Review of Networking and Tangible Security Techniques for Domestic IoT Devices and Initial Ideas

Sameh Zakhary, Neelima Sunil, and Derek McAuley

Index Terms

Home networking, Internet of Thing, Tangable security, Public/private key security, Usability

I. Introduction

The number of connected devices including Internet of Things (IoTs) on the Internet is growing fast. According to recent Gartner research, the estimated number of IoT devices is 5.8 billion in 2020 (Gartner, 2019). The countries that are leading the way to IoT deployment include North America, Western Europe and China (Kandaswamy and Furlonger, 2018). By 2024, the number of Machine-2-Machine (M2M) connections between these devices are expected to reach 27 billion in 2024 (Kandaswamy and Furlonger, 2018). This growth in M2M connectivity is expected to result from wide range of application areas such as smart cities, smart infrastructure, smart energy among many others (Hassija et al., 2019).

This wide spread of IoTs has sparked significant research interest to understand various implications (Airehrour et al., 2016; Neshenko et al., 2019; Hassija et al., 2019). IoTs enable the integration between many objects in our daily life (Aazam et al., 2016; Alaba et al., 2017) such as sensors, objects, wearable devices and other types of machines. IoT devices are capable of communicating directly with one another and sharing data without direct human intervention (Crabtree et al., 2018). These “things” could be any traditional objects such as home appliance (*e.g.* microwave, fridge) or tiny sensor (*e.g.* humidity or health sensors). The devices are capable of constant collections of various sensitive and personal data about many aspects of our lives due to its pervasive deployment (Ren et al., 2019).

This paper provides an overview of the literature relating to securing IoT with an emphasis on usability from a user perspective as well as approaches to securing access to these devices over the Internet. Although IoT deployment occurs in various settings, *i.e.* industrial IoT deployment, we mainly focus in this paper on private residential home deployment (*i.e.* consumer IoTs). We assume that in such settings, users are mostly not experts in security IoT or the underlying networking principles.

This paper is organized as follows: section II discusses various protocols and networking security tools (*e.g.* firewall and Virtual Private Network (VPN)). Section II-D discusses various approaches to simplify cyber-security by using user-centred approaches. In section III, we present a number of existing including *enterprise*-grade solutions that could be adopted to secure remote access to IoT devices in domestic settings.

Corr. author: Sameh Zakhary
Horizon Digital Economy Research
University of Nottingham Innovation Park,
Triumph Road, Nottingham, NG7 2TU, UK.
email: sameh.zakhary@nottingham.ac.uk

(M1) LITERATURE REVIEW OF PREVIOUS ATTEMPTS (BOTH NETWORKING AND USER INTERACTION)

II. IoT NETWORKING

A number of messaging protocols have been developed to meet the various requirements of IoT deployments (Naik, 2017). There are mainly four messaging protocols widely accepted for IoT systems: Message Queuing Telemetry Transport Protocol (MQTT), Constrained Application Protocol (CoAP), Advanced Message Queuing Protocol (AMQP) and Hyper Text Transport Protocol (HTTP). The previous protocols sits above the transport layer (TCP/UDP) protocols. Unlike the Internet which has a single standardized messaging protocol HTTP, IoT has different messaging protocols as currently no single protocol is able to meet all the applications requirements Naik and Jenkins (2016).

On the physical networking layer, many standards have emerged to help connect various IoT devices. Some of the traditional standards are Cellular, IEEE 802.11 (Wi-Fi), Zigbee and Bluetooth Low Energy (BLE). Low Power Wide Area Networkss (LPWANs) represents a number of emerging standards for wireless communication between IoT devices such as Long Range (LoRa), *SIGFOX* (SIGFOX, 2020), *GPRS* and Narrowband-IoT (NB-IoT) (Vejlgaard et al., 2017). For consumer IoT in a home settings, Zigbee is an open-standard for wireless communicating between devices that ranges between $10\,\mathrm{m}$ and $100\,\mathrm{m}$. It has been widely used , and adopted by various vendors to enable communication for consumer devices (Such as Philips Hue and Amazon Alexa).

A. *Network traffic control mechanisms*

There exists many networking security tools that can be used to control the follow of traffic in a network (Mortier et al., 2012; Neshenko et al., 2019). Virtual Private Network, firewalls, Wireless Local Area Network (WLAN) security standards (Wireless Protected Access 2 (WPA2) personal & enterprise) Mortier et al. (2012) show that domestic users seek to better understand and control the different flows within their home network. With the advancement and wide spread of IoTs, home networks are faced with a much greater challenge to secure these many of these devices (Neshenko et al., 2019). Researchers have been assessing the security issues associated with some of these IoT devices, and indeed many have been found to have major security flows (*e.g.* self-signed Secure Socket Layer (SSL) certificates, default credentials or out-of-date software) (Costin et al., 2014). Controlling access of outsiders (*i.e.* through the Internet) is one technique employed to prevent remote exploitation of some of these vulnerabilities and reducing the attack surface of some of a IoT devices.

Home users are usually offered to configure a simple front end firewall interface on a local web-based interface running on the home router or a cloud-based portal. In the case where the home router is running a linux-based operating system, this mostly translates into an iptables (netfliter) rules to control inbound/outbound traffic flows. Linux *iptables*/netfilters is one of the most commonly and well established mechanisms for packet filtering for decades. More recently, extended Berkeley Packet Filter (eBPF) had gained more attention as an alternative for offering packet filtering that meets future server deployment and virtualization scenarios that require additional scalability as the *iptables* rules significantly grow (Bertrone et al., 2018).

Guest WLAN[1] is a feature that exists in commercial as well as open-source router firmware such as *OpenWRT* Operating Systems (OSs). Guest WLAN uses wireless protocols standards -IEEE 802.11 (Wi-Fi)- to allows connected devices to have access to the network but in a more restricted way. The objective is to create a *Guest* network that is easy to connect to by devices that are largely isolated from the main

[1] OpenWRT Guest WLAN basic : https://openwrt.org/docs/guide-user/network/wifi/guestwifi/guest-wlan

network. Most guest networks will provide access to the Internet, while restricting access to the Local Area Network (LAN) in order to minimizes the impact in-case one of the connected devices is compromised. Further more, some guest networks could be provided so devices on this network will be prohibited from routing *any* traffic whether on the WLAN nor the LAN, further restricting how a malicious device can attack other devices on the guest network. Guest network could be subject to additional monitoring and Quality of Service (QoS) restriction to avoid miss-use (*e.g.* a malicious device consuming significant bandwidth and causing starvation to other devices on the LAN).

Additionally, domestic routers usually do not offer ways to monitor bandwidth usage by each devices nor the ability to control the usage. This could become an issue when IoT start contending for bandwidth or trigger high bandwidth operations (*e.g.* OS update, sync operations) that impacts other interactive or real-time services (*e.g.* live streaming or gaming). For these situations, *traffic shaping* (also referred to as packet shaping) offers a useful means to control and limit the network bandwidth usage of devices or traffic type. Traffic shaping enables delaying or restricting the rate of a specific type of network packets according to a defined priority which translate to limiting the bandwidth consumed by the associated traffic. This features is mainly targeted towards enterprise grade solutions and high-end routers where it ensures a better QoS for business-related traffic and interactive applications.

B. Solution for securing remote access

Completely blocking Internet access to IoT devices or limiting its interactions to only in-door environment might be possible for some category of device, but other IoT devices such as CCTV monitoring systems might need to be monitored while occupants are away from their home. There exists multiple solution offering VPN-related (mostly used in enterprise network to enable remote or tele-working) that could enable secure access to local network resources. VPN approaches could be achieved using various existing security protocols and tools such as IP Security (IPsec), OpenSSH, *OpenVPN* and WG.

