this token within the header of packets in an identical manner for performing the Double-Submit pattern, i.e., non-AJAX form-based requests would essentially embed the token in their covert forms. When a server receives a request, the decryption function decodes the hidden token value using the same key that was employed to encode it. Furthermore, the server extracts the SID and timestamp from the decrypted token value as well as authenticates their validity by comparing the SID with the logged-in user's session ID and the current time with the timestamp [165].

Cross-Site Scripting (CSS or XSS) is a type of script injection vulnerability in which a malicious code is injected into trustworthy Web applications. These attacks occur when an attacker utilizes a website to covertly transfer a "malicious code," typically in the form of a client-side script in a browser. Practically, the XSS-based attacks employ technical flaws to break into the victim's browser, which are quite common and occur whenever a website collects input data from an end user (e.g., via the login page), i.e., it creates values without encoding or validating the input parameters [166], [167]. As depicted in Fig. 14, an attacker can exploit the XSS attack to inject a malicious script into the target browser so that the user does not identify its existence. Hence, the browser cannot detect that the injected script is not trustworthy and will launch it.

Since it considers the script sent from a trusted source, the malicious code can access all sensitive information, such as *session tokens*, and *cookies* stored within the browser and utilized with the corresponding website. These contents (scripts) could even maliciously rewrite the source code of the HTML file. Moreover, they can be sent by the attacker to the browser in a hidden way using a "%" instead of "<" or ">" within the malicious script, which often changes the form of a JavaScript segment but can also contain the Flash, HTML, or any other types of source codes that the browser could execute [168]. In general, there exists a great variety of XSS-based attacks that consist of sensitive data, such as sessions (or cookies) to the intruder, redirecting the end user to the Web content trapped by the attacker, or executing other unauthorized activities under the guise of a compromised website. In general, XSS-based attacks can be classified into three groups: 1) stored; 2) reflected; and 3) document object model (DOM)-based.

1) *Stored or persistent XSS-based attacks* involve injecting the script where it is permanently stored on the database server, such as in a visitor log, message forum, comment box, and a database. When the browser requests the stored information, it retrieves the malicious script from the data server. In this scenario, an attacker does not need to entice the victim into clicking/opening a URL. In the case that the server fails to sanitize the malicious code, all connected devices are vulnerable to be infected by the same script [161], [166].
2) *Reflected-based or nonpersistent XSS attacks* embed a malicious script in a covert way which can be reflected by the Web server, alike in the search result, an error message, or any other "response" that contains some or all of the input data transmitted to the server as part of the HTTP request. In practice, they are transmitted to the victim's browser via some other means (e.g., in an e-mail message or using another Web pages). When the victim user is deluded into opening (or clicking) the trapped

TABLE VI
MOST EFFICIENT PREVENTION METHODS AGAINST XSS ATTACKS

Prevention Method	Note	Open-source Codes/Tools
Escape dynamic content [171]	Since websites are typically created using the HTML (e.g., designed in the forms of template files) using a dynamic content pattern, the best prevention technique to filter the abnormal execution is to escape the input data via rendering such templates on browsers. These techniques could mitigate the majority of XSS attacks by validating the user's input and guaranteeing that such data is rendered and escaped by templates.	SimpleBase Library HTML and PHP [173]
HTML sanitization [167], [168], [170]	The sanitization approaches involve analyzing an HTML file and construct a new one by keeping the tags that are identified as "desired", and "safe". They can be applied to protect the content of a Web page against XSS attacks by sanitizing any malicious code injected through the user's input data.	JS_XSS Sanitizer Module using a Whitelist JavaScript and HTML [174]
Content-Security Policies [172]	Modern browsers perform security policies during evaluating the web content which permits the developer of a website to control where the JavaScript (and/or other resources) can be executed or loaded from. Due to XSS-based attacks depend on the ability of the malicious party to run a malicious code on a victim's website – either by tricking the user's browser into retrieving the JavaScript from a URL or by injecting it inside the "<script>" tags somewhere within the <html>tag of a website. In the case that web developer considers a content security policy for the response header; thus, browser can detect not to execute covert JavaScript codes as well as block such domains that direct to destructive scripts.	Content-Security-Policy HTTP response header JavaScript and HTML [175]

link by sending a particularly invisible form or even simply browsing a malicious Web page. Then, the injected script is executed on the website, and the browser performs this malicious code as coming from a "trusted" server. This leaves the victim's browser vulnerable to any unauthorized access by the attacker [161], [166].

3) *DOM-Based XSS Attacks:* The W3C DOM is a language-neutral interface and platform that allows scripts and programs to dynamically access or update the structure, style, and content of a document. The DOM-based XSS attack involves modifying the DOM where the whole affected data flow from the source to the sink occurs in the victim's browser. It implies that DOM contains the source of the sensitive data and the sink and the data flow do not leave the client's browser. For instance, the source is a location where the malicious code is retrieved, which can be a URL, JavaScript, or HTML code, and the sink is tricked into executing of the malicious script (e.g., document.write) [161], [166].

Although the above three types of XSS attacks are typically defined, i.e., Stored, Reflected, and DOM; in practice, however, they overlap in some cases. Since the possibility of two or more of these XSS-based attacks occurring simultaneously is ambiguous, cybersecurity researchers have introduced two new terms to classify the different types of XSS attacks: 1) Server-side XSS and 2) Client-side XSS [167]–[171]. Because the XSS attack is a kind of script injection, experts eventually proposed the prevention methods as listed in Table VI, which can sanitize the possibility of malicious code injections via user's input data and protect the browser against such attacks to some extent.

E. Attacks on hardware-Based AS

Since the hardware-based AS usually employ a physical device (e.g., a security token, or a smart card) to evaluate the multifactor schemes, attackers are involved with the hardware or software vulnerabilities on the device to bypass such security measures [175]. Practically, the text-based ASs are secured by considering the multifactor safety policies (see Table II) and by exploiting communication channel protection approaches that could mitigate the stated attacks, such as replay, eavesdropping, SQL injection, and session hijacking attacks. In addition to text-based schemes, the hardware-based AS can also be applied to address the vulnerabilities of other conventional AS, such as Web based and Biometrics based, by comprising a secret token as the initial authentication eligibility. In other words, it is possible to prevent most of the aforementioned attacks by employing functions of hardware tokens, except those noticed below [175], [176].

Malicious Code Intrusion: In practice, the hardware secret keys or tokens require to be connected to a device (e.g., a smartphone, a laptop, or an ATM) for executing the AS; hence, they are vulnerable to malicious code attacks on the victim machine which can covertly intercept and capture soft credentials while the user enters them on the login page. Even if the hardware token is equipped with the biometrics-based AS (or text based), the malicious code could either catch such sensitive data or wait until the end user activates the token. Due to the variety of malicious code attacks, none of the hardware-based tokens can provide perfect protection against such types of attacks. It is important to emphasize that the hardware-based AS still provide sufficient protection (superior to the other AS). Even if an attacker steals the user's credentials through some forms of spyware, (s)he is not able to crack the security of the device without possessing the token [175], [176].

MitM Attacks on Hardware-Based AS: In general, most phishing cyberthreats utilize the Web browser extensions to launch their malicious code and drive relay and real-time MitM attacks [179]. In the case that an attacker intends to steal the victim's smart card data remotely, (s)he can execute relay attacks on a target card by planting a hidden proxy in front of the corresponding card reader posing as the original one. On the other side, this proxy catches the application data unit (APDU) commands of the card reader, and then, transmits the card's credentials via the proxy-mole link and *vice versa* for cracking the APDU responses. In such a case, if an access code is required to employ the card, the attacker can gain it in advance by processing the card through the mole. Note that intruders can employ the relay attack to execute the MitM attacks between the reader and the smart card depending on the context [180].

To protect against the above attacks, researchers have introduced the Universal Second Factor (U2F) tokens (e.g., FIDO U2F and FIDO2) which are remarkably efficient types of second-factor hardware-based AS [176]. Since these tokens bind their authentication credentials cryptographically to a unique origin, they obstruct phishing attacks unlike other AS, such as time-based OTP, which leave the AS vulnerable to cyberattacks. According to a report published by the Krebsonsecurity website in early 2017 [177], Google mandated that all of its employees utilize U2F tokens, and this company has not detected a single case of corporate credential theft so far. To show the computational cost characteristics of the existing IoT-based AS, we summarized the security

