ZITA: Zero-Interaction Two-Factor Authentication using Contact Traces and In-band Proximity Verification

Nirnimesh Ghose*, Kaustubh Gupta*, Loukas Lazos†, Ming Li†, Ziqi Xu†, and Jincheng Li†
*School of Computing, University of Nebraska–Lincoln, USA
Email: {nghose, kgupta97}@unl.edu
†Department of Electrical and Computer Engineering, University of Arizona, USA
Email: {llazos, lim, zxu1969, jli2972}@arizona.edu

Abstract—Two-factor authentication (TFA) provides an additional layer of protection to commonly-occurring password breaches. However, existing TFA methods, often involve special hardware interfaces, or require human effort which is prone to errors and acts as an adoption detractor for older adults and novice technology users. To address these limitations, we propose a zero-interaction, two-factor authentication (ZITA) protocol. In ZITA, the first factor is implemented using the conventional username and password methods. The second factor is completed without any human effort provided that the user is not accessing the service from an unregistered public device and a designated secondary device is physically co-present. To automate the second factor, ZITA exploits the long-term contact between the login device and the secondary device such as a smartphone. Moreover, to thwart man-in-the-middle and co-located attacks, ZITA incorporates a proximity verification test that relies on the randomness of ambient RF signals. Compared with other zero-effort TFA protocols, ZITA remains secure against advanced threats and does not require out-of-band sensors such as microphones, speakers, or photoplethysmography (PPG) sensors.

Index Terms—Two-factor authentication, Physical-layer Security, Wireless Signal Manipulation Attacks, Man-in-the-Middle Attacks, In-band, COTS wireless devices.

1 Introduction

Identity secrets such as user passwords, PINs, etc. often leak to unauthorized parties via a variety of attacks. For example, the 2014 Yahoo database breach compromised over 500 million accounts [1]. A fraction of those passwords were hashed with the breakable MD5 hash function, leading to almost certain password exposure. Other popular password theft methods include phishing attacks [2], malware installs, key reinstallation attacks (KRACKs) [3], and short-term physical access to personal devices. The latter method has recently gained attention with reports of intimate partner violence (IPV) attacks [4], where a seemingly friendly party gains temporary physical access to a victim's device.

Exposed passwords can be financially and emotionally detrimental to victims. Often, the same passwords are reused across services, platforms, and networks [5]. Any party with access to passwords can gain access to a plethora of web services and even remotely control sensitive devices such as in-home cameras and web-enabled doors. A widely adopted practice for boosting security against password compromise is to employ multi-factor authentication that incorporates additional proof of identity. By far, the most popular methods incorporate two-factor authentication (TFA), such as those based on SMS [6], hardware code generators [7], or time-based one-time passwords (TOTP) [8]. However, most TFA protocols remain vulnerable to attacks such as SIM cloning or shoulder surfing [9], [10]. More secure second authentication factors include token-based authentication such as Duo [11], or USB hardware tokens (aka security keys) such as a YubiKey [7]. These methods either require the validation of an authentication request on an out-of-band channel (e.g., visually) or require the possession of a special hardware key. In TFA, the prover must demonstrate knowledge of the long-term password and a short-term secret (e.g., SMS code). Moreover, from a usability perspective, most TFA methods require user effort which is often a point of frustration for users and a detractor to adoption [12], [13].

To address these limitations, we propose a novel two-factor authentication protocol called **Z**ero-**I**nteraction **T**wo-Factor **A**uthentication (ZITA). A typical use case for ZITA is the access of a web service via a login device (e.g., a laptop) and the use of a secondary device (e.g., the user's smartphone) to complete the second factor. Compared to existing TFA methods, ZITA does not require user interaction and incorporates a form of continuous authentication between the login and the secondary devices. An overview of ZITA is shown in Fig. 1. Initially, the user enrolls a designated login device $\mathcal{D}$ and a secondary device $\mathcal{S}$ with the service,

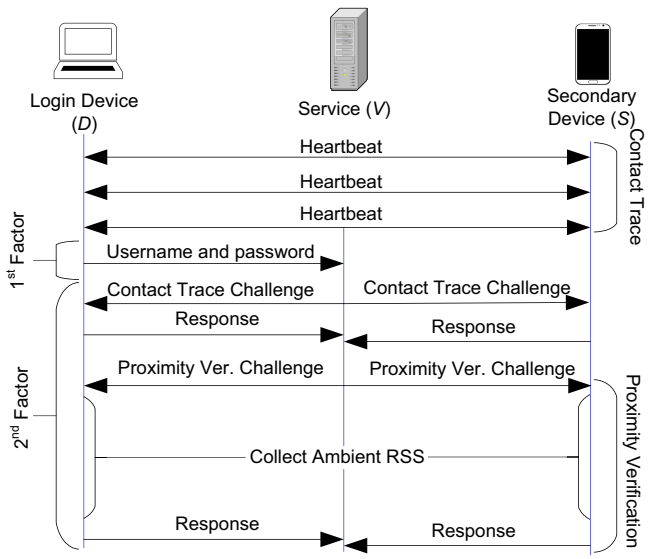


Fig. 1. Overview of the ZITA two-factor authentication.

referred to as the verifier $\mathcal{V}$. The devices $\mathcal{D}$ and $\mathcal{S}$ initiate the periodic exchange of random nonces we call, "heartbeats." The heartbeats generate a common contact trace that proves the prolonged co-existence between the designated login and secondary devices. Heartbeat exchanges build trust between the devices over time instead of at one instance, as is the case with short-term secrets. The heartbeat exchange is paused if the devices separate and resumes upon further contact.