IPsec follows a layering architecture that enables each of the layers to deal with a dedicated set of protocol functions (*i.e.* key exchange, data transport, encryption, interface, etc.) It relies on using Linux transform ("xfrm") layer which allows users to select cipher algorithm and key in addition to other protocol parameters. The key-exchange stage (usually using IKEv2 (Kaufman et al., 2010)) of the protocol updates various data structures which is used by the following data transport stages. IPsec protocol is complex to setup correctly and requires additional firewall semantic and configuration for it to function behind a firewall.

To avoid some of these issues, other approaches have been developed for VPN. For example, *OpenVPN* is TUN/TAP based solution running in the user space (*e.g.* not inside the Linux kernel). It offers a VPN solution that relies on Transport Layer Security (TLS) and is available on many Linux OS distributions. It requires a running daemon responsible for managing the logical network interface (*tun0*). *OpenVPN* is quite complex due to supporting many of the TLS features and functionality. TLS is difficult to manage due to its huge state machine.

Donenfeld (2017) developed WG, a secure OSI layer 3 VPN protocol. It is a relatively new VPN protocol that has recently been merged into the *Linux* kernel V5.6 (Torvalds, 2020). WG offers faster throughput and modern cryptographic support as well as much lower overhead which offers better battery life. As WG is mostly stateless, it is able to support better roaming compared to state-full solutions. As it has been merged into the Linux kernel, it offers a much lower overhead and much higher bandwidth compared to user-space VPN options. Key distribution is not covered by WG protocol, and WG implementation in Linux is mostly agnostic to how users would exchange cryptographic material. This is a similar approach to *OpenSSH*, where users can exchange their public keys Out-of-Band (OOB) to setup secure connectivity. It worth noting that these keying material is used to drive other ephemeral keys during the actual communication over the VPN tunnel. We will look at WG in further details in section II-C.

Some of the VPN solutions we discussed above have been analysed and compared in the literature. Pudelko et al. (2020) have compared the network performance of *OpenVPN*, Linux IPsec and WG. Author show that WG has demonstrated the best performance in terms of high-throughput under their proposed performance measurement scenario (site- to-site setups) using pipelining. Since key exchange required for setting up the public keys for the peers is not covered by the protocol offers, we are performing additional research to exchange/drive these keys based on tangible interactions in the domestic settings in this project.

C. WireGuard: Kernel-space state-of-the-art VPN

Donenfeld (2017) has developed WireGuard, a secure VPN protocol that incorporated modern cryptographic algorithm. Similar to other VPNs protocols, WG offer a secure communication over insecure channels to authenticated parties (called *peers*). WG doesn't have the concept of a client and server, hence the use of the name *peers*, but instead have *Initiator* and *Responder* to refer to the peer that starts the communication, and the peer that accept the communication respectively. These roles could change at any stage using WG, so a given *peer* could be the *Initiator* at one time, but later becomes the *Responder* when it gets contacted at a later stage and vice versa.

WG has gained considerable attention in research (Lipp et al., 2019; Pudelko et al., 2020) as well as the wider networking community with its merge into the *Linux* kernel (Torvalds, 2020). It offers a number of advantages over other existing and well established VPN solutions, mainly its high performance and simple code base. The fact that WG is developed and integrated directly into the kernel space (not clearly separating the functions performed to adhere to the Open Systems Interconnection (OSI) layers (Standardization, 1996)) enabled many performance advantages including reduced overhead in coping packets across different processes (residing in different kernel and user spaces).

WG provides a secure virtual interface at the OSI layer 3 which relies on *Cryptokeys routing*. This term is used by WG to refer to the secure and cryptographic-backed mechanism by which the interface will authenticate incoming packets to peers' keys and encrypt outgoing packets with the key associated with the corresponding peers public key. The two peers form a tunnel using User Datagram Protocol (UDP) for transporting packets for all network flows arriving from higher layers addressed to the give WG interface. The packets arriving from upper layers to an address managed by this interface (at L3) are securely encrypted and encapsulated over in UDP packets to the latest known external peer's Internet Protocol (IP) address. Authenticated Encryption with Additional Data (AEAD) schema is used to protect data sent over the tunnel by providing confidentiality, integrity, and authenticity. In brief, the IP packets arriving at the interface are encrypted/signed using ephemeral keys and encapsulated into a newly created UDP packet and WG port of the corresponding peer's WG interface using the last known IP of this peer-. These processes are done behind the scene with the user not involved in tracking peers' IP, session initialization including producing derived keys nor triggering re-keying (*i.e.* future *Handshake* process after various timeouts events).

Figure 1 shows the high-level sequence of operations during setup and later communication between peers. To setup a WG peer, the two entities need to exchange 32-byte Curve25519 public keys out-of-band. This enable each of the peers to be uniquely identified by its public key, this further protects peers from Denial of Service (DOS) attacks as peers will never respond except if the sender proves that he/she knows their public key. At the start of every communication, the peer that wishes to send data (called the *Initiator*) starts the *Handshake* stage, and the other peer verifies and only respond to legitimate *Handshake Initiation* packet with a *Handshake Response*.

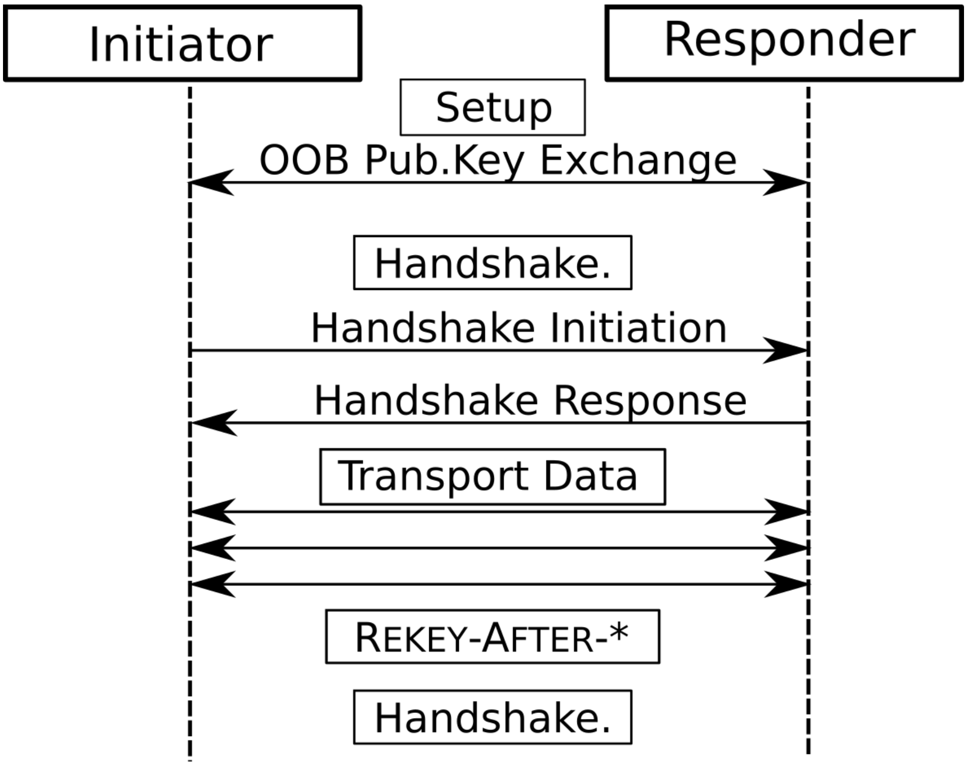


Fig. 1: WireGuard high-level typical sequence of operation between two peers (setup including OOB key-exchange required once for new peer).