TABLE VII
COMPUTATIONAL COMPLEXITY ANALYSIS OF THE STATE-OF-THE ART IOT-BASED SCHEMES

Scheme	Computational Complexity Analysis	Security Proof Logic (Model)	Deployed Security Assumption(s) Cryptography/Machine Learning algorithms	Target Application
[51]	2 Exp, 13 Add	Yes (Real-Or-Random)	Biometric key, hash function and XOR operation	Smartphones, IoT gateways, and sensors
[95]	2 Exp, 2 Add, 5 Mul	Yes(Indistinguishability under Chosen Plaintext Attack model)	Public and private-keys	Smartphones, and IoT traffic sensors
[96]	1 Mul, 2 Add, 2 Exp	Yes(ProVerf tool)	Hash function, symmetric key, and public and private keys	IoT smart building sensors and edge devices
[97]	10 Add, 2Mul	Yes (AVISPA tool and LAM-CIoT)	Hash functions and XOR operations	Smartphones and IoT devices
[116]	3 Mul, 3 Add	No	Dynamic pattern based	IoT touch screen consumer devices
[123]	1 Expo, 4 Add	No	Hash function, keystroke biometrics, and Symmetric key	Smartphones
[127]	13 Add, 1 Exp, 1 Mul	No	SHA-3, Elliptic Curve, and AES 256-bit	Wearable devices
[151]	6 Exp	No	Deep learning based pattern recognition algorithm	IoT touch screen consumer devices
[155]	5 Exp, 3 Add	No	Graphical pattern-based using SVM	IoT touch screen consumer devices
[164]	2 Exp, 1 Mul, 4 Add	Yes (Uniformly-distributed dictionary of exclusively feasible passwords)	Hash function, XOR operation, and Elliptic Curve point addition	IoT devices
[182]	6 Add, 4 Exp, 6 Mul	Yes (Zipfs law)	Hash function, XOR operation, biometric key, Elliptic Curve point addition and scalar multiplication, fuzzy extractor operation	Smartphones, IoT gateways and sensors
[183]	4 Add, 1 Mul	No	HMAC-SHA-1, SHA-256 and SHA-512, XOR operation, and Elliptic Curve DiffieHellman (ECDH)	Smartphones and IoT traffic sensors
[185]	31 Add, 1 Mul, 2 Exp	Yes (Game theory based model)	Elliptic Curve, Hash function, and XOR operation	IoT medical sensors, gateways, and smartphones
[186]	26 Add, 5 Exp, 2Mul	Yes (Real-Or-Random)	Hash function, biometric key, XOR operation, Elliptic Curve point addition, and scalar multiplication, AES-128, and fuzzy extractor operation	IoT gateways, traffic sensors, and smart building devices, and smartphones
[187]	2 Add, 2 Mul	No	Dynamic pattern recognition technique	IoT touch screen consumer devices
[194]	4 Add, 5 Exp	No	Cancelable biometrics, transformation key, and feature extraction technique	Smartphones and IoT consumer devices
[195]	3 Add	Yes(Mao and Boyd logic)	Universal hashing, and symmetric key	Smartphones, IoT gateways, and sensors
[196]	16 Add, 1 Exp	Yes(Game theory based model)	Hash function, XOR operation, fuzzy extractor, and concatenation operation	Smartphones, IoT gateways, and sensors
[198]	2 Add, 3 Exp	No	Blockchain technology+ (Smart contracts, SHA-256, public and private keys)	Smartphones, IoT gateways, and traffic sensors
[200]	10 Add, 2 Exp	No	Blockchain technology+ (Hash function, public and private keys, digital key, passkey, and digital signature)	Smartphones, IoT gateways, and sensors
[201]	16 Add, 4 Exp, 2 Mul	Yes (Real-Or-Random)	Blockchain technology+(Elliptic Curve point multiplication, Elliptic Curve point addition, public and private keys, and XOR operation, Hash function, and Modular multiplication)	Smartphones, IoT gateways, and sensors
[202]	2 Add, 4 Exp, 2 Mul	No	Blockchain technology+ (Elliptic Curve point addition, hash function, public and private keys)	Smartphones, IoT gateways, and sensors
[203]	2 Add, 4 Exp	No	Blockchain technology+ (Smart contract, hash function, public and private keys)	Smartphones, IoT gateways, and sensors
[204]	9 Add, 1 Exp	No	Recurrent Neural Networks (RNN), Hamming distance, and XOR operation	Smartphones, IoT gateways, and sensors

*Note that we extracted the minimum computational cost of each scheme based on the deployed assumption(s), and ranked them according to the required computing cost: Exponential (Exp), Addition (Add), and Multiplication (Mul).

TABLE VIII
REMAINING OPEN PROBLEMS IN MODERN AS AND RECOMMENDED COUNTERMEASURES

AS Mechanisms	Various Types	Support devices	Remaining Open Problems	Recommended Countermeasures
† Text-based [1],[4],[8],[9] [10], [13]-[20], [106]-[114], [125]-[126], [183], [193] [198]-[204]	† Passcode † Pincode	† Computers † Smart gadgets † IoT devices	† Crackability of default credentials against the Mirai-based DDoS attacks † Difficulty with memorability of complex passwords † Vulnerable to network and guessing attacks.	† Removing the default password or reset strategy from the IoT devices. † Introducing the OTP based online verification instead of the default or reset password strategy. † Limiting the number of password checking. † Adding the strength estimation to help users selecting a strong password. † Including the AI-based hint option as a reminder to help users. † Considering the geographical location and auditing policy during the authentication phase to track abnormal behaviors. † Adding a cryptographic salt to password combinations before being hashed.
• Graphical-based [11], [12], [116]-[119], [149]-[150], [187], [188]	• Recognition-based • Pure-recall-based • Hybrid	• Computers • Smart gadgets	• Vulnerable to peeping attack using a camera or a spyware. • Difficulty with memorability of complex patterns. • Fixed click points make the gestures/patterns vulnerable to guessing attacks.	• Generating random click points to solve the vulnerability against guessing attacks. • Adding invisible hints to remind the complex patterns. • Applying machine learning techniques to detect abnormal behaviors during the drawing gestures/ patterns.
◇ Biometrics-based [26], [32]-[34],[104] [120],[121],[123] [151]-[158], [182], [189], [192],[194] [195], [197]	◇ Fingerprint ◇ Voice recognition ◇ Face recognition ◇ Retina verification ◇ Iris recognition ◇ Multi-biometrics	◇ Computers ◇ Smart gadgets ◇ Hardware tokens ◇ IoT devices	◇ Open accessibility of biometric traits to anyone. ◇ Generating a fake object using legitimate features. ◇ Vulnerable to several types of attacks on different phase of the biometric AS.	◇ Employing the AI-based liveness detection techniques to recognize the fake or legitimate objects. ◇ Applying information hiding methods to protect the biometric traits from being caught against replay attack. ◇ Using template protection methods (cancelable or cryptosystem) to secure the traits against manipulation by attackers where templates are saved in the database.
∗ Web-based [2], [122]-[126] [161]-[171]	∗ Login page ∗ The OTP	∗ Computers ∗ Smart gadgets ∗ IoT devices	∗ Vulnerable to SQLIA, CSRF, and XSS attacks. ∗ Depending on browser's safety policies which makes it vulnerable to network attacks. ∗ Extensions on browsers can steal the sessions containing login credentials.	∗ Developing the login page based on safe policies which the extensions or cookies could not access the user's session (or credentials). ∗ Employing the sanitization methods to protect the web login against XSS attacks. ∗ Escaping the dynamic HTML content to prevent covert script injection within data.
‡ Hardware-based [127],[128] [175]-[180], [185], [186]	‡ Secret keys (tokens) ‡ Smart cards	‡ Computers ‡ Smart gadgets	‡ Vulnerable to hardware theft attacks. ‡ Possibility of losing the hardware token. ‡ Breach of malicious codes through software to steal the victim's credentials. ‡ Smart cards are susceptible to MitM attacks.	‡ Applying the U2F tokens to protect sensitive devices due to their results were promising as Google experienced. ‡ Adding a tracking device (GPS) into the token to find it where it is being stolen. ‡ Employing multi-factor biometrics for protecting smart cards.

proofs and their underlying deployed assumptions as listed in Table VII.

V. DISCUSSIONS AND RECOMMENDATIONS FOR FUTURE WORKS

In this section, we briefly summarize the open problems of modern AS and recommend potential countermeasures to address such issues as future research directions (see Table VIII). Note that we obtained the following open problems empirically as rules of thumb, and readers should not consider them dogmatically or rigidly. However, the following points might help cybersecurity researchers in developing an efficient modern AS in terms of usability and reliability.

A. Open Problems

In this section, we briefly describe the existing open problems in modern AS that need to be addressed to enhance the usability and reliability of such mechanisms in IoT-based systems.

IP Spoofing or (Amplification) Is the Remaining Vulnerability in IoT Devices: In practice, most of the DDoS attacks often employ IP spoofing to overwhelm a victim device by sending a substantial number of requests with falsified source IP address(es) to disguise the identity of the attacker. Hence, the most mitigation strategies fail to detect such attacks due to the anonymity of their exact source. If the IP address of the source (attacker) is continuously randomized and falsified, blocking such IP packets (requests) becomes a very tedious task. It also makes the process of tracking the attacker very difficult for cybersecurity experts and law enforcement agencies. However, recent researches [190], [195], [198] proved that the use of two-factor-based AS during the connection establishment in wireless network access could address such challenges efficiently.

Difficulty With Memorability of Complex Passwords: Since the complex and long combinations are difficult to memorize, a reminder option is essential in the modern AS for guiding users. Another remaining problem is that if a user selects a strong password (e.g., a combination of various symbols), the memorability of such combinations must be facilitated, as it is an immensely hard task after a while. On the contrary, however, the short-length passwords (less than six characters) can be memorized/remembered easily, but they are susceptible to cracking attacks [8], [15]. To address this issue, we recommend that developers employ security checking policies such as an AI-based hint (asking some questions concerning the recent activities of the end user on the device), identifying the geographical location and other safety settings (see Table II). The current form of password hint is a fixed message defined by the end user and is easily susceptible to social engineering attacks.

Vulnerability of Default Password and Reset Strategies in IoT Devices: As already mentioned in Table II, in the supplementary material, the Mirai-like botnets utilize default passwords of IoT devices to breach and take control of them. To prevent such attacks, we suggest manufacturers to eliminate the default password strategy because anyone aware of this issue can break the AS. For example, a pin option used in most Wi-Fi routers could reset the credentials to the default settings [79]. As an alternative, we recommend developers to design an OTP-based online strategy (e.g., random CAPTCHA, e-mail, or phone number) in the modern AS for activating the IoT devices upon the first use. Then, the end user could utilize the selected option to reset the login credentials if they do not remember them. This strategy mitigates the possibility of being cracked by malicious attacks such as the Mirai [184].