When a user attempts a login, the first-factor authentication is implemented using conventional methods such as user passwords. *The second factor is automated and requires zero user effort.* To complete the second factor, the verifier challenges $\mathcal{D}$ and $\mathcal{S}$ to demonstrate knowledge of the contact trace. The contact trace proves the use of the enrolled login device $\mathcal{D}$, thus allowing the automation of the second authentication factor. To protect against advanced threats such as man-in-the-middle (MiTM) attacks, ZITA further implements physical proximity verification between $\mathcal{D}$ and $\mathcal{S}$ using the ambient radio-frequency (RF) environment. The two devices sample ambient RF signals and report them to the verifier. If $\mathcal{D}$ and $\mathcal{S}$ are in proximity, their samples are highly correlated. Physical proximity is used to verify that both $\mathcal{D}$ and $\mathcal{S}$ are under the user's control.

Upon the successful completion of the second factor, a notification is pushed to the user's devices. In case of failure (e.g., due to the use of an unregistered device or an attack), ZITA defaults to a push-button verification similar to Duo, requiring the user's action to complete authentication.

Difference from RSS-based proximity verification and key-extraction: We highlight that the use of ambient RSS to verify proximity in our setting is under an entirely different threat model compared to prior works [14], [15]. Classical proximity verification methods aim at preventing a prover from claiming to be at close distance to a verifier, while far away. In our model, the adversary aims at making the login device $\mathcal{D}$ and the secondary device $\mathcal{S}$ (both of which are benign) appear to be in close contact, while far away. The two devices share trust and are interested in inferring proximity, while the adversary manipulates their distance. This difference in the system/threat models poses new challenges and enables new solutions.

Moreover, the problem at hand is fundamentally different from pairwise key extraction based on RSS [16], [17]. Such methods aim at extracting a common key from the randomness of small-scale fading, which is introduced by multi-path distortions. The secrecy of the key is based on the unique location of the interacting devices rather than their proximity. Theoretically, such devices can be at any distance as long as their transmissions are measurable. In contrast, ZITA targets to prevent a MiTM adversary from emulating the RF environment near $\mathcal{D}$ and $\mathcal{S}$ when they are not in proximity. Moreover, $\mathcal{D}$ and $\mathcal{S}$ have already established trust, so no key needs to be extracted.

Another unique advantage of the RSS-based proximity verification method for ZITA is that it relies on large-scale fading to allow for practical distances (e.g., within a room) between the login device and the secondary device. This is in contrast with some prior methods that use small-scale fading to verify proximity within a few wavelengths (a few cm).

Summary of Contributions:

- We develop ZITA, a TFA protocol that does not require any user interaction to complete the second factor. Compared to prior zero-effort TFA protocols [22], [30], [31], [35], [36], ZITA is resistant to MiTM attacks and co-located attacks thanks to the implementation of the contact trace and proximity verification primitives. The former continuously builds and updates long-term trust between the login and secondary device. The latter thwarts MiTM login attempts by verifying that both $\mathcal{D}$ and $\mathcal{S}$ are under the user's control. Proximity verification operates in-band using universally-available RF interfaces without requiring special sensors such as microphones/speakers [22], [31], light detectors [37], or photoplethysmography (PPG) sensors [30]. This makes ZITA easy to deploy.
- We analyze the security of ZITA under a strong adversary model that assumes the exposure of the first factor and advanced threats against the second factor. The array of threats considered includes the deployment of MiTM attacks between $\mathcal{D}$ and $\mathcal{S}$, physical access to the login device $\mathcal{D}$ by an intimate partner, a co-located attack where the adversary is in proximity to the login and secondary devices, and loss of the secondary device.
- We perform testbed experiments to demonstrate the security of ZITA. We show that the proximity verification mechanism is resistant to both passive and active overshadowing attacks. We further implement a proof-of-concept ZITA application on the Android 10 platform.

Organization: In Section 2, we discuss related work. In Section 3, we present the system and threat models. The ZITA protocol is described in Section 4. We analyze the security of ZITA in Section 5. The experimental evaluation and proof of concept implementation of ZITA are presented in Section 6. We conclude in Section 7.

TABLE 1
Security of two-factor authentication schemes.

Method	Zero Interaction	Required Modality	MiTM Attack Security	Device Compromise Security
Hardware code generators [18]	×	User	✓	×
SMS-based [6]	×	User	✓	×
2FA-PP [19]	×	User	✓	×
Duo [11]	×	User/RF	✓	×
YubiKey [7]	×	User	✓	×
Jarecki *et al.* [20]	×	User/Visual	✓	×
Sound-Login [21], Sound-Proof [22], SoundAuth [23], Listening-watch [24]	✓	Audio	×	×
Wi-Auth [25]	✓	User	×	×
PINTA [26], PACA [27], WACA [28]	✓	Accelerometer/Gyroscope	×	✓
Proximity-Echo [29]	✓	Audio	×	×
PPGPass [30]	✓	PPG	×	✓
Proximity-Proof [31]	✓	Audio	✓	×
FastZIP [32]	✓	Accelerometer/Gyroscope	✓	×
DASK [33]	✓	GPS	×	×
ivPair [34]	✓	Accelerometer/Gyroscope	×	×
ZITA (this work)	✓	RF	✓	✓

2 Related Work

Two Factor Authentication. Existing TFA methods can be broadly classified into those requiring user interaction and those with zero effort.

TFA with user interaction: Most commercially available TFA methods can be classified into single code use and token-based ones. The most popular single-use code methods are the time-based one-time password (TOTP), SMS-based TFA [6], and hardware code generators (such as the RSA SecurID) [18]. In SMS-based methods, a time-sensitive one-time password is delivered to the registered phone number as a text message. This method has been shown insecure due to sim cloning and shoulder surfing [9]. For hardware code generators, a hardware device stores a pre-generated one-way hash chain, and each chain value is used in reverse order. This method has been shown vulnerable to shoulder surfing and seed compromise, as the security of the one-way hash chain is based on the secrecy of the seed [10].

In more recent TFA methods such as Duo [11], Google prompt [38], or the YubiKey USB hardware token [7], typing a one-time password is replaced by a token-based authentication. In Duo and Google Prompt, the user registers with an application on a secondary device. Each login attempt has to be approved by the user by pushing a button on the application. Authentication is completed by executing a challenge-response protocol between the service and the application where the application proves the knowledge of a fresh token obtained from a one-way hash chain. In contrast to Duo, *ZITA provides a zero-effort user experience when logging in from a registered device, as no button needs to be pressed on the secondary device*. If the secondary device is co-located with the registered device, the second factor is automatically executed and the user is just notified of the TFA execution.