Handshake stage is performed using the same UDP protocol port number as the one used for data transport. During the 1.5-Round-Trip Time (1.5-RTT), the *Initiator* can start sending encrypted messages as soon as it receives the handshake response, while the responder may only send encrypted data once it has received an Acknowledge (ACK) from the Initiator confirming that it has indeed received the response. This is in order to prevent replay-attacks, where an adversary could replay older handshake messages to cause the responder to regenerate its ephemeral key and invalidating the key of the legitimate Initiator. To mitigate against such attack, WG includes 12-byte TAI64N (Bernstein, 1997) that is encrypted and authenticated. The greatest time stamp is stored by the *responder* for values received from each peer so as to discard any packet that holds values less than or equal to it.

The Noise Protocol Framework (Perrin, 2018) is used during the handshake stage to ensure mutual authentication, key agreement and forward secrecy. This prevents adversaries (un-authenticated or malicious parties) from scanning various networks and hosts for the WG port without knowing the peer long-lived public key. Although WG public keys are not meant to be kept secret (as they get exchanged with other peers), it is important to note that these keys are not expected to become public (*i.e.* published similar to a Pretty Good Privacy (PGP) key) in order to prevent from DOS attacks.

After a period of communication (timeouts) or a number of messages have been exchanged peers will re-enter into a new *handshake* stage to change the ephemeral (short-lived) keys currently being used. After this stage, the two peers will resume transporting data using the new keys. After this 1.5-RTT key exchange handshake, the peers can use the agreed pair of symmetric keys (one for sending and one for receiving) to send/receive data over the tunnel.

D. User Interactions

With the wide variations of IoT devices functionality and capabilities, designing a suitable and user-friendly security mechanism becomes a challenging task. The complexity of the many existing approaches and techniques of security solutions for devices, network and application could cause a number of exploitable vulnerabilities. A plethora of research has looked into techniques for enhancing a Human Computer Interface (HCI), for example using "Proxemic interactions" Marquardt and Greenberg (2012). These interactions can be classified into one of five dimensions (*i.e.*)distance, orientation, movement, identity, and location). We mainly focus in this section on how to leverage these users' interactions with the IoT devices to offer more secure communication (*e.g.* for security initialization, user authentication and authorization). Below, we will review and present a summary of some of the most interesting and relevant research.

1) Use of Extra Portable Device: Password-based authentication has been criticized for their usability and security issues (Payne et al., 2016) and many IoT vulnerabilities have been attributed to the use of default credentials (Neshenko et al., 2019). To ease the use of passwords and help users to manage the growing number of services, a number of solutions have been developed to help users store and retrieve them.

Karole et al. (2010) present a comparative study assessing the usability of three of the most popular password managers (namely, *LastPass*, *KeePassMobile* and *Roboform2Go*). *LastPass*, an online password managers, had a lowered trust since it is managed by third parties that store passwords on remote servers. *KeePassMobile* is a phone-based password managers. *Roboform2Go* is a USB-based password managers that needs to be directly inserted into terminals. The fact that users need an "extra" device to carry is a drawback and not all devices have Universal Serial Bus (USB) ports to plug into. The study concludes that using password manger on a mobile device is more convenient and had an increased trust (owing to phones being carried and controlled by the user). On the other hand, password managers has a usability drawback as users need to transfer the password from the manager to the system they are authenticating against on the same or another device (*e.g.* web-application, VPN client).

Another approach to authentication is token-based. Payne et al. (2016) have conducted a serious of semi-structured interviews using *grounded theory* to evaluate the use of portable token-based security devices. In their study, they have focused on evaluating the use of *Pico* (Stajano, 2011). *Pico* is a token-based scheme (prototype) that makes authentication more tangible using multiple wearable devices). The study concludes that it increases perceptions of personal responsibility for mitigating security risks and managing physical items, which is potentially inconvenient and anxiety-provoking for users. To overcome these, they suggested three methods: 1) Increase willingness to take on more personal responsibility for security by reducing annoyance and anxiety, 2) Avoid putting system failures (issues of reliability) in the hands of the user, 3) Find a way of aligning potential users' mental models of *Pico* with how it actually works to reduce the likelihood that the scheme will be rejected before it is tried .

As an alternative to carrying a dedicated device for authentication, an open industry alliance—including PyaPal, Microsoft and Visa—has released specifications for online authentication Fast IDentity Online (FIDO). This includes two sets of specifications for online authentication: Universal Authentication Framework (UAF) and Universal Second Factor (U2F). Both involve authenticating to online services with a device as a second form of authentication. 1) UAF replaces passwords by allowing users to authenticate from a FIDO-enabled device (*e.g.* a smartphone). It involves registering the user's device to online services and selecting a biometric authentication action, such as swiping a finger, performed on the device, 2) U2F replaces the second factor in two-factor authentication (2FA) with a dongle that the user inserts into a

USB port. After typing a password, the user is asked to press a button on the dongle to authenticate to services.

2) Opinions: We have extracted the following findings from the literature to inform our future prototype and possibly support the design in future:

From a usability presepective, users generally not like 2FA when they are asked to carry a dedicated device to perform this step. De Cristofaro et al. (2013) found that dedicated tokens were the least used method of 2FA: almost 90% of participants used email or SMS; approximately 45% used a phone app; less than a quarter used tokens. Most used a hardware token only when enforced upon (by their employer or bank). Krol et al. (2015) findings further supports this points, as they found that carrying and operating a hardware token for 2FA with online banking negatively correlated with satisfaction.

But physical tokens tend to be resilient to physical observation and guessing (Bonneau et al., 2012). They also alleviate memory issues like remembering passwords and linking the right ones to the right accounts. There are, however, important drawbacks. Tokens require users to carry an additional item—a usability issue (Bonneau et al., 2012). There is also the potential for loss and theft. This is both an inconvenience (in terms of token-recovery) and a security challenge (Bonneau et al., 2012; Karole et al., 2010): if physical tokens are not always something that users really “have”, then unauthorized persons could access services that belong to someone else.

3) Gestures: Using gestures to authenticate users has been investigated due to the pervasive deployment of wearable and smart devices (such as mobile phones, smart glasses). This section presents a brief overview at some of the available security options that employ gestures.

Research has looked into using mobile devices for authenticating end-users, for example by inferring user-specific and uniques attributes. Leung et al. (2018) have presented *TwistIn*, a tangible authentication of a smart devices based on motion co-analysis with another wearable device (*i.e.* smartwatch). It captures a simple gesture movement from a smartwatch as an authentication token to unlock other smart devices (*e.g.* mobile phone). To unlock a device, the user would pick it up and performs a few twists which is co-analyzed by the smartwatch and the device which then extends the authentication from the watch to the device.

For mobile authentication, using grid patterns and PINs have been used to speed-up authentication and increase memorability, but they offer less secure alternative (less entropy) compared to strong password. Free-form gesture passwords have been proposed to offer better security and usability for mobile authentication, but avoids the drawback of text-based passwords. Recently, some research has been conducted to understand how users actually utilize free-form gestures and what benefits it offers and what are its drawbacks. Yang et al. (2016) have conducted a study to evaluate the use of free-form gesture passwords. The study compares the use of free-form gestures and text passwords as a baseline. The study show that free-form gestures outperformed text passwords in mobile authentication as participants were able to better memorize and authenticate faster using them. A remark author has made is that the collected dataset of gestures from the participants showed a bias towards common shapes, which might affect the security. In real life, this means that there might be fewer shapes an attacker needs to guess from before he/she is able to crack the free-form gesture password. To elaborate, we present in Table I a summary and grouping of some related studies into gestures-based authentication.