Possibility of Data Breaches and Leakage in IoT-Based Systems: Most connected machines or sensors involve authenticating the login credentials by sending some data packets to a database server. In practice, attackers can compromise the modern AS by performing different types of cyberattacks (e.g., Replay, MitM, Phishing, or Keyloggers) on such data packets, as they are transmitted over the network without being hashed or salted. To guard against such attacks, we recommend that developers apply a time-based OTP methodology so that AS can compose a randomized key with the current timestamp using hash functions for generating one-time valid data packets. On the server side, the modern AS can decode hashed packets only during a short period (e.g., 2 min). When the server (or device) receives an invalid packet (after the time has expired), it must send an acknowledgment packet to another side to request new valid variants.

Shoulder Surfing in Public Places: In general, end users utilize smartphones in different crowded places, such as universities or train stations and may employ one or more forms of modern AS to unlock their phones. Whenever malicious users observe the screen of the victim's device, they could discover/guess the credentials (e.g., pattern or password) from the way a legitimate user draws or types them. Fig. 15 depicts the dashed ASs which are vulnerable to shoulder surfing attacks. To guard the AS against such attacks, we recommend that developers design untraceable gestures or patterns on the screen. Moreover, a combination of randomly generated click points and covert hints using sensors could be an alternative to protect the modern AS from being captured by shoulder surfing attacks. Note, however, that these types of ASs are still vulnerable to peeping attacks.

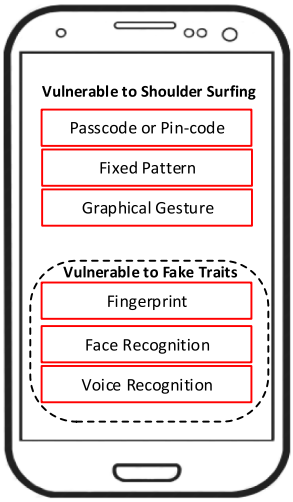

Fig. 15. Modern AS on smartphones vulnerable to shoulder surfing attack and fake biometric traits.

Vulnerability of Web Browsers to Code Injection Attacks: Since end users utilize browsers for purchasing goods from online stores, investing in stock trading markets (e.g., cryptocurrencies), and engaging in other sensitive activities, they are potential targets for various code injection attacks [199]. As we already pointed out, there exist several flaws through browsers allowing attackers to perform code injection via user input which leaks the sessions with credentials. To defend the Web-based AS against such attacks, we suggest that while developers design a login page, they should employ the sanitization or/and escaping dynamic content (e.g., HTML and JavaScript) strategies to filter the user input data from any possible script injection. Such mechanisms can be utilized as client side safeguards to equip browsers via extensions that automatically detect abnormal execution attempts. However, the client side technique does not work correctly against some types of XSS attacks, except for continuous XSS, where the injected script is not transferred via input values [167].

Complexity in Identifying a Legitimate User on a Device: Another open problem is the ambiguity of identifying a legitimate user among other ones when they use the same credentials on a device. Since most ASs (except biometrics-based) utilize fixed passwords or patterns, anyone with such credentials can bypass the AS. Here, the problem is that no matter how a malicious user gains credentials, but (s)he is not a legitimate one. To distinguish which users are employing credentials on a particular device, a multifactor biometrics-based AS that takes into account different types of traits and user behaviors can be applied to address this issue [110], [192].

Blockchain-Based Schemes as Decentralized Security Controllers: Several studies have demonstrated the potential of utilizing blockchain to efficiently control and manage the

authentication of devices, such as edge nodes, gateways, and other connected machines in IoT-based systems [200]–[203]. Conventional ASs for IoT applications [96], [97], [99], [109], [182] typically rely on a centralized server or certificate authorization center as a trusted third party; therefore, such systems are at risk of single-point failure [202]. However, in blockchain-based AS (a decentralized distributed system), encrypted blocks are performed on IoT devices as the sidechains that can be applied as a new strategy to support device authentication in IoT-based systems [203]. Nevertheless, as blockchain-based ASs are still in the exploratory phase for IoT security management, there exist open problems in the implementation of these approaches, i.e., challenges associated with the execution of encrypted blocks due to energy consumption, storage, and computational constraints in edge nodes and/or embedded devices [196].

B. Lessons Learned

Modern ASs are still considered promising for enabling continuous user identification on numerous digital systems. In this empirical study, we have discussed various types of state-of-the-art mechanisms and their susceptibility to different attacks. Eventually, we learned several lessons from this study that might enhance the security and reliability of smartphones and IoT devices as well as formulate the following points.

Performance Evaluation Metrics: In general, the developers lack sufficient technical knowledge concerning weak AS-related security policies and possible vulnerabilities that allow attackers to bypass some of the modern AS. Despite all the numerous existing solutions, standardized performance metrics are needed in this area for evaluating different modern AS in IoT-based systems. While several studies have proposed some metrics, such as simplicity, memorability, and safety, none of these criteria are technically accepted in industry and academia as a standard for evaluating security and reliability of modern AS [184].

The default password strategy and reset option in IoT devices are still the most serious flaws, which could allow Mirai-based botnets to break into the IoT-based systems and cause great damage. To mitigate such vulnerabilities, we recommend that the manufacturers must update their existing devices by eliminating such strategies or produce new generations of IoT machines, which have unique passwords based on a safe reset policy (e.g., OTP). Moreover, new IoT devices should come with safety labels indicating the security level and the measures taken to ensure security. On the other hand, governments should adopt standardized security policies and consider them into account when allowing manufacturers to sell IoT devices in their market [115]. These policies need to be adopted in every country to filter the use of next-generation IoT devices considering their safety measures [191]. However, there are still a massive number of unsecured devices in use within the IoT-based systems around the world that are exposed to various attacks (see Table II, in the supplementary material and Table VIII).

Balance Between Usability and Security: One of the primary goals of modern AS is to enhance the balance between usability and security that allows simplicity in access to the device while providing a high-security level. Several studies based on keystroke dynamics (typing patterns) [147], linguistic profiling, and behavior profiling [181] have been proposed that suffer from high false-rejection rate (FRR) and false-acceptance rate (FAR) values. In practice, these approaches are not very user-friendly, since the input features based on the user's behavior are different in various conditions that cause the device to be locked consecutively. In such mechanisms, the end user must behave the same way as registered behaviors, wherever (s)he unlocks the device. This point is incompatible with the goals of the modern AS considering simplicity which makes it difficult to use each time. Hence, the EER is a kind of measure (%) rate, which both rejects and accepts errors as equals (ERR $=$ [(FAR $+$ FRR)/2]), has to be decreased to address this problem.

Employing the U2F Hardware Tokens for Protecting Sensitive Devices: As we have already summarized in Table VIII, all existing modern AS suffer from vulnerabilities that allow malicious third parties to utilize such backdoors to break the security of smartphones and IoT devices. While it is necessary to protect a sensitive device as safe as possible, we recommend that developers exploit the U2F-based hardware tokens as an efficient option to meet these requirements. However, these tokens need exclusive connectivity ports, such as NFC or USB-C, which limit their applications in some types of IoT devices. Moreover, they are susceptible to some malicious attempts, but the possibility of cracking both authentication factors is an immensely complex task for attackers. Thus, this feature makes such hardware-based AS much safer variants of the modern AS than others.

Considering Computational Complexity When Designing AS for IoT Applications: As there are a variety of different IoT devices (e.g., sensors, edge, and/or embedded devices) with various computational capabilities, developers need to consider the computational complexity as a significant factor while designing a device AS in IoT-based systems [51], [204]. Since most of the state-of-the-art methods have exponential computing cost, they lose their efficiency in real-world IoT applications due to the limited computational power of edge and/or embedded devices (see Table VII).

VI. Conclusion

As the number of smartphone and IoT users rapidly increases, these devices and the sensitive information stored on them will be dramatically targeted by attackers. For example, users deploy smartphones as a central controller for managing different services/machines in IoT-based systems, such as a smart home/building. Hence, vulnerabilities in such devices can be exploited to target the users or the associated data. In general, there exist five types of modern AS, including text-based, graphical-based, biometrics-based, Web-based, and hardware-based solutions, which are associated with behavioral features or knowledge-based metrics to authenticate the user identity. In the empirical investigation in this article, different types of ASs, their vulnerability to attacks, and possible solutions have been studied and discussed in detail from

different perspectives in terms of the deployment in IoT, memorability, and security. Finally, we have highlighted remaining open problems and recommended countermeasures to solve these issues that will draw the attention of cybersecurity researchers/developers for future works.

REFERENCES

[1] W. Luo, Y. Hu, H. Jiang, and J. Wang, "Authentication by encrypted negative password," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 114–128, 2019.

[2] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of Web authentication schemes," in *Proc. IEEE Symp. Security Privacy*, 2012, pp. 553–567.

[3] G. Hu, "On password strength: A survey and analysis," in *Studies in Computational Intelligence*, vol. 721. Cham, Switzerland: Springer, 2017, pp. 165–186.

[4] M. Yildirim and M. Mackie, "Encouraging users to improve password security and memorability," *Int. J. Inf. Security*, vol. 18, pp. 741–759, Apr. 2019.

[5] A. Bianchi, I. Oakley, and H. Kim, "PassBYOP: Bring your own picture for securing graphical passwords," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 3, pp. 380–389, Jun. 2016.

[6] M.-C. Chuang, J.-F. Lee, and M.-C. Chen, "SPAM: A 'secure password authentication mechanism for seamless handover in proxy mobile IPv6 networks," *IEEE Syst. J.*, vol. 7, no. 1, pp. 102–113, Mar. 2013.

[7] X. Liu and Y. M. Cheung, "Learning multi-boosted HMMs for lip-password based speaker verification," *IEEE Trans. Inf. Forensics Security*, vol. 9, pp. 233–246, 2013.