The YubiKey hardware token is based on the FIDO U2F authentication protocol [39]. To complete the second factor, the user is required to carry a hardware USB token which is automatically executing a challenge-response protocol using one-time passwords generated from one-way hash chains. The latter has been shown vulnerable to offline dictionary attacks on the hash chain seed [20]. An attacker can generate an offline dictionary for all possible nonces, similar to the attack on the hash of salted passwords. To improve security, Jarecki *et al.* proposed an end-to-end secure TFA scheme that combats the offline dictionary attack by implementing an out-of-band verification mechanism based on visually-verified short-authentication-strings [20]. The 2nd Factor Phishing Prevention (2FA-PP) performs an additional verification to ensure that both the login and the secondary devices are on the same domain [19].

Zero-effort TFA : Several prior TFA methods were designed to eliminate the user interaction for completing the second factor. They primarily address well-known usability concerns among older adults and less technology-savvy users [12], [13]. Sound-Login [21] Sound-Proof [22], SoundAuth [23] and Listening-watch [24] implement the second factor over an audio channel. In Sound-Login, the secondary device plays audio of a one-time code to the login device which is then used as a second factor. However, Sound-login is susceptible to MiTM attacks where an adversary can record sound close to the secondary device and replay it close to the login device, while the two devices are far apart. Moreover, the protocol is susceptible to a co-located attack where a co-located adversary can overhear the one-time password and pass the second factor. Sound-proof implements the second factor using ambient noise recorded by microphones to verify device proximity [22]. This method differs from ZITA in two ways. First, Soundproof has been shown to be vulnerable to attacks that can predict the ambient sound. Second, a MiTM adversary can launch active attacks and influence the ambient sound environment [40]. The SoundAuth protocol [23] is an improvement of the Sound-Proof protocol which verifies proximity by playing random sounds in near-ultrasound frequencies. Because these sounds are randomized, they cannot be predicted. However, near-ultrasound frequencies are still audible by humans and pets and can be quite uncomfortable. Moreover, SoundAuth is vulnerable to a MiTM attack where an adversary relays sounds from the primary login device to the secondary device. Listening-watch [24] is another sound-based method that aims at

improving the security of Sound-Proof. Based on the same system model and idea, it replaces the use of ambient background noise with a short random voice-based sound played by an active browser. Similar to SoundAuth, the listening watch is vulnerable to a MiTM attack where an adversary with temporary access to the primary device can record, relay, and replay sound in the vicinity of the secondary device and successfully pass the second factor. The threat model only considers a remote adversary. ZITA, on the other hand, can withstand a MiTM adversary who performs a record and relay attack and does not require very close proximity to the primary device. Moreover, *security is maintained even if the adversary gets close to the primary/secondary devices.* This is because the recorded RSS is not sufficient to pass verification and influencing the RF environment in a predictable manner at multiple locations is a hard problem in the presence of multipath and ambient RF sources.

Other TFA methods rely on unique features of user behavior captured through wearable devices to provide continuous authentication [26]–[28]. However, these methods require a user enrollment phase to capture sufficient training data for each user. Moreover, the randomness of user behavior cannot be renewed. Han *et al.* proposed a Proximity-proof to combat a MiTM attacker using an ultrasonic channel [31]. The key difference with sound-proof is that acoustic fingerprints are difficult to replicate by an active attacker. Ren *et al.* proposed proximity detection as TFA by capturing acoustic beeps alternatively emitted by both legitimate devices and performing similarity analysis on the energy loss during a period [29]. In PPGPass, Cao *et al.* proposed the use of photoplethysmography sensor on a wearable for detecting authenticated users as the TFA [30]. In contrast to the above works, ZITA does not require any out-of-band channel, and is secure against MiTM attacks and co-located attacks, while removing the vulnerabilities and usability limitations introduced by human verification. A comparison between ZITA and most related two-factor authentication schemes is shown in Table 1.

Proximity Verification. ZITA uses a form of proximity verification to implement the second authentication factor. Several prior works have investigated methods for verifying proximity or co-presence as the first factor. These works can be broadly classified to out-of-band [41]–[43] and in-band verification [14], [15], [25], [44]–[47].

Out-of-band Proximity Verification : The main idea of out-of-band proximity verification is to simultaneously measure some physical property that exhibits high spatial correlation. Choi *et al.* proposed a passive keyless entry method where the similarity of ambient acoustic signals between a user's device and his car are used to validate proximity [41]. Miettinen *et al.* employed fuzzy vaults to reconcile ambient measurements of luminosity and audio [42]. They demonstrated that longitudinal luminosity can increase the entropy of the context and lead to faster key agreement. Fomichev *et al.* proposed FastZIP, a zero-interaction device pairing method, that fuses ambient context to verify the proximity of two devices and establish a common secret [32]. The use of multiple sensors increases the entropy and reduces the pairing time. Yang *et al.* proposed the DASK protocol for pairing devices co-located within the same vehicle [33]. The main idea is to exploit the unknown vehicle trajectory to extract common randomness (via GPS signals) and establish shared keys. Li *et al.* proposed ivPair, a device pairing protocol that derives a continuously renewed fingerprint from the vibrational response of the vehicle [34].

FastZIP, DASK, and ivPair are designed as key pairing protocols that extract common randomness from ambient context. They are zero effort since secrets are established without pre-loading or user input. However, they do not provide explicit device authentication (this was not their intent). In fact, authentication can only be achieved through presence, and a trusted boundary is required to surround the legitimate devices. Furthermore, the identities of two devices co-located within the same context cannot be differentiated as the devices would sample the same common context and extract correlated keys. ZITA, on the other hand, targets two-factor authentication and assumes the pre-existence of secrets (passwords), which could be compromised. Moreover, the threat model of ZITA makes no assumption about a trusted boundary and successfully defends against co-located attacks. We refer the reader to a comprehensive survey on context-based co-presence detection techniques for a detailed comparison among various methods [43].