4) Other approaches:

- Camera-Based Systems: *Pixie*, a novel, camera based two factor authentication solution for mobile and wearable devices. A quick and familiar user action of snapping a photo is sufficient for Pixie to simultaneously perform a graphical password authentication and a physical token based authen-

TABLE I: A list of recent related gesture-based techniques.

Scheme	Technique	Description
(Li, 2010)	multi-touch	Protractor: information theoretical metric to quantify the security of free-form gestures, a way to authenticate multi-touch gestures
(Leung et al., 2018)	3D gestures	user performs a few twists of a device that are co-analyzed to unlock it
(Tian et al., 2013)	3D gestures	KinWrite: users perform gestures in 3D (midair) for authentication
(Aumi and Kratz, 2014)	3D gestures	AirAuth: 3D midair gesture recognition method
(Sahami Shirazi et al., 2012)	3D magnetic	3D magnetic gesture recognition system
(De Luca et al., 2013)	2D gestures	2D gestures collected on Back-of-device (BoD) using two phones connected back to back
(De Luca et al., 2014)	stroke-based	XSide: stroke-based authentication mechanism that uses front or back of smartphones
(Zheng et al., 2014)	tapping gestures	authentication system by recognizing user's tapping password behavior based on a list of features like acceleration, pressure, size and time collected during authentication
(De Luca et al., 2012)	Touch pattern	authentication scheme based on a user's touch pattern
(Burgbacher and Hinrichs, 2014)	gesture keyboards	authentication scheme based on gesture keyboards

tication, yet it does not require any expensive, uncommon hardware. Pixie outperformed text based passwords on memorability, speed, and user preference. It was also easily discoverable by new users and accurate under field use (Azimpourkivi et al., 2017).

- *Acoustics* and *Vision*: *EchoPrint*, leverages acoustics and vision for secure and convenient user authentication, without requiring any special hardware. *EchoPrint* actively emits almost inaudible acoustic signals from the earpiece speaker to “illuminate” the user's face and authenticates the user by the unique features extracted from the echoes bouncing off the 3D facial contour (Zhou et al., 2018).
- Radio-Frequency Transmitters: *PhoneAuth*, a system intended to provide security assurances comparable to or greater than that of conventional two-factor authentication systems while offering the same authentication experience as traditional passwords alone (Czeskis et al., 2012).
- Acoustic Signals : Proximity-Proof, a secure and usable mobile 2FA system without involving user interactions. Proximity-Proof automatically transmits a user's 2FA response via inaudible OFDM-modulated acoustic signals to the login browser. The results of a user study indicate Proximity-Proof is very easy to use, unobtrusive, and more preferable than Duo (Han et al., 2018).
- Location-IoT : *Icelus*, a system that uses user locations as an additional factor of authentication in scenarios where physical presence is required, such as when making in-person purchases or unlocking a vehicle. It leverages the increasing number of IoT devices carried and used by users and the

smart environments that observe these devices. It also exploits the ability of many IoT devices to "sense" the user. The experiments with it show that it exhibits a smaller false-rejection rate than smartphone-based location-based authentication (LBA) and it rejects attackers with few errors (i.e., false acceptances) (Agadakos et al., 2016).

- Sound-Proof, a usable and deployable two-factor authentication mechanism. Sound-Proof does not require interaction between the user and his phone. In Sound-Proof the second authentication factor is the proximity of the user's phone to the device being used to log in. The proximity of the two devices is verified by comparing the ambient noise recorded by their microphones (Karapanos et al., 2015).
- Listening-Watch, a new 2FA mechanism based on a wearable device (watch/bracelet) and active browser- generated random speech sounds. As the user attempts to login, the browser populates a short random code encoded into speech, and the login succeeds if the watch's audio recording contains this code (decoded using speech recognition), and is similar enough to the browser's audio recording. The remote attacker, who has guessed the user's environment or created predictable phone/watch sounds, will be defeated since authentication success relies upon the presence of the random code in watch's recordings (Shrestha and Saxena, 2018).

(M2) UNDERSTAND APPLICABILITY OF EXISTING (ENTERPRISE) SYSTEMS AND THEIR SCOPE FOR REUSE

III. APPLICABILITY OF EXISTING VPN SOLUTIONS

In the market for enterprise VPN solutions, there many exists vendors. The services are ranging from private to public VPNs. The private VPN could be either on-premises or cloud-based VPN (*e.g. VPN-as-a-Service*) dedicated to a particular client (such as a major bank). Most enterprise-scale network equipment manufactures include VPN, such as Cisco Network Convergence Series Routers. In case of public or shared VPN, the infrastructure is shared to more public VPN solution and is billed as per the usage (*e.g.* number of connections, clients or bandwidth).

In addition to providing the infrastructure to connect to one of the VPN servers securely, enterprise solutions tends to offer policy and centralized access management to administrators of large organizations (required for compliance and auditing purposes). We have also surveyed some of the key features thought after for enterprise deployment, and we show the features and add important features for home deployment: 1) Ease of deployment into the existing infrastructure, 2) Integration (or compatibility) with IoT, 3) Management and alerting tools, 4) Support during the appliance life, 5) Overall cost (*e.g.* licensing, support and hardware costs), 6) and Mobility support.

Support for mobile end-devices to access the VPN as the devices get handed-over across networks is referred to as "mobile VPN". For some VPN deployment scenarios, "Mobility" support is not important and might contribute to unnecessary overheads. One example is when a company wants to integrate a remote branches subnets into the larger company network to provide seamless access inside any of the branches. Such scenario requires an enterprise VPN solutions that is focused on wired end-to-end systems with high-speed and reliable infrastructure (referred to as site-to-siteVPN).

On the other hand, "Mobility" support is a corner requirement for most scenarios where an enterprises is enabling remote workers to access local network resources. Also, a usable VPN solution for a domestic network must support mobility as users are mostly roaming on other networks who wish to access their network without degradation of performance or overhead in establishing VPN connections repeatedly.

In addition to the above features, we evaluate the configuration choice supported by each of the possible VPN solutions. The VPN solutions would offer one or more of the configurations listed in table II depending on how the VPN and client are configured. In an enterprise, the choice of which configuration is directly affected by the ownership model of the mobile device which is used to access the VPN. In other words, the level of control and impact on other installed applications on the mobile device could limit which configuration to use for an employee-owned device. For example, an *always-on* VPN" offers the highest security and control for system administrators as it captures all traffic originated from the mobile device and direct it towards the enterprise network. This traffic then is subject to the security policy of the enterprise, such as rules of filtering, blocking and monitoring. This is obviously not suitable when the device is owned by the employee who sometimes need to access limited enterprise services (*e.g.* enterprise email account).

TABLE II: A possible VPN configuration options.

Configuration	**Description**
Standard VPN	Usually requires dialing into a VPN server where all network traffic from the client is routed through the VPN server. This enables monitoring and rule enforcement but causes degradation in performance as the VPN server becomes a single point of failure.
On-demand VPN	VPN reconnect automatically upon accessing some corporate resource. This enables mobile device to not incur significant delays when accessing other services outside the enterprise.
Per-app VPN	This VPN configuration allows specific mobile devices' applications to access the corporate VPN server for a specific service (such as emails, file server or sales system). This is similar to the on-demand configuration but focuses on a mobile application instead of a defined network subnet(s).
Always-on VPN	This configuration enables the device to always be connected the corporate network through a VPN which offers the highest security and control over the devices' network traffic. This enables full monitoring and control over what flows are enabled or which services are accessible to the mobile device. The VPN is started automatically when the device is booted and stays on capturing all traffic (*i.e.* locking-in the device).