[8] S. Ji, S. Yang, X. Hu, W. Han, Z. Li, and R. Beyah, "Zero-sum password cracking game: A large-scale empirical study on the crackability, correlation, and security of passwords," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 5, pp. 550–564, Sep./Oct. 2017.

[9] M. Alsaleh, M. Mannan, and P. C. Van Oorschot, "Revisiting defenses against large-scale online password guessing attacks," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 1, pp. 128–141, Jan./Feb. 2012.

[10] S. Yang, S. Ji, and R. Beyah, "DPPG: A dynamic password policy generation system," *IEEE Trans. Inf. Forensics Security*, vol. 13, pp. 545–558, 2018.

[11] B. B. Zhu, J. Yan, G. Bao, M. Yang, and N. Xu, "CAPTCHA as graphical passwords—A new security primitive based on hard AI problems," *IEEE Trans. Inf. Forensics Security*, vol. 9, pp. 891–904, 2014.

[12] B. MacRae, A. Salehi-Abari, and J. Thorpe, "An exploration of geographic authentication schemes," *IEEE Trans. Inf. Forensics Security*, vol. 11, pp. 1997–2012, 2016.

[13] H.-M. Sun, Y.-H. Chen, and Y.-H. Lin, "oPass: A user authentication protocol resistant to password stealing and password reuse attacks," *IEEE Trans. Inf. Forensics Security*, vol. 7, pp. 651–663, 2012.

[14] M. M. Christiansen and K. R. Duffy, "Guesswork, large deviations, and Shannon entropy," *IEEE Trans. Inf. Theory*, vol. 59, no. 2, pp. 796–802, Feb. 2013.

[15] E. I. Tatli, "Cracking more password hashes with patterns," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 1997–2012, 2015.

[16] J. Song, D. Wang, Z. Yun, and X. Han, "Alphapwd: A password generation strategy based on mnemonic shape," *IEEE Access*, vol. 7, pp. 119052–119059, 2019.

[17] R. Shay *et al.*, "Designing password policies for strength and usability," *ACM Trans. Inf. Syst. Security*, vol. 18, no. 4, p. 13, 2016

[18] J. Galbally, I. Coisel, and I. Sanchez, "A new multimodal approach for password strength estimation—Part I: Theory and algorithms," *IEEE Trans. Inf. Forensics Security*, vol. 12, pp. 2829–2844, 2017.

[19] J. Galbally, I. Coisel, and I. Sanchez, "A new multimodal approach for password strength estimation—Part II: Experimental evaluation," *IEEE Trans. Inf. Forensics Security*, vol. 12, pp. 2845–2860, 2017.

[20] R. Shay *et al.*, "Can long passwords be secure and usable?" in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2014, pp. 2927–2936.

[21] "Your Home at Your Fingertips." [Online]. Available: http://www.belkin.com/us/Products/home-automation/c/wemo-home-automation/ (Accessed: Feb. 7, 2020).

[22] "Rollertrol Automation Systems." [Online]. Available: https://rollertrol.com/ (Accessed: Feb. 7, 2020).

[23] "Life with Nest Thermostat." Nest. [Online]. Available: https://nest.com/thermostat/life-with-nest-thermostat/ (Accessed: Feb. 7, 2020).

[24] "Hue, Professional Wireless LED Lighting." Phillips. [Online]. Available: http://www.meethue.com (Accessed: Feb. 7, 2020).

[25] "Did You Lock Your Doors This Morning?" Kwikset. [Online]. Available: http://www.kwikset.com/Wireless-Technology/Homeowners/RemoteAccess.aspx (Accessed: Feb. 7, 2020).

[26] Manisha and N. Kumar, "Cancelable biometrics: A comprehensive survey," *Artif. Intell. Rev.*, vol. 53, no. 11, pp. 3403–3446, 2020.

[27] "Works With SmartThings." SmartThings. [Online]. Available: http://www.smartthings.com/product/works-with-smartthings/ (Accessed: Feb. 7, 2020).

[28] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2027–2051, 3rd Quart., 2016.

[29] M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: A literature survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2091–2121, 4th Quart., 2013.

[30] S. Salamatian, W. Huleihel, A. Beirami, A. Cohen, and M. Médard, "Why botnets work: Distributed brute-force attacks need no synchronization," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 2288–2299, 2019.

[31] M. El-hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of Internet of Things (IoT) authentication schemes," *Sensors*, vol. 19, no. 5, p. 1141, 2019.

[32] A. Alzubaidi and J. Kalita, "Authentication of smartphone users using behavioral biometrics," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1998–2026, 3rd Quart., 2016.

[33] B. Choudhury, P. Then, B. Issac, V. Raman, and M. K. Haldar, "A survey on biometrics and cancelable biometrics systems," *Int. J. Image Graph.*, vol. 18, no. 1, 2018, Art. no. 1850006.

[34] W. Meng, D. S. Wong, S. Furnell, and J. Zhou, "Surveying the development of biometric user authentication on mobile phones," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1268–1293, 3rd Quart., 2015.

[35] I. Makhdoom, M. Abolhasan, J. Lipman, R. P. Liu, and W. Ni, "Anatomy of threats to the Internet of Things," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1636–1675, 2nd Quart., 2019.

[36] G. Gardašević *et al.*, "The IoT architectural framework, design issues and application domains," *Wireless Pers. Commun.*, vol. 92, no. 1, pp. 127–148, 2017.

[37] "State of the IoT 2018: Number of IoT Devices." [Online]. Available: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/ (Accessed: Apr. 5, 2021).

[38] B. B. Gupta and M. Quamara, "An overview of Internet of Things (IoT): Architectural aspects, challenges, and protocols," *Concurrency Comput. Pract. Exp.*, vol. 32, no. 21, 2020, Art. no. e4946.

[39] M. Alaa, A. A. Zaidan, B. B. Zaidan, M. Talal, and M. L. M. Kiah, "A review of smart home applications based on Internet of Things," *J. Netw. Comput. Appl.*, vol. 97, pp. 48–65, Nov. 2017.

[40] W. Xu *et al.*, "The Design, implementation, and deployment of a smart lighting system for smart buildings," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 7266–7281, Aug. 2019.

[41] S. Chouikhi, L. Merghem-Boulahia, M. Esseghir, and H. Snoussi, "A game-theoretic multi-level energy demand management for smart buildings," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6768–6781, Nov. 2019.

[42] M. Ashabani, M. A. Latify, H. R. Karshenas, and H. B. Gooi, "Multiobjective autonomous intelligent load control for hybrid single-/three-phase AC/DC smart buildings," *IEEE Trans. Sustain. Energy*, vol. 9, no. 3, pp. 1220—1233, Jul. 2018.

[43] "Can You Trust Your Smart Building?" 2019. [Online]. Available: https://www.iotsecurityfoundation.org/wp-content/uploads/2019/06/IoTSF-Smart-Buildings-White-Paper-PDF-1.pdf (Accessed: Feb. 12, 2020).

[44] C. E. Jiménez, A. Solanas, and F. Falcone, "E-government interoperability: Linking open and smart government," *Computer*, vol. 47, no. 10, pp. 22–24, Oct. 2014.

[45] G. P. Cid, "Smart cities, smart governments and smart citizens," *Int. J. E-Planning Res.*, vol. 4, no. 2, pp. 1–4, 2015.

[46] L. Cui, G. Xie, Y. Qu, L. Gao, and Y. Yang, "Security and privacy in smart cities: Challenges and opportunities," *IEEE Access*, vol. 6, pp. 46134–46145, 2018.

[47] L. P. Galperina, A. T. Girenko, and V. P. Mazurenko, "The concept of smart economy as the basis for sustainable development of Ukraine," *Int. J. Econ. Financ. Issues*, vol. 6, no. 8, pp. 307–314, 2016.

[48] F. Purnomo and H. Prabowo, "Smart city indicators: A systematic literature review," *J. Telecommun. Electron. Comput. Eng.*, vol. 8, no. 3, pp. 161–164, 2016.

[49] S. B. Letaifa, "How to strategize smart cities: Revealing the SMART model," *J. Bus. Res.*, vol. 68, no. 7, pp. 1414–1419, 2015.

[50] Z. U. Shamszaman and M. I. Ali, "Toward a smart society through semantic virtual-object enabled real-time management framework in the social Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2572–2579, Aug. 2018.

[51] M. Wazid, A. K. Das, K. V. Bhat, and A. V. Vasilakos, "LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment," *J. Netw. Comput. Appl.*, vol. 150, Jan. 2020, Art. no. 102496.

[52] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, "Internet-of-Things-based smart environments: State of the art, taxonomy, and open research challenges," *IEEE Wireless Commun.*, vol. 23, no. 5, pp. 10–16, Oct. 2016.

[53] C. Gomez, S. Chessa, A. Fleury, G. Roussos, and D. Preuveneers, "Internet of Things for enabling smart environments: A technology-centric perspective," *J. Ambient Intell. Smart Environ.*, vol. 11, no. 1, pp. 23–43, 2019.

[54] S. Mayer, R. Verborgh, M. Kovatsch, and F. Mattern, "Smart configuration of smart environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1247–1255, Jul. 2016.

[55] I. Docherty, G. Marsden, and J. Anable, "The governance of smart mobility," *Transp. Res. A Policy Pract.*, vol. 115, pp. 114–125, Sep. 2018.

[56] G. Cledou, E. Estevez, and L. S. Barbosa, "A taxonomy for planning and designing smart mobility services," *Govt. Inf. Quart.*, vol. 35, no. 1, pp. 61–76, 2018.