In-band Proximity Verification: In-band proximity verification utilizes the small-scale fading of RF signals for verifying the proximity of devices within a short distance [14], [15], [25], [44]–[47]. This is because small-scale fading is mainly caused by multipath distortion which quickly decorrelates with distance. Typical distances are a few wavelengths (e.g., the wavelength is 12.5cm at 2.4GHz). In Wanda, Pierson *et al.* exploited the exponential drop in the ratio of received signal strength when a "wand" with two antennas separated by 7cm was moved away from a legitimate device [14]. Hessar *et al.* have proposed the use of small-scale fading for verification of devices on a human body by verification of signals produced by fingerprint sensors and touchpads [44]. Authors have proposed to utilize a novel technique to synthesize RSSI data on different channels to increase the entropy of proximity-based key generation [45]. In SNAP, authors have proposed to utilize a single antenna's near field effect to detect devices in proximity [46]. Luo *et al.* have proposed the use of backscattering ambient signals to exploit the small-scale fading to verify proximity verification [47]. In Wi-Auth, following the verification of the user credentials, the user is prompted to place the secondary device *in close proximity (i.e. within a few cm)* of the primary device. The two devices measure the CSI from a connected AP and record the CSI data as evidence of their proximity [25]. If the data is highly correlated, the TFA is successfully completed. Existing in-band methods achieve verification within a limited range (a few cm) due to their reliance on small-scale fading. This limits the usability of the protocol. For instance, if a laptop is used, the user will need to know the precise location of the laptop antenna(s) for the correct secondary device

placement. This is impractical for most users who do not process intricate domain expertise.

In ZITA, we improve usability by relaxing the proximity verification distance while preventing co-location attacks. It is quite common for users to have their secondary devices (smartphone, smartwatch) nearby, but not within a few cm. For instance, the user may wear his smartwatch, or have his smartphone resting on the desk, in his pocket, etc. For all those common scenarios, our method requires no interaction.

3 Preliminaries

3.1 System Model

We consider the standard TFA model established in prior works such as Duo [11], Google 2-step verification [38], and Proximity-Proof [31]. The model is depicted in Fig. 1. A user wants to securely access a web service via a login device $\mathcal{D}$ such as a laptop, a desktop, a tablet, or use a public computer. The communication between the server and the login device is secured using traditional server-client security protocols such as TLS. When the user attempts to log into the web service, she enters her login and password information that implements the first-factor authentication.

To implement the second factor, the user enrolls a secondary device with the web service. We refer to the web server as the *verifier* $\mathcal{V}$, the designated login device as $\mathcal{D}$, the secondary device as $\mathcal{S}$, and a public login device as $\mathcal{T}$. During enrollment, the web server establishes a secret key K_{VD} with the designated login device $\mathcal{D}$ and a secret key K_{VS} with the secondary device $\mathcal{S}$. Moreover, $\mathcal{D}$ and $\mathcal{S}$ establish their own secret key K_{DS} and can directly communicate via an RF interface such as Bluetooth, a Wi-Fi ad-hoc connection, or a local area network when in range. We emphasize that *setting up the connection between $\mathcal{D}$ and $\mathcal{S}$ requires a one-time effort*.

Message exchanges between legitimate parties sharing cryptographic keys are secured via an authenticated encryption function AE$(\cdot)$. The function guarantees the confidentiality and integrity of the exchanged messages and also verifies the authenticity of the source. Any such AE$(\cdot)$ can be utilized with the proposed protocol. We refer the reader to [48] for more details on authenticated encryption. We leave the exact specification of AE$(\cdot)$ open to allow for both symmetric and/or asymmetric methods.

3.2 Threat Model

The goal of an adversary $\mathcal{M}$ is to log into the web service, represented by the verifier $\mathcal{V}$. Given that our goal is to implement a zero-effort TFA protocol, we adopt a threat model similar to prior works [29]–[31]. Specifically, we assume that the user's login username and password have been exposed to $\mathcal{M}$. This can be done if the same credentials are re-used in another web service that has been compromised [1], or the user becomes a victim of a phishing attack [2]. The login browser or app used to access the service is secure. Moreover, the registered devices $\mathcal{D}$ and $\mathcal{S}$ are under the user's physical control, although $\mathcal{D}$ may come under $\mathcal{M}$'s control temporarily (e.g. when the user is away from $\mathcal{D}$). Any pairwise secrets stored in the devices' memories are inaccessible to the adversary. The adversary is in control of the wireless channel between the legitimate parties and can intercept, modify, and inject any messages of his choosing. For zero-effort TFA protocols, four additional attack scenarios are particularly relevant [29]–[31].

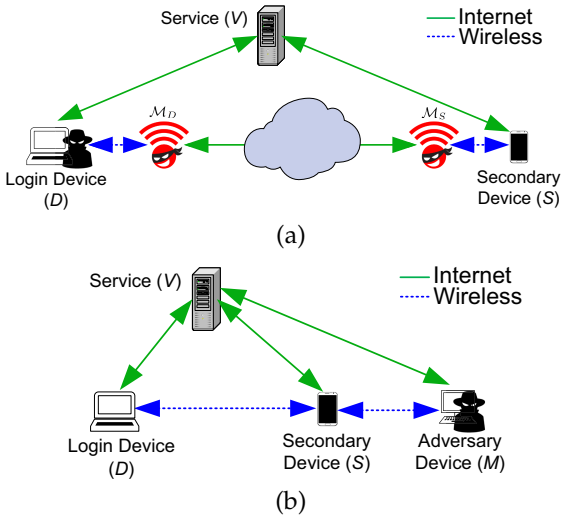


Fig. 2. (a) Man-in-the-middle attack when $\mathcal{D}$ and $\mathcal{S}$ are not in proximity, and (b) co-located attack when $\mathcal{M}$ is in proximity with $\mathcal{D}$ and $\mathcal{S}$.