Alshalan et al. (2016) presented a survey of "mobile VPN" technologies including key software solutions at the date of publication. We revisit the list of enterprise solutions to reflect the recent changes. In this report, we focus on the technologies in the market for enterprise VPN solutions that support *Mobility*. Based on the aforementioned configuration options, we have evaluated the key players in the **enterprise mobile VPN market** in terms of the possible configuration(s) and reliance on dedicated hardware/software server in table III.

For consumer cloud-based VPN solutions, there are many vendors offering the user a way to tunnel the network traffic through their servers (for privacy or security when using un-trusted networks), for example, *NordVPN*[2] and *ExpressVPN*[3]. Looking specifically at available open-source VPNs technology, namely *OpenVPN* and WG), there exists a paid enterprise or consumer cloud-hosted related service, *OpenVPN* for business and Tailscale[4] respectively. In table IV, we present a short comparison between *NordVPN* and *OpenVPN*

There are specific enterprise-level features (*e.g.* single sign-on, remote access) while other features are useful to all users (*e.g.* anonymous browsing, peer-to-peer). Single sign-on refers to the ability for

[2] NordVPN

[3] ExpressVPN

[4] Tailscale Github

TABLE III: Key enterprise mobile VPN solutions: summary of common deployment option and configurations.

Enterprise VPN	Summary
AnyConnect VPN (Cisco Inc., 2020)	➢ TCP-based application access or Datagram Transport Layer Security (DTLS) ➢ always-on, on-demand or per-app VPN
(Citrix Systems, 2020) SSO	➢ layer 3 SSL connectivity ➢ requires Citrix Gateway ➢ always-on, on-demand or per-app configuration
Mobile Connect (SonicWall, Inc., 2020)	➢ SSL-based VPN ➢ requires SonicWall Secure Mobile Access (SMA) or Next-Generation Firewall ➢ on-demand or per-app conf.
Connect Secure (Pulse Secure, 2020)	➢ SSL-based VPN solution ➢ requires a VPN gateway ➢ always-on or per-app conf.
Workspace ONE (VMWare Inc., 2020)	➢ requires Workspace ONE Assist Server ➢ on-demand or per-app conf.
Synopsis (Radio IP Software, 2020)	➢ requires a gateway ➢ standard conf.
(NetMotion, 2020) VPN	➢ requires NetMotion gateway ➢ always-on or on-demand conf.
WireGuard (Donenfeld, 2017)	➢ no specific server (uses peers) ➢ on-demand (specific WG peers' IPs/keys) conf.
(OpenVPN Inc., 2020)	➢ requires access server for business ➢ or local deployment for home users ➢ on-demand or standard conf.

TABLE IV: Sample comparison of a popular VPN solutions.

Feature	NordVPN	OpenVPN
single sign-on	✘	✔
Anonymous Browsing	✘	✘
DNS Leak Protection	✔	✔
Multi-Protocol	✔	✔
Kill Switch	✔	✔
Peer-to-Peer	✘	✘
Policy Management	✘	✘
Remote Access	✘	✔
Web Inspection	✘	✘
Client/Server Open Source	✘	✔
Deployment	Cloud	Cloud/Private

users to use they existing enterprise logins to sign-on to the VPN instead of creating separate credentials. Anonymous Browsing is a feature protecting user identity on the web by hiding their real IP (*e.g.* using onion routing) or tracking technologies such as cookies (*e.g.* using Tor browser or other plugins). *DNS Leak Protection* refers to a property where as the a Domain Name System (DNS) related queries and responses are sent through —over a VPN— to DNS servers running in the VPN as opposed to a DNS server provided by the current network. This prevents curious or malicious Internet Service Providers (ISPs) from tracking and recording DNS traffic, hence mitigate against a privacy attack where outsiders are able to know all websites or services the user or devices are accessing. *Web Inspection* is the ability of the VPN server to inspect web traffic (which usually means performing packet inspection at the application layer—OSI Layer 7—), to detect malware, spyware or other harmful content, and respond to any threat.

Many enterprise solutions exist for the purpose of visualization and obtaining analytics of the network traffic, alerts and other events. We have considered the following examples in this report focusing on open source: Cacti[5], Nagios[6], Grafana[7]. These tools enable performing many of the visualization tasks in networks of varied sizes including showing time-series Siimple Network Management Protocol (SNMP), examine different services' logs (*e.g.* for an application or server), and in some cases remotely controlling (*e.g.* restarting) services. This allows home users or system administrators to configured different monitoring probes or alerts when a specific condition has occurred and possibly automate some actions.

For most enterprise settings, administrator usually need to develop and customize the configuration of various dash-boards to monitor different aspects of a large network. On the other hand, for a home environment, users are likely to share specific monitoring and alert configuration for various IoT devices, hence reduce overhead in creating these every time a new device is introduced. User also can access some pre-built (or template) dash-boards, probes or alerts which they can use for their own network.

1) Existing Firewall Solutions for Home Network: As for firewall solutions, there exists many enterprise and consumer grade solutions. Most routers sold to consumers incorporate VPN, firewall as well as other functionality (such as Network Attached Storage (NAS)) —which are typically deployed and managed separately in an enterprise setting— all in a single box (*e.g.* home router). We have surveyed many existing firewall solutions, and we focused this section on firewalls that can be deployed in domestic settings (*i.e.* on a router or a general purpose Personal Computer (PC)). Namely, we focus on *OPNsense* as it is an open-source and the project is under active development.

OPNsense© is forked from pfSense© and has been mostly re-written. pfSense© is a commercial product that has some features released in an open-source version called pfSense© Community Edition (CE). The open-source community edition is released and attracts contribution from a community of interested developers. The enterprise version of pfSense© has license fees. As both of these firewalls have overlapping interested community, the two have been deployed by end-users who wish to customize and add new features. We provide here a high-level comparison of some of the existing firewall solutions as an example. In table V, we show a brief set of features with a focus on the ability to deploy and customize (add/remove a specific functionality) in domestic setting.

TABLE V: Summary of features of *OPNsense*© and pfSense© firewalls.

	OPNsense	**pfSense CE**
License	BSD 2-Clause "Simplified" or "FreeBSD"	Apache License 2.0
Contribution License	BSD 2-Clause "Simplified"	subscribe and electronically sign agreement
IPS	Suricata Inline Intrusion Prevention System (IPS)	Snort IPS package
Extension	✔ plugins	✔ packages
Security Update	weekly	patch release

[5] Cacti website

[6] Magios website

[7] Grafena website

IV. High-level Design of Domestic Network to incorporate IoT

In this section, we discuss a proposed design which deploys WG to facilitate secure tunnelled access into home network. As discussed earlier, initializing and distributing the cryptographic keys is required to enable secure deployment of WG in domestic environment and increase usability.