[57] M. Amoretti, L. Belli, and F. Zanichelli, "UTravel: Smart mobility with a novel user profiling and recommendation approach," *Pervasive Mobile Comput.*, vol. 38, pp. 474–489, Jul. 2017.

[58] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on Internet-scale IoT exploitations," *IEEE Commun. Surveys. Tuts.*, vol. 21, no. 3, pp. 2702–2733, 3rd Quart., 2019.

[59] "Attack Landscape H1 2020." F-Secure. [Online]. Available: https://blog-assets.f-secure.com/wp-content/uploads/2020/09/17104329/F-Secure-Attack-Landscape-h1-2020.pdf (Accessed: Dec. 6, 2020).

[60] J. Park and A. Tyagi, "Using power clues to hack IoT devices: The power side channel provides for instruction-level disassembly," *IEEE Consum. Electron. Mag.*, vol. 6, no. 3, pp. 92–102, Jul. 2017.

[61] M. Ge, J. B. Hong, S. E. Yusuf, and D. S. Kim, "Proactive defense mechanisms for the software-defined Internet of Things with non-patchable vulnerabilities," *Future Gener. Comput. Syst.*, vol. 78, pp. 568–582, Jan. 2018.

[62] L. Watkins *et al.*, "Exploiting multi-vendor vulnerabilities as backdoors to counter the threat of rogue small unmanned aerial systems," in *Proc. 1st ACM MobiHoc Workshop Mobile IoT Sens. Security Privacy*, 2018, pp. 1–6.

[63] L. Sha, F. Xiao, W. Chen, and J. Sun, "IIoT-SIDefender: Detecting and defense against the sensitive information leakage in industry IoT," *World Wide Web*, vol. 21, no. 4, pp. 59–88, 2018.

[64] "10 IoT Security Incidents That Make You Feel Less Secure." [Online]. Available: https://www.cisomag.com/10-iot-security-incidents-that-make-you-feel-less-secure/ (Accessed: Feb. 20, 2020).

[65] "When I Load the Xiaomi Camera in My Google Home Hub I Get Stills from Other People's Homes." [Online]. Available: https://www.reddit.com/r/googlehome/comments/eine1m/when_i_load_the_xiaomi_camera_in_my_google_home/ (Accessed: Feb. 20, 2020).

[66] "Ring's Wi-Fi Cameras." [Online]. Available: https://www.courtlistener.com/docket/16630199/1/orange-v-ring-llc/ (Accessed: Feb. 20, 2020).

[67] "Amazon's Ring Video Doorbell Pro." [Online]. Available: https://labs.bitdefender.com/2019/11/ring-video-doorbell-pro-under-the-scope/?adobe_mc=MCMID (Accessed: Feb. 20, 2020).

[68] "Amazon-Owned Blink XT2 Security Camera Systems." [Online]. Available: https://www.tenable.com/security/research/tra-2019-51 (Accessed: Feb. 20, 2020).

[69] "How 1.5 Million Connected Cameras Were Hijacked." [Online]. Available: https://www.vice.com/en_us/article/8q8dab/15-million-connected-cameras-ddos-botnet-brian-krebs (Accessed: Feb. 21, 2020).

[70] A. Maiti and M. Jadliwala, "Light Ears: Information leakage via smart lights," *Proc. ACM Interactive Mobile Wearable Ubiquitous Techn.*, vol. 3, no. 3, pp. 1–27, 2019.

[71] "Faxploit: Breaking the Unthinkable." [Online]. Available: https://blog.checkpoint.com/2018/08/12/faxploit-hp-printer-fax-exploit/ (Accessed: Feb. 21, 2020).

[72] "Oregon FBI Tech Tuesday: Securing Smart TVs." [Online]. Available: https://www.fbi.gov/contact-us/field-offices/portland/news/press-releases/tech-tuesdaysmart-tvs (Accessed: Feb. 21, 2020).

[73] "Smart TVs, Subscription Services Leak Data to Facebook." Google. [Online]. Available: https://threatpost.com/smart-tvs-leak-data/148482/ (Accessed: Feb. 21, 2020).

[74] "Attackers Found a (Not-So-Easy) Way to Make the Amazon Echo a Spy Bug." [Online]. Available: https://www.wired.com/story/attackers-turn-amazon-echo-into-spy-bug/ (Accessed: Feb. 5, 2021).

[75] "The Internet of Thing: How a Single Coffee Maker's Vulnerabilities Symbolize a World of IoT Risks." [Online]. Available: https://blog.avast.com/avast-hacked-a-smart-coffee-maker (Accessed: Feb. 6, 2021).

[76] "How A Coffee Machine Infected Factory Computers with Ransomware." [Online]. Available: https://www.hackread.com/how-a-coffee-machine-infected-factory-computers-with-ransomware/ (Accessed: Feb. 22, 2020).

[77] "Garmin Takes App & Services Offline After Suspected Ransomware Attack." [Online]. Available: https://media.cert.europa.eu/cert/moreclusteredition/en/ITSecurityNews2-2ed43449cdaa07807e53ba0c71bad775.20200724.en.html (Accessed: Dec. 7, 2020).

[78] D. Kumar *et al.*, "All things considered: An analysis of IoT devices on home networks," in *Proc. 28th Security Symp.(USENIX) Security*, vol. 19, 2019, pp. 1169–1185.

[79] "All the Default Passwords of Routers are Accessible." [Online]. Available: https://www.routerpasswords.com/router-password/ (Accessed: Feb. 4, 2021).

[80] W. Zhou, Y. Jia, A. Peng, Y. Zhang, and P. Liu, "The effect of iot new features on security and privacy: New threats, existing solutions, and challenges yet to be solved,"*IEEE Internet Things J.*, Vol. 6, no. 2, pp. 1606–1616, Apr. 2019.

[81] R. Vishwakarma and A. K. Jain, "A survey of DDoS attacking techniques and defence mechanisms in the IoT network," *Telecommun. Syst.*, vol. 73, no. 2, pp. 3–25, 2020.

[82] M. Antonakakis *et al.*, "Understanding the mirai botnet," in *Proc. 26th USENIX Secur. Symp. (USENIX) Security*, vol. 17, 2017, pp. 1093–1110.

[83] N. Vlajic and D. Zhou, "IoT as a land of opportunity for DDoS attackers," *Computer*, vol. 51, no. 7, pp. 26–34, Jul. 2018.

[84] "OVH DDoS Attack." [Online]. Available: https://elie.net/blog/security/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/ (Accessed: Feb. 27, 2020).

[85] "DDoS Attack on KrebsOnSecurity Cost IoT Device Owners." [Online]. Available: https://krebsonsecurity.com/2018/05/study-attack-on-krebsonsecurity-cost-iot-device-owners-323k/ (Accessed: Feb. 27, 2020).

[86] "IoT DYN DoS Attacks." [Online]. Available: https://blog.finjan.com/iot-dos-attacks/ (Accessed: Nov. 27, 2020).

[87] "Liberian Lonester MTN." [Online]. Available: https://www.zdnet.com/article/attacker-bestbuy-sentenced-to-prison-for-operating-mirai-ddos-botnet/ (Accessed: Feb. 27, 2020).

[88] "Deutsche Telekom Routers." [Online]. Available: https://www.databreachtoday.com/mirai-botnet-knocks-out-deutsche-telekom-routers-a-9565 (Accessed: Feb. 27, 2020).

[89] "Massive DDoS Attacks powered by Mirai IoT Bots Continue to Spread into 2017." [Online]. Available: https://www.allot.com/blog/massive-ddos-attacks-powered-by-mirai-iot-bots-continue-to-spread-into-2017/ (accessed: Feb.13, 2020).

[90] "Kaspersky Lab DDoS Intelligence Quarterly Report." [Online]. Available: https://www.kaspersky.com/about/press-release/2018_amplification-attacks-and-old-botnets (Accessed: Feb. 28, 2020).

[91] "DDoS attacks in Q4 2018." [Online]. Available: https://securelist.com/ddos-attacks-in-q4-2018/89565/ (Accessed: Feb. 28, 2020).

[92] "Mirai-Based Botnet Launches Massive DDoS Attack on Streaming Service." [Online]. Available: https://www.securityweek.com/mirai-based-botnet-launches-massive-ddos-attack-streaming-service (Accessed: Apr. 8, 2020).

[93] "DDoS Attacks in Q1-Q4 2019, and Q1-Q4 2020, and Q1-Q2 2021." [Online]. Available: https://securelist.com/ddos-attacks-in-q1-2021/102166/ (Accessed: Nov. 3, 2021).

[94] "IBM X-Force Threat Intelligence Index 2021." [Online]. Available: https://www.ibm.com/downloads/cas/M1X3B7QG (Accessed: Apr. 12, 2021).

[95] N. Li, D. Liu, and S. Nepal, "Lightweight mutual authentication for IoT and its applications," *IEEE Trans. Sustain. Comput.*, vol. 2, no. 4, pp. 359–370, Oct.–Dec. 2017.

[96] Z. Wang, "A privacy-preserving and accountable authentication protocol for IoT end-devices with weaker identity," *Future Gener. Comput. Syst.*, vol. 82, pp. 342–348, May 2018.

[97] P. K. Dhillon and S. Kalra, "A lightweight biometrics based remote user authentication scheme for IoT services," *J. Inf. Security Appl.*, vol. 34, pp. 255–270, Jun. 2017.

[98] P. M. Cao, Y. Wu, and S. S. Banerjee, "CAUDIT: Continuous auditing of SSH servers to mitigate brute-force attacks," in *Proc. 16th USENIX Symp. Netw. Syst. Design Implement.*, 2019, pp. 667–682.

[99] F. Samie, V. Tsoutsouras, L. Bauer, S. Xydis, D. Soudris, and J. Henkel, "Distributed trade-based edge device management in multi-gateway IoT," *ACM Trans. Cyber-Phys. Syst.*, vol. 2, no. 3, p. 17, 2018.