Man-in-The-Middle Attack (MiTM): In a MiTM attack, the adversary aims at deceiving the secondary device into assuming it is close to the login device. This scenario is depicted in Fig. 2(a). The user, who is in possession of the secondary device is away from the login device $\mathcal{D}$. The adversary gains physical access to $\mathcal{D}$ and uses the compromised login information to access the web service. The adversary deploys a high-speed link between $\mathcal{D}$ and $\mathcal{S}$ (e.g., via the Internet infrastructure) that is invisible to the legitimate devices. Using this link, he can relay messages between $\mathcal{D}$ and $\mathcal{S}$ as if the two devices were in proximity. Given the range of contemporary wireless technologies, the covertness of the high-speed link can be easily achieved by installing two eavesdropping devices within proximity of the targets, but out of the user's sight. Finally, the adversary can inject his own transmissions to influence the wireless environment around $\mathcal{D}$ and $\mathcal{S}$.

Co-located Attack This attack is depicted in Fig. 2(b). The adversary is in physical proximity to $\mathcal{D}$ and $\mathcal{S}$, while the devices are under the user's control. The adversary attempts to log in from its own device $\mathcal{D}$, which triggers the second-factor authentication.

Loss of Secondary Device: In addition to the above threats, we consider that the secondary device could be lost or stolen. Although $\mathcal{S}$ is under the physical control of the adversary, the device remains locked and cannot be accessed without the device's PIN. However, the adversary can move the device to a location of his own choosing. We do not assume simultaneous physical access to both the login device $\mathcal{D}$ and the secondary device $\mathcal{S}$. To the best of our knowledge, such

a strong attacker model cannot be dealt with by any of the existing zero-effort TFA methods.

Device compromise: The strongest threat that we consider is the case where $\mathcal{M}$ manages to temporarily compromise the login device $\mathcal{D}$. This can be achieved if the adversary installs malware on one of the devices. The malware provides access to any session keys that may be stored in the compromised device. The compromise is temporary and eventually, the malware is removed from the infected device (e.g., via malware scanning performed once a day).

We do not consider denial-of-service (DoS) attacks, where $\mathcal{M}$ forces $\mathcal{D}$ and $\mathcal{S}$ to engage in repeated failed login attempts. Such attacks do not serve the purpose of gaining access to the web service and are easily mitigated by limiting the login attempt.

4 ZITA: Zero Interaction Two-Factor Authentication

4.1 Protocol Overview

ZITA is a zero-effort authentication protocol that enables the automated execution of the second factor without any user interaction when the following conditions are met:

1) The user attempts to log in from a designated device $\mathcal{D}$.
2) The secondary device $\mathcal{S}$ is in proximity to $\mathcal{D}$.
3) The two devices $\mathcal{S}$ and $\mathcal{D}$ can provide proof of long-term contact via a contact trace.

The first two conditions occur when a user frequently accesses a web service from her laptop or desktop computer and uses the enrolled smartphone or smartwatch as a second factor. The enrolled secondary device is at the user's desk, pocket, backpack, or body, in close proximity to the login device. Like most TFA mechanisms, the first two conditions are verified over one instance. The third condition is added to our protocol to establish a long-term trust relationship between the designated and secondary devices.

The protocol consists of three phases. (a) the enrollment phase, (b) the first-factor authentication, and (c) the second-factor authentication. The enrollment phase is a one-time effort to enroll the login device $\mathcal{D}$ and the secondary device $\mathcal{S}$ with the verifier $\mathcal{V}$. During this phase, the user establishes a login username and password with the web service and also enables the second authentication factor.

Upon enrollment, the login device and the secondary device initiate the *contact trace* collection. A contact trace is a continuously updated token that is maintained by $\mathcal{D}$ and $\mathcal{S}$ by periodically exchanging random nonces when they are in proximity. We call such nonces *heartbeats*. The main idea is that if the attempted login is performed from the designated device $\mathcal{D}$, then $\mathcal{D}$ can additionally provide evidence of its identity by using the contact trace that is only known to $\mathcal{D}$ and $\mathcal{S}$. Note that the username and password authenticate the user and not the designated device itself.

Fig. 1 shows a high-level overview of the first and second authentication factors. In the first-factor authentication, the user logs in to the web service using $\mathcal{D}$ (we treat the case where the user accesses the device via a public device $\mathcal{T}$, separately) by entering her username and password. The authentication also establishes a session key K_{VD} between $\mathcal{V}$ and $\mathcal{D}$, which is derived from the user password. The web service is authenticated using conventional methods such as server certificates. Completing the first-factor authentication automatically triggers TFA.

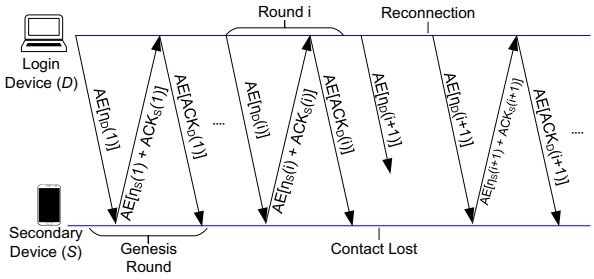


Fig. 3. Timeline of contact trace collection. The protocol proceeds in rounds. The two devices exchange two random nonces per round using authenticated encryption (AE). Receipt of nonces is acknowledged.