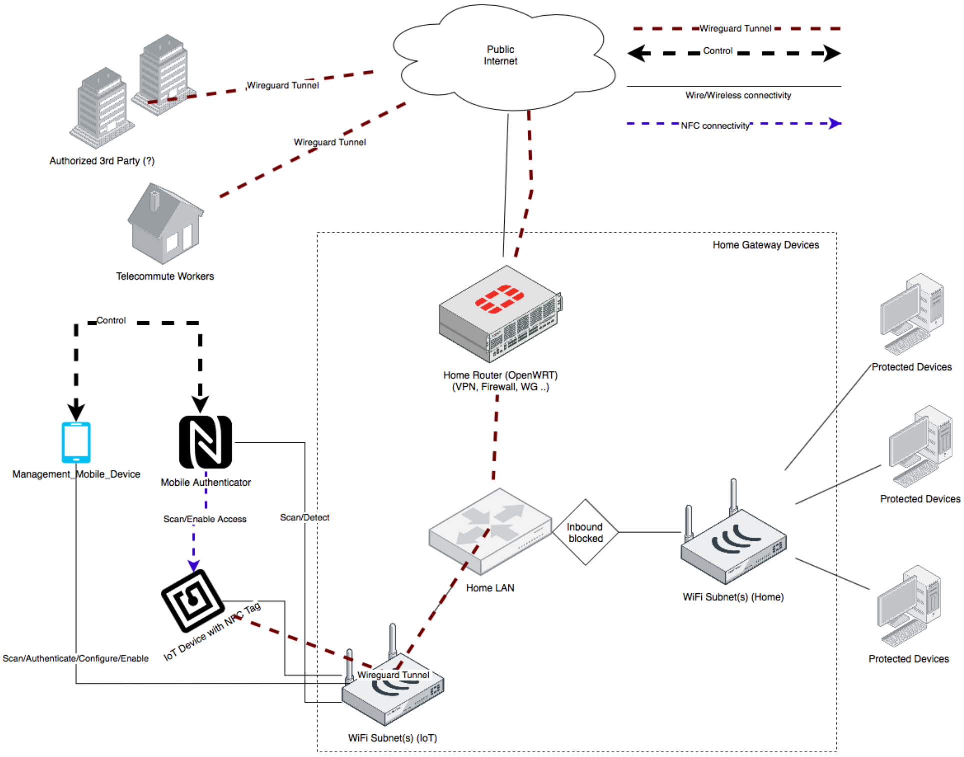


Fig. 2: Overall domestic network diagram using WG to enable secure remote access to IoT devices via tangible interactions.

Figure 2 shows a proposed deployment of WG as a VPN solution. This ensures a secure tunnelling where remote management clients (*e.g.* the users designated device) is able to monitor and receive alerts related to the various problems occurring in the network. We propose that the secure association between the users' management device and a certain subnet is performed using a token-based Near-Field Communication (NFC) tangible interaction, which will then enable this VPN access for this device to one or more of the subnets (more discussion on the subnets later). To avoid enabling wide network access, we propose to have the tunnel associated with a particular subnet where the IoT devices will be deployed.

Figure 3 shows our proposed approach to partitioning the domestic home network into multiple subnets. We show -as an example- the corresponding subnetting using IP version 4 (IPv4), CIDR prefix to drive the subnet Mask in figure 4. Classless Inter-Domain (CIDR) enables efficient routing as a system will search locally for the target host if it is within the same subnet, otherwise -if the target is on another network- traffic will be forward to the gateway to perform routing. This would be useful in the scenarios where a domestic users deploy multiple access-points to handle specific subnets where most of the traffic within a subnet is forwarded directly (instead of relying back to the home gateway).

We use IPv4 in figure 4 because it is widely supported by home routers. It is fairly trivial to perform this same exercise for IP version 6 (IPv6)-only network using a single *Global* ID in the "Unique Unicast" range of *fc00::*/7 (which is not routable on the global Internet as defined in RFC4193) and assign hierarchically the different subnets in a similar way. Note that we are not using site-local addressing range which were deprecated by RFC3879).

The rational behind segregating the network into these different subnets is to enable the use of Virtual Local Area Network (VLAN). Most mid-range domestic routers support using VLAN, each of the VLANs

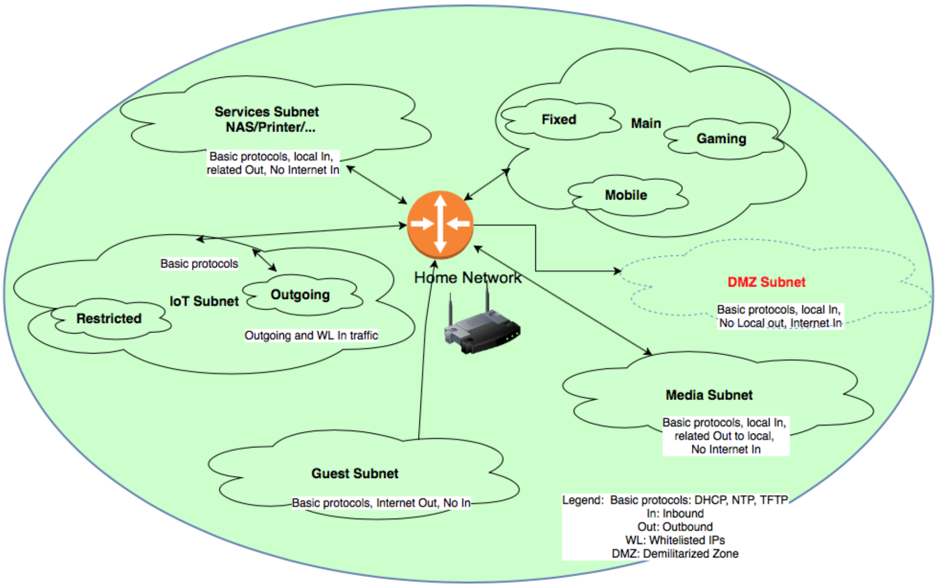
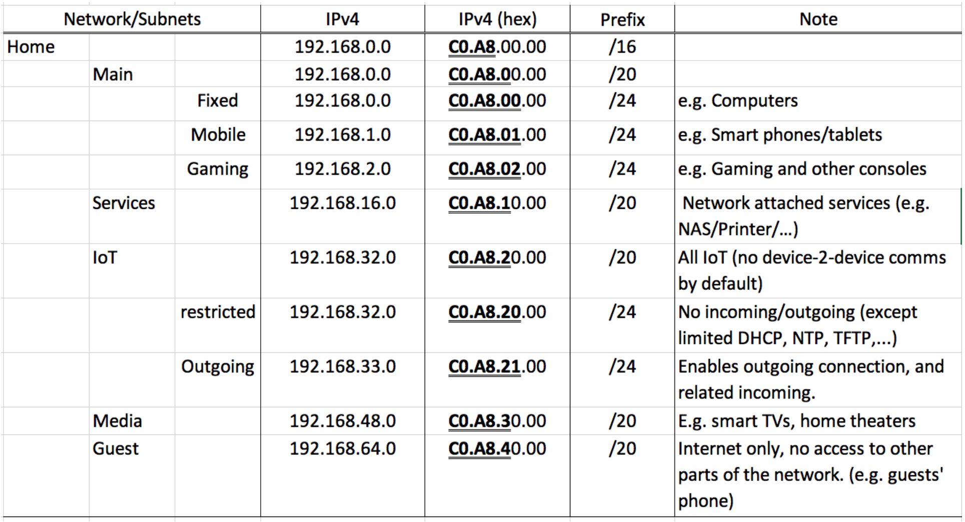

Services Subnet
NAS/Printer/...
Basic protocols, local In, related Out, No Internet In
Fixed
Main
Gaming
Mobile
Basic protocols
IoT Subnet
Outgoing
Restricted
Outgoing and WL In traffic
Home Network
DMZ Subnet
Basic protocols, local In, No Local out, Internet In
Media Subnet
Basic protocols, local In, related Out to local, No Internet In
Guest Subnet
Basic protocols, Internet Out, No In
Legend: Basic protocols: DHCP, NTP, TFTP
In: Inbound
Out: Outbound
WL: Whitelisted IPs
DMZ: Demilitarized Zone

Fig. 3: Domestic network potential subnets and various devices.

would be linked to one of subnets. This ensures that the different subnets are kept separate in management and monitoring, and ensure isolation except where routing and firewall rules permits access.