[100] J. Chen, M. Sun, and K. Zhang, "Security analysis of device binding for IP-based IoT devices," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops*, 2019, pp. 900–905.

[101] "The CoAP Protocol is the Next Big Thing for DDoS Attacks." [Online]. Available: https://www.zdnet.com/article/the-coap-protocol-is-the-next-big-thing-for-ddos-attacks/ (Accessed: Apr. 6, 2021).

[102] "Solving The IoT Privacy Problem." [Online]. Available: https://www.pwc.com/us/en/services/consulting/cybersecurity/library/broader-perspectives/solving-iot-privacy-problem.html (Accessed: Mar. 9, 2020).

[103] M. H. Barkadehi, M. Nilashi, O. Ibrahim, A. Z. Fardi, and S. Samad, "Authentication systems: A literature review and classification," *Telematics Informat.*, vol. 35, no. 5, pp. 1491–1511, 2018.

[104] K. Niinuma, U. Park, and A. K. Jain, "Soft biometric traits for continuous user authentication," *IEEE Trans. Inf. Forensics Security*, vol. 5, pp. 771–780, 2010.

[105] W. Kang, Y. Lu, D. Li, and W. Jia, "From noise to feature: Exploiting intensity distribution as a novel soft biometric trait for finger vein recognition," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 858–869, 2019.

[106] Z. Zaheer, A. Khan, M. S. Umar, and M. H. Khan, "One-tip secure: Next-gen of text-based password," in *Information and Communication Technology for Competitive Strategies*. Singapore: Springer, 2019, pp. 235–243.

[107] T.-S. Wu, M.-L. Lee, H.-Y. Lin, and C.-Y. Wang, "Shoulder-surfing-proof graphical password authentication scheme," *Int. J. Inf. Security*, vol. 13, no. 3, pp. 245–254, 2014.

[108] Y. Yang, J. Lindqvist, and A. Oulasvirta, "Text entry method affects password security," in *Proc. LASER Workshop Learning from Authoritative Security Experiment Results*, 2014, pp. 1–10.

[109] N. Shone, C. Dobbins, W. Hurst, and Q. Shi, "Digital memories based mobile user authentication for IoT," in *Proc. IEEE Int. Conf. Comput. Inf. Technol.*, Liverpool, U.K., 2015, pp. 1796–1802.

[110] E. Stobert and R. Biddle, "The password life cycle: User behaviour in managing passwords," in *Proc. 10th Symp. Usable Privacy Security*, 2014, pp. 243–255.

[111] L. Gong, J. Pan, B. Liu, and S. Zhao, "A novel one-time password mutual authentication scheme on sharing renewed finite random sub-passwords," *J. Comput. Syst. Sci.*, vol. 79, pp. 122–130, Feb. 2013.

[112] J. L. Williams, "System and method for automated policy audit and remediation management," U.S. Patent 9 094 434, 2015.

[113] S. A. Kumar and M. S. Anbarasi, "Noble authentication protocol with privacy preservation policy for public auditing on shared data," *J. Comput. Theor. Nanosci.*, vol. 16, no. 8, pp. 3252–3258, 2019.

[114] B. Knieriem, X. Zhang, P. Levine, F. Breitinger, and I. Baggili, "An overview of the usage of default passwords," in *Proc. Int. Conf. Digit. Forensics Cyber Crime*, 2017, pp. 195–203.

[115] "IoT Security Crackdown: Stop Using Default Passwords and Guarantee Updates, Tech Companies Told." [Online]. Available: https://www.zdnet.com/article/iot-security-crackdown-stop-using-default-passwords-and-guarantee-updates-tech-companies-told/ (Accessed: Apr. 6, 2021).

[116] M. A. Khan, I. U. Din, S. U. Jadoon, M. K. Khan, M. Guizani, and K. A. Awan, "g-RAT| A novel graphical randomized authentication technique for consumer smart devices," *IEEE Trans. Consum. Electron.*, vol. 65, no. 2, pp. 215–223, May 2019.

[117] X. Liu, J. Qiu, L. Ma, H. Gao, and Z. Ren, "A novel cued-recall graphical password scheme," in *Proc. IEEE 6th Int. Conf. Image Graph.*, 2011, pp. 949–956.

[118] Z. Zhao, G.-J. Ahn, and H. Hu, "Picture gesture authentication: Empirical analysis, automated attacks, and scheme evaluation," *ACM Trans. Inf. Syst. Security*, vol. 17, no. 4, pp. 1–37, 2015.

[119] P. P. Ray, "Ray's scheme: Graphical password based hybrid authentication system for smart hand held devices," *J. Inf. Eng. Appl.*, vol. 2, no. 2, pp. 1–11, 2012.

[120] Y.-L. Lai *et al.*, "Cancellable iris template generation based on indexing-first-one hashing," *Pattern Recognit.*, vol. 64, pp. 105–117, Apr. 2017.

[121] C. Li, J. Hu, J. Pieprzyk, and W. Susilo, "A New biocryptosystem-oriented security analysis framework and implementation of multibiometric Cryptosystems based on decision level fusion," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 1193–1206, 2015.

[122] Z.-Y. Wu, Y.-J. Tseng, Y. Chung, Y.-C. Chen, and F. Lai, "A reliable user authentication and key agreement scheme for Web-based hospital-acquired infection surveillance information system," *J. Med. Syst.*, vol. 36, no. 4, pp. 2547–2555, 2012.

[123] M. Nauman, T. Ali, and A. Rauf, "Using trusted computing for privacy preserving keystroke-based authentication in smartphones," *Telecommun. Syst.*, vol. 52, no. 4, pp. 2149–2161, 2013.

[124] W. Han, Z. Li, M. Ni, G. Gu, and W. Xu, "Shadow attacks based on password reuses: A quantitative empirical analysis," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 2, pp. 309–320, Mar./Apr. 2018.

[125] E. Erdem, and M. T. Sandıkkaya, "OTPaaS—One time password as a service," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 743–756, 2019.

[126] D. Zhao and W. Luo, "One-time password authentication scheme based on the negative database," *Eng. Appl. Artif. Intell.*, vol. 62, pp. 396–404, Jun. 2017.

[127] N.-W. Lo and A. Yohan, "BLE-based authentication protocol for micro-payment using wearable device," *Wireless Pers. Commun.*, vol. 112, no. 4, pp. 2351–2372, 2020.

[128] "The Best Hardware Security Keys For Two-Factor Authentication." [Online]. Available: https://www.theverge.com/2019/2/22/18235173/the-best-hardware-security-keys-yubico-titan-key-u2f (Accessed: Apr. 5, 2020).

[129] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proc. IEEE Symp. Security Privacy*, San Francisco, CA, USA, 2012, pp. 20–25.

[130] "CrackStation's Password Cracking Dictionary." [Online]. Available: https://crackstation.net/crackstation-wordlist-password-cracking-dictionary.htm (Accessed: Apr. 8, 2020).

[131] "MD5-Hash Cracking With Rainbow Tables and Java." [Online]. Available: https://github.com/lakka/MD5Crack (Accessed: Apr. 8, 2020).

[132] "Android Application to Brute Force WiFi Passwords (Java Source)." [Online]. Available: https://github.com/faizann24/wifi-bruteforcer-fsecurify (Accessed: Apr. 8, 2020).

[133] "A Collection of Social Engineering Tricks and Payloads." [Online]. Available: https://github.com/bhdresh/SocialEngineeringPayloads (Accessed: Apr. 8, 2020).

[134] "Social Engineering Toolkit." [Online]. Available: https://github.com/mwsrc/SEToolkit-Mod (Accessed: Apr. 8, 2020).

[135] "Projekt RC5-72." [Online]. Available: http://www.distributed.net/Main_Page (Accessed: Apr. 6, 2021).

[136] "Brute-Force Attacks." [Online]. Available: https://www.password-depot.de/en/know-how/brute-force-attacks.htm (Accessed: Apr. 10, 2020).

[137] "How Many Possible Combinations in 8 Character Password?" [Online]. Available: https://math.stackexchange.com/questions/739874/how-many-possible-combinations-in-8-character-password (Accessed: Apr. 10, 2020).

[138] "Man-in-the-Middle Attack Framework Used for Phishing Credentials and Session Cookies of Any Web Service." [Online]. Available: https://github.com/wi-fi-analyzer/evilginx (Accessed: Apr. 8, 2020).

[139] S. Ortolani, C. Giuffrida, and B. Crispo, "Unprivileged black-box detection of user-space keyloggers," *IEEE Trans. Dependable Security Comput.*, vol. 10, no. 1, pp. 40–52, Jan./Feb. 2013.

[140] "Java-Keylogger." [Online]. Available: https://github.com/vakho10/Java-Keylogger (Accessed: Apr. 8, 2020).

[141] H. S. Sánchez, D. Rotondo, T. Escobet, V. Puig, J. Saludes, and J. Quevedo, "Detection of replay attacks in cyber-physical systems using a frequency-based signature," *J. Franklin Inst.*, vol. 356, pp. 2798–2824, Mar. 2019.

[142] "SMS-Based-Payment-System." [Online]. Available: https://github.com/saptarshi1729/SMS-Based-Payment-System (Accessed: Apr. 9, 2020).

[143] "LittleProxy—MiTM." [Online]. Available: https://github.com/ganskef/LittleProxy-mitm (Accessed: Apr. 9, 2020).

[144] C. Ntantogian, S. Malliaros, and C. Xenakis, "Evaluation of password hashing schemes in open source Web platforms," *Comput. Security*, vol. 84, pp. 206–224, Jul. 2019.