In the second-factor authentication phase, the verifier first challenges $\mathcal{D}$ and $\mathcal{S}$ to prove that they have been in prolonged contact. The two devices provide the verifier with proof of their contact trace, thus authenticating the identity of the designated device. Finally, the verifier challenges $\mathcal{D}$ and $\mathcal{S}$ to prove that they are in physical proximity. Proximity verification is required to defend against a MiTM attack where the second factor is automatically executed while $\mathcal{S}$ is far away from $\mathcal{D}$. Note that a contact trace is not sufficient to verify the proximity of $\mathcal{D}$ and $\mathcal{S}$ at a particular instance, as the trace is periodically collected and indicates past contact. Even if a fresh heartbeat exchange is required to complete the second factor, that heartbeat can be relayed by a MiTM adversary when $\mathcal{D}$ and $\mathcal{S}$ are far away. To prove physical proximity, $\mathcal{D}$ and $\mathcal{S}$ simultaneously record ambient RF signals. If $\mathcal{D}$ and $\mathcal{S}$ are in close proximity, they will overhear highly correlated ambient signals.

If the second factor fails, a default mode is triggered where the user needs to authorize the second factor by, for example, pressing a button in his secondary device (e.g., in the Duo application). Before presenting ZITA in detail, we describe the two main building primitives, namely contact tracing and proximity verification.

4.2 Secure Contact Tracing

The main idea of a contact trace is to use the frequent co-presence of the login and secondary devices to build a form of identity authentication for the login device. Such authentication cannot be achieved with the first factor, as the username and password are tied to the user identity and remain the same regardless of the login device.

To create a contact trace, the login device $\mathcal{D}$ and the secondary device $\mathcal{S}$ periodically exchange random nonces we call "heartbeats." The heartbeats are used to continuously update a token known only to $\mathcal{D}$ and $\mathcal{S}$. Security is drawn from the secrecy of the trace that accumulates with time and contact between $\mathcal{D}$ and $\mathcal{S}$. Fig. 3 shows the timeline of the contact trace collection. Tracing is initialized after the

enrollment phase is completed with the execution of the first round called, *the genesis round*. Heartbeats are continuously collected by completing rounds periodically and are "chained" into the token The steps for generating a contact trace are as follows:

1) **Initialization:** The device $\mathcal{D}$ securely broadcasts a request-to-trace message $\text{AE}_{K_{DS}}$(RTC), containing ID_D and ID_S. Secondary device $\mathcal{S}$ replies with a confirm-to-trace message $\text{AE}_{K_{DS}}$(CTC), containing ID_S and ID_D.
2) **Heartbeat Exchange:** The devices $\mathcal{D}$ and $\mathcal{S}$ periodically exchange heartbeats, which are randomly generated nonces. The heartbeat exchange proceeds in rounds. On the i^{th} round, $\mathcal{D}$ sends $\text{AE}_{K_{DS}}(\eta_D(i))$ to $\mathcal{S}$. Upon correct receipt of $\eta_D(i)$ (verified by the attached MAC), $\mathcal{S}$ replies with its own nonce, $\text{AE}_{K_{DS}}(\eta_S(i))$. Reliability of the exchange is provided with a duplex stop-and-go protocol, by including ACKs in each direction [49]. Only if the successful nonce exchange is confirmed during the i^{th} round, do the devices proceed to round $i+1$.
3) **Generating contact tokens:** Upon the successful completion of each round i, $\mathcal{D}$ and $\mathcal{S}$ compute the following, respectively:

$$Tr_D(i) := f(Tr_D(i-1)||\eta_D(i)||\eta_S(i)),$$
$$Tr_S(i) := f(Tr_S(i-1)||\eta_D(i)||\eta_S(i)).$$

For the first round, $Tr_D(0) = Tr_S(0) = 0$. Here, $f(\cdot)$ is a key derivation function (KDF).

The contact tokens generated by $\mathcal{D}$ and $\mathcal{S}$ are the digest of random nonces exchanged while the two devices are in proximity. The heartbeats are hashed into a continuously updating digest, starting from the genesis round. Therefore, all heartbeats contribute to the composition of the tokens. This prevents the token compromise even if the shared secret K_{DS} used to secure the heartbeat exchange is compromised sometime after the genesis round. If the secondary device were to be remotely compromised, the trace allows for the recovery of trust once the compromise is over (i.e., the device returns to the user's control) due to the automatic renewal of the trace. As the frequency of device compromise events is expected to be low, we fixed the frequency of heartbeat exchanges to once every half hour. This frequency can be tuned to different security requirements at the expense of energy consumption. It should be noted that the secrecy of the contact trace is guaranteed by the encryption that is applied to the heartbeats and the use of a hash chain to generate the trace. Even if the adversary could decrypt the heartbeats due to a key compromise, the contact trace still remains secure if the genesis block is unknown.

Trace Reliability. In the absence of attacks, $\mathcal{D}$ and $\mathcal{S}$ exchange heartbeats using a reliable stop-and-go protocol. One party will add a new heartbeat to the trace only if the other party acknowledges the correct reception of the previous heartbeat. This ensures that both devices are synchronized to the same trace and can account for trace interruptions. A possible interruption scenario is heartbeat corruption due to impairments of the wireless channel. Such corruptions are detected using error detection codes and are remedied using the retransmission mechanism of the stop-and-go protocol.

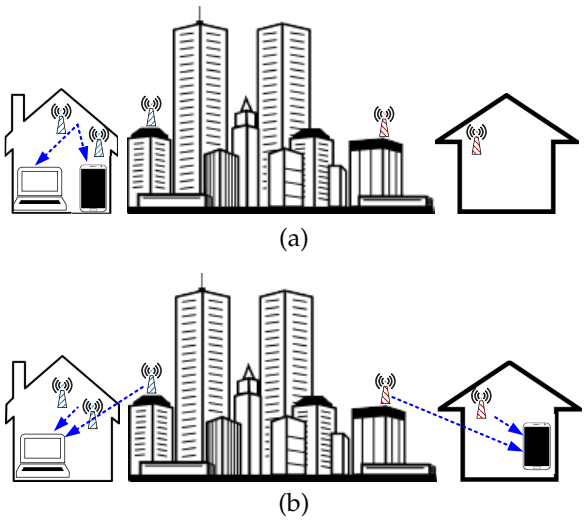


Fig. 4. (a) Proximity verification of $\mathcal{D}$ and $\mathcal{S}$ by collecting ambient RSS, and (b) failure in verification when $\mathcal{D}$ and $\mathcal{S}$ are in different locations.