Network/Subnets			IPv4	IPv4 (hex)	Prefix	Note
Home			192.168.0.0	**C0.A8**.00.00	/16	
	Main		192.168.0.0	**C0.A8.0**0.00	/20	
		Fixed	192.168.0.0	**C0.A8.00**.00	/24	e.g. Computers
		Mobile	192.168.1.0	**C0.A8.01**.00	/24	e.g. Smart phones/tablets
		Gaming	192.168.2.0	**C0.A8.02**.00	/24	e.g. Gaming and other consoles
	Services		192.168.16.0	**C0.A8.1**0.00	/20	Network attached services (e.g. NAS/Printer/...)
	IoT		192.168.32.0	**C0.A8.2**0.00	/20	All IoT (no device-2-device comms by default)
		restricted	192.168.32.0	**C0.A8.20**.00	/24	No incoming/outgoing (except limited DHCP, NTP, TFTP,...)
		Outgoing	192.168.33.0	**C0.A8.21**.00	/24	Enables outgoing connection, and related incoming.
	Media		192.168.48.0	**C0.A8.3**0.00	/20	E.g. smart TVs, home theaters
	Guest		192.168.64.0	**C0.A8.4**0.00	/20	Internet only, no access to other parts of the network. (e.g. guests' phone)

Fig. 4: Home network proposed subnets design to further isolate IoT and other home devices.

Acknowledgement

Funding: This work was supported by the PETRAS 2: EP/S035362/1 National Centre of Excellence for IoT Systems Cybersecurity under the project “Tangible Security” (TanSec).

References

Aazam, M., St-Hilaire, M., Lung, C.-H., Lambadaris, I., 2016. Pre-fog: Iot trace based probabilistic resource estimation at fog. In: 2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE, pp. 12–17.

Agadakos, I., Hallgren, P., Damopoulos, D., Sabelfeld, A., Portokalidis, G., 2016. Location-enhanced authentication using the iot: because you cannot be in two places at once. In: Proceedings of the 32nd Annual Conference on Computer Security Applications. pp. 251–264.

Airehrour, D., Gutierrez, J., Ray, S. K., 2016. Secure routing for internet of things: A survey. Journal of Network and Computer Applications 66, 198 – 213.
URL https://doi.org/10.1016/j.jnca.2016.03.006

Alaba, F. A., Othman, M., Hashem, I. A. T., Alotaibi, F., 2017. Internet of things security: A survey. Journal of Network and Computer Applications 88, 10 – 28.
URL https://doi.org/10.1016/j.jnca.2017.04.002

Alshalan, A., Pisharody, S., Huang, D., 2016. A survey of mobile vpn technologies. IEEE Communications Surveys Tutorials 18 (2), 1177–1196.

Aumi, M. T. I., Kratz, S., 2014. Airauth: evaluating in-air hand gestures for authentication. In: Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services. pp. 309–318.

Aura, T., Sethi, M., 2018. Nimble out-of-band authentication for eap (eap-noob). Tech. rep.
URL -

Azimpourkivi, M., Topkara, U., Carbunar, B., 2017. Camera based two factor authentication through mobile and wearable devices. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 1 (3), 1–37.

Bernstein, D. J., 1997. Tai64, tai64n, and tai64na. [Online] Available: URL https://cr.yp.to/libtai/tai64.html [Accessed on: 11/05/2020].

Bertrone, M., Miano, S., Risso, F., Tumolo, M., 2018. Accelerating linux security with ebpf iptables. In: Proceedings of the ACM SIGCOMM 2018 Conference Posters and Demos. pp. 108–110.

binti Mohamad Noor, M., Hassan, W. H., 2019. Current research on internet of things (IoT) security: A survey. Computer Networks 148, 283 – 294.
URL https://doi.org/10.1016/j.comnet.2018.11.025

Bonneau, J., Herley, C., Van Oorschot, P. C., Stajano, F., 2012. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In: 2012 IEEE Symposium on Security and Privacy. IEEE, pp. 553–567.

Burgbacher, U., Hinrichs, K., 2014. An implicit author verification system for text messages based on gesture typing biometrics. In: Proceedings of the SIGCHI conference on human factors in computing systems. pp. 2951–2954.

Cisco Inc., 2020. Cisco anyconnect secure mobility client data sheet. [Online] Available: URL https://www.cisco.com/c/en/us/products/collateral/security/anyconnect-secure-mobility-client/datasheet-c78-733184.html [Accessed on: 12/09/2020].

Citrix Systems, I., 2020. Citrix sso. [Online] Available: URL https://play.google.com/store/apps/details?id=com.citrix.CitrixVPN [Accessed on: 12/09/2020].

Costin, A., Zaddach, J., Francillon, A., Balzarotti, D., 2014. A large-scale analysis of the security of embedded firmwares. In: 23rd {USENIX} Security Symposium ({USENIX} Security 14). pp. 95–110.

Crabtree, A., Lodge, T., Colley, J., Greenhalgh, C., Glover, K., Haddadi, H., Amar, Y., Mortier, R., Li, Q., Moore, J., Wang, L., Yadav, P., Zhao, J., Brown, A., Urquhart, L., McAuley, D., 2018. Building accountability into the internet of things: the iot databox model. Journal of Reliable Intelligent Environments 4 (1), 39–55.
URL 10.1007/s40860-018-0054-5

Czeskis, A., Dietz, M., Kohno, T., Wallach, D., Balfanz, D., 2012. Strengthening user authentication through opportunistic cryptographic identity assertions. In: Proceedings of the 2012 ACM conference on Computer and communications security. pp. 404–414.

De Cristofaro, E., Du, H., Freudiger, J., Norcie, G., 2013. A comparative usability study of two-factor authentication. arXiv preprint arXiv:1309.5344.

De Luca, A., Hang, A., Brudy, F., Lindner, C., Hussmann, H., 2012. Touch me once and i know it's you! implicit authentication based on touch screen patterns. In: proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 987–996.

De Luca, A., Harbach, M., von Zezschwitz, E., Maurer, M.-E., Slawik, B. E., Hussmann, H., Smith, M., 2014. Now you see me, now you don't: protecting smartphone authentication from shoulder surfers. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2937–2946.

De Luca, A., Von Zezschwitz, E., Nguyen, N. D. H., Maurer, M.-E., Rubegni, E., Scipioni, M. P., Langheinrich, M., 2013. Back-of-device authentication on smartphones. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2389–2398.

Donenfeld, J. A., 2017. Wireguard: Next generation kernel network tunnel. In: The Network and Distributed System Security Symposium (NDSS). pp. 06–30.

Fan, X., Susan, F., Long, W., Li, S., 2017. Security analysis of zigbee. Massachusetts Institute of Technology.

Gartner, 2019. Gartner says 5.8 billion enterprise and automotive iot endpoints will be in use in 2020. [Online] Available: URL https://www.gartner.com/en/newsroom/press-releases/2019-08-29-gartner-says-5-8-billion-enterprise-and-automotive-io [Accessed on: 14/08/2020].

Han, D., Chen, Y., Li, T., Zhang, R., Zhang, Y., Hedgpeth, T., 2018. Proximity-proof: Secure and usable mobile two-factor authentication. In: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking. pp. 401–415.

Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., Sikdar, B., 2019. A survey on iot security: Application areas, security threats, and solution architectures. IEEE Access 7, 82721–82743.

Jinlong, E., Ma, J., 2013. A hybrid transmission system based on nfc-enabled mobile phones. Journal of Software 8 (11), 2738–2748.