[145] W. Ali and A. A. Ahmed, "Hybrid intelligent phishing Website prediction using deep neural networks with genetic algorithm-based feature selection and weighting," *IET Inf. Security*, vol. 13, no. 6, pp. 659–669, 2019.

[146] H. H. Nguyen and D. T. Nguyen, "Machine learning based phishing Web sites detection," in *Recent Advances in Electrical Engineering and Related Sciences*. Cham, Switzerland: Springer, 2015, pp. 123–131.

[147] K. A. Rahman, K. S. Balagani, and V. V. Phoha, "Snoop-forge-replay attacks on continuous verification with keystrokes," *IEEE Trans. Inf. Forensics Security*, vol. 8, pp. 528–541, 2013.

[148] "What Is a Replay Attack?" [Online]. Available: https://www.kaspersky.com/resource-center/definitions/replay-attack (Accessed: Apr. 9, 2020).

[149] R. Heartfield and G. Loukas, "A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks," *ACM Comput. Surv.*, vol. 48, no. 3, pp. 1–39, 2015.

[150] P. C. Van Oorschot, A. Salehi-Abari, and J. Thorpe, "Purely automated attacks on passpoints-style graphical passwords," *IEEE Trans. Inf. Forensics Security*, vol. 5, pp. 393–405, 2010.

[151] Z. Rui and Z. Yan, "A survey on biometric authentication: Toward secure and privacy-preserving identification," *IEEE Access*, vol. 7, pp. 5994–6009, 2018.

[152] E. Gonzalez-Sosa, J. Fierrez, R. Vera-Rodriguez, and F. Alonso-Fernandez, "Facial soft biometrics for recognition in the wild: Recent works, annotation, and COTS evaluation," *IEEE Trans. Inf. Forensics Security*, vol. 13, pp. 2001–2014, 2018.

[153] A. Dantcheva, P. Elia, and A. Ross, "What else does your biometric data reveal? A survey on soft biometrics," *IEEE Trans. Inf. Forensics Security*, vol. 11, pp. 441–467, 2016.

[154] M. K. Khan, L. Xie, and J. Zhang, "Chaos and NDFT-based spread spectrum concealing of fingerprint-biometric data into audio signals," *Digit. Signal Process.*, vol. 20, pp. 179–190, Jan. 2010.

[155] D. F. Smith, A. Wiliem, and B. C. Lovell, "Face recognition on consumer devices: Reflections on replay attacks," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 736–745, 2015.

[156] Z. Akhtar, C. Micheloni, and G. L. Foresti, "Biometric liveness detection: Challenges and research opportunities," *IEEE Security Privacy*, vol. 13, no. 5, pp. 63–72, Sep./Oct. 2015.

[157] D. Paul, M. Pal, and C. Saha, "Spectral features for synthetic speech detection," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 4, pp. 605–617, Jun. 2017.

[158] A. K. Jain and K. Nandakumar, "Biometric authentication: System security and user privacy," *Computer*, vol. 45, no. 11, pp. 87–92, Nov. 2012.

[159] V. M. Patel, N. K. Ratha, and R. Chellappa, "Cancelable biometrics: A review," *IEEE Signal Process. Mag.*, vol. 32, no. 5, pp. 54–65, Sep. 2015.

[160] E. Maiorana, G. E. Hine, and P. Campisi, "Hill-climbing attacks on multibiometrics recognition systems," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 900–915, 2015.

[161] B. Nagpal, N. Chauhan, and N. Singh, "SECSIX: Security engine for CSRF, SQL injection and XSS attacks," *Int. J. Syst. Assur. Eng. Manage.*, vol. 8, no. 2, pp. 631–644, 2017.

[162] "How to Fix SQL Injection Using Microsoft .Net Parameterized Queries." [Online]. Available: https://software-security.sans.org/developer-how-to/fix-sql-injection-microsoft-.net-with-parameterized-queriesp (Accessed: May 2, 2020).

[163] M. Srokosz, D. Rusinek, and B. Ksiezopolski, "A new WAF-based architecture for protecting Web applications against CSRF attacks in malicious environment," in *Proc. IEEE Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, 2018, pp. 391–395.

[164] M. Rana *et al.*, "A secure and lightweight authentication scheme for next generation IoT infrastructure," *Comput. Commun.*, vol. 165, pp. 85–96, Jan. 2021.

[165] G. Deepa and P. S. Thilagam, "Securing Web applications from injection and logic vulnerabilities: Approaches and challenges," *Inf. Softw. Technol.*, vol. 74, pp. 160–180, Jun. 2016.

[166] S. Gupta and B. B. Gupta, "Cross-Site Scripting (XSS) attacks and defense mechanisms: Classification and state-of-the-art," *Int. J. Syst. Assur. Eng. Manage.*, vol. 8, pp. 512–530, Jan. 2017.

[167] S. Gupta and B. B. Gupta, "XSS-SAFE: A server-side approach to detect and mitigate cross-site scripting (XSS) attacks in JavaScript code," *Arabian J. Sci. Eng.*, vol. 41, no. 3, pp. 897–920, 2016.

[168] "Cross Site Scripting (XSS) Attack Tutorial with Examples, Types & Prevention." [Online]. Available: https://www.softwaretestinghelp.com/cross-site-scripting-xss-attack-test/ (Accessed: Mar. 6, 2021).

[169] B. K. Ayeni, J. B. Sahalu, and B. K. Ayeni, "Detecting cross-site scripting in Web applications using fuzzy inference system," *J. Comput. Netw. Commun.*, vol. 2018, Aug. 2018, Art. no. 8159548.

[170] M. Van Gundy and H. Chen, "Noncespaces: Using randomization to defeat cross-site scripting attacks," *Comput. Security*, vol. 31, no. 4, pp. 612–628, 2012.

[171] D. Mitropoulos, K. Stroggylos, D. Spinellis, and A. D. Keromytis, "How to train your browser: Preventing XSS attacks using contextual script fingerprints," *ACM Trans. Privacy Security*, vol. 19, pp. 1–31, Aug. 2016.

[172] "SimpleBase Library." [Online]. Available: https://github.com/kobaltkween/SimpleBase (Accessed: May 8, 2020).

[173] "JS-XSS Sanitizer Module." [Online]. Available: https://github.com/leizongmin/js-xss (Accessed: May 8, 2020).

[174] "Content Security Policy (CSP) Quick Reference Guide." [Online]. Available: https://content-security-policy.com/ (Accessed: May 8, 2020).

[175] M.-M. Bidmeshki, G. R. Reddy, L. Zhou, J. Rajendran, and Y. Makris, "Hardware-based attacks to compromise the cryptographic security of an election system," in *Proc. IEEE 34th Int. Conf. Comput. Design*, Scottsdale, AZ, USA, 2016, pp. 153–156.

[176] E. Dauterman, H. Corrigan-Gibbs, D. Mazières, D. Boneh, and D. Rizzo, "True2F: Backdoor-resistant authentication tokens," in *Proc. IEEE Symp. Security Privacy*, San Francisco, CA, USA, 2019, pp. 398–416.

[177] B. Krebs. "Google: Security Keys Neutralized Employee Phishing." [Online]. Available: https://krebsonsecurity.com/2018/07/google-security-keys-neutralized-employee-phishing/ (Accessed: May 15, 2020).

[178] J. Reynolds, "A tale of two studies: The best and worst of yubikey usability," in *Proc. IEEE Symp. Security Privacy*, 2018, pp. 872–888.

[179] G. Varshney, M. Misra, and P. Atrey, "Secure authentication scheme to thwart RT MITM, CR MITM and malicious browser extension based phishing attacks," *J. Inf. Security Appl.*, vol. 42, pp. 1–17, Oct. 2018.

[180] L. Sportiello, "'Internet of Smart Cards': A pocket attacks scenario," *Int. J. Crit. Infrastruct. Prot.*, vol. 26, Sep. 2019, Art. no. 100302.

[181] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, "Continuous user authentication on mobile devices: Recent progress and remaining challenges," *IEEE Signal Process. Mag.*, vol. 33, no. 4, pp. 49–61, Jul. 2016.

[182] S. Mandal, B. Bera, A. K. Sutrala, A. K. Das, K.-K. R. Choo, and Y. Park, "Certificateless-signcryption-based three-factor user access control scheme for IoT environment," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3184–3197, Apr. 2020.

[183] B. Hammi, A. Fayad, R. Khatoun, S. Zeadally and Y. Begriche, "A lightweight ECC-based authentication scheme for Internet of Things (IoT)," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3440–3450, Sep. 2020.

[184] V. Sharma, I. You, K. Andersson, F. Palmieri, M. H. Rehmani, and J. Lim. "Security, privacy and trust for smart mobile-Internet of Things (M-IoT): A survey," *IEEE Access*, vol. 8, pp. 167123–167163, 2020.

[185] X. Li, J. Peng, M. S. Obaidat, F. Wu, M. K. Khan, and C. Chen, "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems," *IEEE Syst. J.*, vol. 14, no. 1, pp. 39–50, Mar. 2019.

[186] S. Banerjee *et al.*, "A provably secure and lightweight anonymous user authenticated session key exchange scheme for Internet of Things deployment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8739–8752, Oct. 2019.

[187] L. Fang *et al.*, "THP: A novel authentication scheme to prevent multiple attacks in SDN-based IoT network," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5745–5759, Jul. 2020.

[188] U. Chatterjee *et al.*, "Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 3, pp. 424–437, May/Jun. 2019.

[189] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2884–2895, Aug. 2018.

[190] J. Heo, Y. Yoo, J. Suh, W. Park, J. Paek, and S. Bahk, "FMS-AMS: Secure proximity-based authentication for wireless access in Internet of Things," *J. Commun. Netw.*, vol. 22, no. 4, pp. 338–347, Aug. 2020.