Another scenario includes the temporary separation of $\mathcal{D}$ and $\mathcal{S}$. This could occur for a number of reasons such as either device being powered off, or being physically away from each other. In that case, the collection of the contact trace is temporarily halted in an automated fashion, since the lack of ACKs does not allow the trace advancement at either party. For more details of the stop-and-go protocol, we ask the readers to refer to [49].

4.3 Proximity Verification

Proximity verification between $\mathcal{D}$ and $\mathcal{S}$ is an additional step to ensure the user is attempting a login from $\mathcal{D}$ aimed at preventing MiTM attacks. Note that proximity cannot be verified via the long-term contact trace because the trace collection is interrupted when the devices are far away. Moreover, even if a fresh heartbeat exchange is requested upon the completion of the first factor, a MiTM adversary could relay heartbeats over fast links to make $\mathcal{D}$ and $\mathcal{S}$ appear close (see attack described in Fig. 2(a)).

To verify proximity, we use a simple in-band method that relies on the large-scale fading effects of the RF channel [50]. Large-scale fading is the result of signal attenuation due to signal propagation through large distances and diffraction around large objects in the propagation path. Since the large-scale fading is impacted by terrain configuration, it significantly varies in different environments.

Proximity verification method: The chief idea is presented in Fig. 4. The two devices $\mathcal{D}$ and $\mathcal{S}$ simultaneously sample the ambient received signal strength (RSS) on a pre-agreed channel for a period of time. The devices report their RSS samples to the verifier who eliminates short-scale fading by applying an M-point moving average on the collected samples and then computes the sample correlation. If $\mathcal{D}$ and $\mathcal{S}$ are co-located as shown in Fig. 4(a), the RSS correlation is expected to be high. This is because the two devices sample

the same ambient signals and are in a similar environment. On the other hand, if $\mathcal{S}$ is away from $\mathcal{D}$ as shown in Fig. 4(b), the RSS measurements will exhibit low correlation because different RF sources are sampled and RF signals propagate in different channels. Formally, proximity verification is implemented in two phases, namely a collection phase and a proximity verification phase.

RSS Collection Phase. In this phase, the login device $\mathcal{D}$ and the secondary device $\mathcal{S}$ sample an agreed frequency at an agreed sample rate for a fixed time period.

1) The login device $\mathcal{D}$ sends the sampled frequency band F, the start time t_s, sample rate r, and the number of samples n to the secondary device $\mathcal{S}$.
$$\mathcal{D} \rightarrow \mathcal{S} : AE_{K_{DS}}(F, t_s, r, n).$$
2) The login device and the secondary device simultaneously collect RSS samples Γ_D and Γ_S, respectively.
$$\Gamma_D = \{\gamma_D(1), \gamma_D(2), \ldots, \gamma_D(n)\},$$
$$\Gamma_S = \{(\gamma_S(1), \gamma_S(2), \ldots, \gamma_S(n)\},$$
where $\gamma_X(i)$ is the i-th RSS sample collected by $X = \{\mathcal{D}, \mathcal{S}\}$.
3) The login device and the secondary device report Γ_D and Γ_S to the verifier signed and encrypted with the respective session keys K_{VD} and K_{VS}.

Proximity Verification Phase. In this phase, $\mathcal{V}$ verifies the physical proximity of $\mathcal{D}$ and $\mathcal{S}$ by computing the correlation between the reported RSS measurements Γ_D and Γ_S.

5) The verifier separates Γ_D and Γ_S into K subsets of size N samples. Let Γ_X^k denote the k-th subset of Γ_X, $X = \{\mathcal{D}, \mathcal{S}\}$.
6) The verifier applies an M-point moving average to each subset. The M-moving average for the i-th RSS sample of Γ_X^k, $X = \{\mathcal{D}, \mathcal{S}\}$ is given by
$$\overline{\gamma_X}(i) = \frac{\gamma_X(i - \lfloor \frac{M}{2} \rfloor) + ... + \gamma_X(i) + ... + \gamma_X(i + \lfloor \frac{M}{2} \rfloor)}{M} \quad (1)$$
7) The verifier computes the correlation $\rho(k)$ for the subsets Γ_D^k and Γ_S^k for $k = 1, 2, \cdots, K$ using Pearson's correlation function,
$$\rho(k) = \frac{\sum_{i=1}^{\ell} (\overline{\gamma_D}(i) - \overline{\gamma_D})(\overline{\gamma_S}(i) - \overline{\gamma_S})}{\sqrt{\sum_{i=1}^{\ell} (\overline{\gamma_D}(i) - \overline{\gamma_D})^2} \sqrt{\sum_{i=1}^{\ell} (\overline{\gamma_S}(i) - \overline{\gamma_S})^2}}, \quad (2)$$
where $\ell = \frac{n}{K}$ is the size of each subset. The verifier obtains K correlation values $\rho(1), \rho(2), \ldots, \rho(K)$.
8) The verifier compares each correlation value $\rho(k)$ with a passing threshold τ. If a fraction α ($0 < \alpha \leq 1$) of correlation values exceeds the passing threshold τ, the verifier ACCEPTS. Otherwise, the verifier REJECTS. That is, the proximity verification test is passed if
$$\sum_{k=1}^{K} \frac{I(\rho(k) \geq \tau)}{K} \geq \alpha, \quad (3)$$
where $I(\cdot)$ is the indicator function.

Parameter α can be tuned to drive the passing rate close to one when $\mathcal{D}$ and $\mathcal{S}$ are in proximity while reducing the passing rate close to zero when $\mathcal{D}$ and $\mathcal{S}$ are far apart. We investigate the selection of all proximity test parameters (τ, α, K, M) in Section 6.1.

4.4 The ZITA Protocol

The ZITA protocol consists of three phases: (a) the enrollment phase, (b) the first-factor authentication, and (c) the second-factor authentication. We present each phase in detail.