Kandaswamy, R., Furlonger, D., 2018. Blockchain-based transformation: a gartner trend insight report. [Online] Available: URL https://www.gartner.com/en/doc/3869696-blockchain-based-transformation-a-gartner-trend-insight-report [Accessed on: 10/6/2020].

Karapanos, N., Marforio, C., Soriente, C., Capkun, S., 2015. Sound-proof: usable two-factor authentication based on ambient sound. In: 24th {USENIX} Security Symposium ({USENIX} Security 15). pp. 483–498.

Karole, A., Saxena, N., Christin, N., 2010. A comparative usability evaluation of traditional password managers. In: International Conference on Information Security and Cryptology. Springer, pp. 233–251.

Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., Kivinen, T., 2010. Internet key exchange protocol version 2 (ikev2). Tech. rep., RFC 5996, September.

Khan, M. A., Salah, K., 2018. Iot security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems 82, 395–411.

Krol, K., Philippou, E., De Cristofaro, E., Sasse, M. A., 2015. "they brought in the horrible key ring thing!" analysing the usability of two-factor authentication in uk online banking. arXiv preprint arXiv:1501.04434.

Leung, H.-M. C., Fu, C.-W., Heng, P.-A., 2018. Twistin: Tangible authentication of smart devices via motion co-analysis with a smartwatch. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2 (2), 1–24.

Li, Y., 2010. Protractor: a fast and accurate gesture recognizer. In: Proceedings of the SIGCHI conference on Human Factors in computing systems. pp. 2169–2172.

Lipp, B., Blanchet, B., Bhargavan, K., 2019. A mechanised cryptographic proof of the wireguard virtual private network protocol. In: 2019 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, pp. 231–246.

Marquardt, N., Greenberg, S., 2012. Informing the design of proxemic interactions. IEEE Pervasive Computing 11 (2), 14–23.

Mortier, R., Rodden, T., Lodge, T., McAuley, D., Rotsos, C., Moore, A. W., Koliousis, A., Sventek, J., 2012. Control and understanding: Owning your home network. In: 2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012). pp. 1–10.

Naik, N., 2017. Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. In: 2017 IEEE international systems engineering symposium (ISSE). IEEE, pp. 1–7.

Naik, N., Jenkins, P., 2016. Web protocols and challenges of web latency in the web of things. In: 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN). IEEE, pp. 845–850.

Neshenko, N., Bou-Harb, E., Crichigno, J., Kaddoum, G., Ghani, N., 2019. Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations. IEEE Communications Surveys Tutorials 21 (3), 2702–2733.

NetMotion, 2020. Core functionality of NetMotion VPN solution. [Online] Available: URL https://www.netmotionsoftware.com/solutions/enterprise-vpn-software-solutions [Accessed on: 15/06/2020].

OHara, K., 2016. The seven veils of privacy. IEEE Internet Computing 20 (2), 86–91.

OpenVPN Inc., 2020. Private tunnel. [Online] Available: URL https://openvpn.net/private-tunnel/ [Accessed on: 21/06/2020].

Payne, J., Jenkinson, G., Stajano, F., Sasse, M. A., Spencer, M., 2016. Responsibility and tangible security: Towards a theory of user acceptance of security tokens. arXiv preprint arXiv:1605.03478.

Perrin, T., 2018. The noise protocol framework revision 34. [Online] Available: URL https://noiseprotocol.org/noise.html [Accessed on: 07/06/2020].

Pudelko, M., Emmerich, P., Gallenmüller, S., Carle, G., 2020. Performance analysis of vpn gateways. In: 2020 IFIP Networking Conference (Networking). pp. 325–333.

Pulse Secure, L., 2020. Pulse connect secure. [Online] Available: URL https://www.pulsesecure.net/products/remote-access-overview/ [Accessed on: 15/09/2020].

Radio IP Software, 2020. Radio IP synopsis VPN features. [Online] Available: URL https://radio-ip.com/en/solutions/synopsis/ [Accessed on: 12/09/2020].

Ren, J., Dubois, D. J., Choffnes, D. R., Mandalari, A. M., Kolcun, R., Haddadi, H., 2019. Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach. In: Proceedings of the Internet Measurement Conference, IMC 2019, Amsterdam, The Netherlands, October 21-23, 2019. ACM, pp. 267–279.
URL 10.1145/3355369.3355577

Sahami Shirazi, A., Moghadam, P., Ketabdar, H., Schmidt, A., 2012. Assessing the vulnerability of magnetic gestural authentication to video-based shoulder surfing attacks. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. pp. 2045–2048.

Sethi, M., Oat, E., Di Francesco, M., Aura, T., 2014. Secure bootstrapping of cloud-managed ubiquitous displays. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing. pp. 739–750.

Shrestha, P., Saxena, N., 2018. Listening watch: Wearable two-factor authentication using speech signals resilient to near-far attacks. In: Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks. pp. 99–110.

SIGFOX, 2020. SIGFOX global 0g network. [Online] Available: URL https://www.sigfox.com/en/what-sigfox/technology [Accessed on: 16/05/2020].

Sivaraman, V., Gharakheili, H. H., Vishwanath, A., Boreli, R., Mehani, O., 2015. Network-level security and privacy control for smart-home iot devices. In: 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE, pp. 163–167.

SonicWall, Inc., 2020. Sonicwall mobile connect datasheet. [Online] Available: URL https://d3ik27cqx8s5ub.cloudfront.net/media/uploads/2020/03/Datasheet-SonicWallMobileConnect-VG-US-1811.pdf [Accessed on: 15/09/2020].

Stajano, F., 2011. Pico: No more passwords! In: International Workshop on Security Protocols. Springer, pp. 49–81.

Standardization, I., 1996. Iso/iec 7498-1: 1994 information technology–open systems interconnection–basic reference model: The basic model. International Standard ISOIEC 74981, 59.

Tian, J., Qu, C., Xu, W., Wang, S., 2013. Kinwrite: Handwriting-based authentication using kinect. In: NDSS. Vol. 93. p. 94.

Torvalds, L., 2020. Linux kernel 5.6 announcement. [Online] Available: URL https://lore.kernel.org/lkml/CAHk-=wi9ZT7Stg-uSpX0UWQzam6OP9Jzz6Xu1CkYu1cicpD5OA@mail.gmail.com/ [Accessed on: 3/05/2020].

Vejlgaard, B., Lauridsen, M., Nguyen, H., Kovács, I. Z., Mogensen, P., Sorensen, M., 2017. Coverage and capacity analysis of sigfox, LORA, GPRS, and NB-IoT. In: 2017 IEEE 85th vehicular technology conference (VTC Spring). IEEE, pp. 1–5.

VMWare Inc., 2020. Vmware workspace ONE UEM documentation. [Online] Available: URL https://docs.vmware.com/en/VMware-Workspace-ONE-UEM/index.html [Accessed on: 16/09/2020].

Wu, P., 2019. Analysis of the wireguard protocol. Master's thesis. Eindhoven University of Technology, Eindhoven, Netherlands.

Yang, Y., Clark, G. D., Lindqvist, J., Oulasvirta, A., 2016. Free-form gesture authentication in the wild. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. pp. 3722–3735.

Zheng, N., Bai, K., Huang, H., Wang, H., 2014. You are how you touch: User verification on smartphones via tapping behaviors. In: 2014 IEEE 22nd International Conference on Network Protocols. IEEE, pp. 221–232.

Zhou, B., Lohokare, J., Gao, R., Ye, F., 2018. Echoprint: Two-factor authentication using acoustics and vision on smartphones. In: Proceedings of the 24th Annual International Conference on Mobile Computing and Networking. pp. 321–336.