[191] M. A. Azad, S. Bag, C. Perera, M. Barhamgi, and F. Hao, "Authentic caller: Self-enforcing authentication in a next-generation network," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3606–3615, May 2020.

[192] Y. Liang, S. Samtani, B. Guo, and Z. Yu, "Behavioral biometrics for continuous authentication in the Internet-of-Things era: An artificial intelligence perspective," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 9128–9143, Sep. 2020.

[193] D. Wang, X. Zhang, J. Ming, T. Chen, C. Wang, and W. Niu, "Resetting your password is vulnerable: A security study of common SMS-based authentication in IoT device," *Wireless Commun. Mobile Comput.*, vol. 2018, Jul. 2018, Art. no. 7849065.

[194] P. Punithavathi, S. Geetha, M. Karuppiah, S. H. Islam, M. M. Hassan, and K.-K. R. Choo, "A lightweight machine learning-based authentication framework for smart IoT devices," *Inf. Sci.*, vol. 484, pp. 255–268, May 2019.

[195] M. N. Aman, M. H. Basheer, and B. Sikdar, "Two-factor authentication for IoT with location information," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3335–3351, Apr. 2019.

[196] P. Gope and B. Sikdar, "Lightweight and privacy-preserving two-factor authentication scheme for IoT devices," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 580–589, Feb. 2019.

[197] M. Komeili, N. Armanfard, and D. Hatzinakos, "Liveness detection and automatic template updating using fusion of ECG and fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 13, pp. 1810–1822, 2018.

[198] C. Esposito, M. Ficco, and B. B. Gupta, "Blockchain-based authentication and authorization for smart city applications," *Inf. Process. Manage.*, vol. 58, no. 2, 2021, Art. no. 102468.

[199] "U.S. Attorneys, District of Massachusetts News, Charged Hacker in New York." 2020. [Online]. Available: https://www.justice.gov/usao-ma/pr/new-york-city-man-charged-hacking-credit-card-trafficking-and-money-laundering

[200] A. A.-N. Patwary, A. Fu, S. K. Battula, R. K. Naha, S. Garg, and A. Mahanti, "FogAuthChain: A secure location-based authentication scheme in fog computing environments using blockchain," *Comput. Commun.*, vol. 162, pp. 212–224, Oct. 2020.

[201] A. Vangala, A. K. Sutrala, A. K. Das, and M. Jo, "Smart contract-based blockchain-envisioned authentication scheme for smart farming," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10792–10806, Jul. 2021.

[202] M. Li, H. Tang, A. R. Hussein, and X. Wan, "A sidechain-based decentralized authentication scheme via optimized two-way peg protocol for smart community," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 282–292, 2020.

[203] Z. Cui *et al.*, "A hybrid blockchain-based identity authentication scheme for multi-WSN," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 241–251, Mar./Apr. 2020.

[204] J. Zhang, C. Shen, H. Su, M. T. Arafin, and G. Qu, "Voltage over-scaling-based lightweight authentication for IoT security," *IEEE Trans. Comput.*, early access, Jan. 6, 2021, doi: 10.1109/TC.2021.3049543.

Milad Taleby Ahvanooey (Senior Member, IEEE) received the Ph.D. degree (Hons.) in computer engineering (cybersecurity) from Nanjing University of Science and Technology (NJUST), Nanjing, China, in 2019.

He currently works as a Faculty Member with Nanjing University, Nanjing, where he completed his Postdoctorate Fellowship with the Institute of Multimedia Information Processing, from December 2019 to December 2021. His research interests include the application of AI in cyber security and digital forensics area as well as holds two patent applications.

Dr. Ahvanooey has been distinguished as one of top three Ph.D. talents for China Scholarship Council (CSC) Elite Outstanding Award by the CSC at NJUST, in October 2019. He serves as a Guest Editor of *Cryptography* (MDPI) from December 2021 to August 2022, as a Reviewer Board of *ACM Transactions on Multimedia Computing, Communications, and Applications* and *Computers in Human Behavior* (Elsevier), a Technical Program Committee Member of several International conferences, such as ARES 2021 & 2022, IEEE ICME 2021 & 2022, MIC-Security 2021, and IEEE MIC-Computing 2021.

Mark Xuefang Zhu (Member, IEEE) received the B.E. degree in telecommunications engineering from Xidian University, Xi'an, China, in 1982, the M.S.E. degree in telecommunications engineering from Nanjing University of Posts and Telecommunications, Nanjing, China, in 1990, and the Ph.D. degree in mathematics from the Department of Mathematics, Peking University, Beijing, China, in 1994.

He is a Full Professor and the Head of the Institute of Multimedia Information Processing, School of Information Management, Nanjing University, Nanjing. He has published more than 200 research papers on pattern recognition, speaker identification, multimedia information processing, information security and retrieval, natural language processing, data mining, computer graphics, haptic interaction and VR in public cultural service, and information service.

Prof. Zhu is a Member of the ASIS&T and AAIS.

Qianmu Li (Member, IEEE) received the B.Sc. and Ph.D. degrees from Nanjing University of Science and Technology (NJUST), Nanjing, China, in 2001 and 2005, respectively.

He is a Full Professor with the School of Cyber Science and Engineering, NJUST. He was a Visiting Researcher with FUI, USA, in 2012, and WU, USA, in 2017. His main research interest is in the area of data mining, security, and privacy. He contributed more than 210 papers in topmost international peer-reviewed journals and conferences, and eight books in the above areas.

Prof. Li received the China Network and Information Security Outstanding Talent Award in 2016, and multiple Education Ministry Science and Technology Awards in 2012.

Wojciech Mazurczyk (Senior Member, IEEE) received the B.Sc., M.Sc., Ph.D. (with Hons.), and D.Sc. (Habilitation) degrees in telecommunications from Warsaw University of Technology (WUT), Warsaw, Poland, in 2003, 2004, 2009, and 2014, respectively.

He is a Professor with the Institute of Computer Science, Faculty of Electronics and Information Technology, WUT. He has authored or coauthored two books, over 150 papers, two patent applications, and over 35 invited talks. He has been involved in many international and domestic research projects as a principal investigator or as a senior researcher.

Prof. Mazurczyk served as a Guest Editor of many special issues devoted to network security, such as IEEE TDSC, IEEE S&P, and IEEE Commag. He is serving as a Technical Program Committee Member of RAID, IEEE GLOBECOM, IEEE ICC, IEEE LCN, IEEE CNS, ACSAC, ARES, and ACM IH&MMSec). Since 2016, he has been the Editor-in-Chief of an open access *Journal of Cyber Security and Mobility*. Since 2018, he has been an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and an MCN Series Editor for the *IEEE Communications Magazine*.

Kim-Kwang Raymond Choo (Senior Member, IEEE) received the Ph.D. degree in information security from Queensland University of Technology, Brisbane, QLD, Australia, in 2006.

He currently holds the Cloud Technology Endowed Professorship with the University of Texas at San Antonio (UTSA), San Antonio, TX, USA.

Dr. Choo was a recipient of the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher), the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, the British Computer Society's 2019 Wilkes Award Runner-Up, and several other prestigious awards. In 2015, he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen–Nuremberg. He is the Founding Co-Editor-in-Chief of *ACM Distributed Ledger Technologies: Research and Practice*, and the Founding Chair of IEEE Technology and Engineering Management Society's Technical Committee on Blockchain and Distributed Ledger Technologies. He is an ACM Distinguished Speaker and an IEEE Computer Society Distinguished Visitor from 2021 to 2023, and included in Web of Science's Highly Cited Researcher in the field of Cross-Field in 2020.

Birij B. Gupta (Senior Member, IEEE) received the Ph.D. degree in information and cyber security from the Indian Institute of Technology Roorkee, Roorkee, India.

In 2009, he was selected for Canadian Commonwealth Scholarship awarded by Government of Canada. He has published more than 300 research papers in International Journals and Conferences of high repute. He is also working as a principal investigator of various R&D projects. His biography was selected and published in the 30th Edition of Marquis Who's Who in the World, 2012. His research interest includes information security, cyber security, cloud computing, Web security, intrusion detection, and phishing.

Dr. Gupta received the Young Faculty Research Fellowship Award from MeitY, Government of India in 2017. He is serving as an Associate Editor for IEEE ACCESS and *International Journal of Instrumentation and Control Systems* (Inderscience) and an Executive Editor of *International Journal of Internet of Things and Cyber-Assurance* (Inderscience).

Mauro Conti (Fellow, IEEE) received the Ph.D. degree from Sapienza University, Rome, Italy, in 2009.

After his Ph.D., he was a Postdoctoral Researcher with Vrije Universiteit Amsterdam, Amsterdam, The Netherlands. He is a Full Professor with the University of Padua, Padua, Italy, and an Affiliate Professor with the University of Washington, Seattle, WA, USA. In 2011, he joined as an Assistant Professor with the University of Padua, where he became an Associate Professor in 2015, and a Full Professor in 2018. His research is also funded by companies, including Cisco, Intel, and Huawei. His main research interest is in the area of Security and Privacy. In this area, he published more than 280 papers in topmost international peer-reviewed journals and conferences.

Prof. Conti has been awarded with a Marie Curie Fellowship in 2012 by the European Commission, and with a Fellowship by the German DAAD in 2013. He currently serves as the Editor in Chief of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, an Area Editor-in-Chief for IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and an Associate Editor for several journals, including IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, and IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT. He was the Program Chair for TRUST 2015, ICISS 2016, and WiSec 2017 and the General Chair for SecureComm 2012 and ACM SACMAT 2013.