4.4.1 Enrollment Phase

The enrollment phase is a one-time effort to distribute the credentials necessary for the TFA. Initially, the user enrolls the designated login device $\mathcal{D}$ and the secondary device $\mathcal{S}$ with the verifier $\mathcal{V}$. On registration, the user chooses a username ID_D, and password. The verifier stores the username and the hashed password. Moreover, the verifier establishes a long-term pairwise key K_S with $\mathcal{S}$. Pairing with $\mathcal{S}$ can be achieved with any conventional method designated by the verifier. Finally, the login device $\mathcal{D}$ and the secondary device $\mathcal{S}$ establish a pairwise key K_{DS}. The final credentials held by each entity are as follows.

$$\begin{aligned} \mathcal{D} &: ID_S, K_{DS}, \\ \mathcal{S} &: K_S, ID_D, K_{DS} \\ \mathcal{V} &: h(\text{pswd}), K_S, ID_D, ID_S. \end{aligned}$$

4.4.2 First Factor Authentication

During the first-factor authentication, the user enters the username and password on $\mathcal{D}$. Upon correct login, $\mathcal{V}$ and $\mathcal{D}$ use the password to establish a session key K_{VD}. In addition, $\mathcal{S}$ establishes the pairwise session key K_{VS} with the verifier $\mathcal{V}$, using long-term key K_S. The first-factor authentication can use any of the existing methods for agreeing to session keys such as a Password-Authenticated Key Exchange (PAKE) [51], an authenticated Diffie-Hellman (DH) key exchange [52], or other methods [53].

4.4.3 Second Factor Authentication

Upon successful completion of the first factor, the second factor is automatically triggered by the verifier who challenges $\mathcal{D}$ and $\mathcal{S}$ to provide the same contact trace and verify their proximity. If the automated second factor fails, the verifier $\mathcal{V}$ defaults to requesting user approval on $\mathcal{S}$ by the push of a button (similar to the Duo application). The steps of the second factor are as follows.

1) **Initialization:** $\mathcal{V}$ sends a request-to-authenticate (RTA) message containing one challenge (random nonce) η to $\mathcal{S}$ and $\mathcal{D}$.
$$\mathcal{V} \rightarrow \mathcal{D} : m_V(1) := \text{AE}_{K_{VD}}(RTA, \eta), \quad (4)$$
$$\mathcal{V} \rightarrow \mathcal{S} : m_V(2) := \text{AE}_{K_{VS}}(RTA, \eta). \quad (5)$$
2) **Token Generation:** Upon receipt of $\widehat{\eta}$, $\mathcal{D}$ generates token $TK_D = f(\widehat{\eta}, Tr_D)$ and $\mathcal{S}$ generates token $TK_S =$

$f(\widehat{\eta}, Tr_S)$, where $f(\cdot)$ is a key derivation function and Tr_S and Tr_D are the contact traces collected by $\mathcal{S}$ and $\mathcal{D}$, respectively. The secondary device sends TK_S to $\mathcal{V}$ via authenticated encryption.

$$\mathcal{V} \leftarrow \mathcal{S}: \quad m_S(1) := \mathrm{AE}_{K_{VS}}(TK_S). \tag{6}$$

3) **Identity Verification:** The verifier $\mathcal{V}$ and device $\mathcal{D}$ independently generate secrets:

$$\mathcal{V} \quad : \quad TK'_{VD} := f(\widehat{TK}_S, K_{VD}), \tag{7}$$
$$\mathcal{D} \quad : \quad TK''_{VD} := f(TK_D, K_{VD}). \tag{8}$$

The verifier $\mathcal{V}$ and $\mathcal{D}$ mutually verify that secrets TK'_{VD} and TK''_{VD} they independently generated, match. This is done via a challenge-response verification.

$$\mathcal{V} \rightarrow \mathcal{D} \quad : \quad \eta_1, \tag{9}$$
$$\mathcal{V} \leftarrow \mathcal{D} \quad : \quad \eta_2, m_D(1) := f(\widehat{\eta}_1, \eta_2, TK''_{VD}) \tag{10}$$
$$\mathcal{V} \rightarrow \mathcal{D} \quad : \quad m_V(3) := f(\widehat{\eta}_2, \eta_1, TK'_{VD}). \tag{11}$$

The verifier accepts if

$$f(\eta_1, \widehat{\eta}_2, TK'_{VD}) \stackrel{?}{=} \widehat{m}_D(1). \tag{12}$$

The device accepts if

$$f(\eta_2, \widehat{\eta}_1, TK''_{VD}) \stackrel{?}{=} \widehat{m}_V(3). \tag{13}$$

4) **Proximity Verification:** The device $\mathcal{D}$ initiates the ambient RSS collection by sending a message to $\mathcal{S}$

$$\mathcal{S} \leftarrow \mathcal{D} : m_D(2) := \mathrm{AE}_{K_{DS}}(F, t_s, r, n).$$

where F is the sampled frequency band, t_s is the start time for the sample collection, r is the sampling rate, and n is the number of collected samples. The secondary device $\mathcal{S}$ verifies the authenticity of $m_D(3)$ and $\mathcal{D}$ and $\mathcal{S}$ capture RSS samples Γ_D and Γ_S, respectively. The samples are sent to $\mathcal{V}$.

$$\mathcal{V} \leftarrow \mathcal{D}: \quad m_D(3) := \mathrm{AE}_{K_{VD}}(\Gamma_D), \tag{14}$$
$$\mathcal{V} \leftarrow \mathcal{S}: \quad m_S(2) := \mathrm{AE}_{K_{VS}}(\Gamma_S). \tag{15}$$

The verifier $\mathcal{V}$ verifies the authenticity and integrity of $\widehat{m_D}(4)$ and $\widehat{m_S}(2)$, decrypts them and recovers the RSS samples $\widehat{\Gamma}_D$ and $\widehat{\Gamma}_S$, respectively. $\mathcal{V}$ then computes

$$\rho(1), \rho(2), \cdots, \rho(K),$$

using (2). If

$$\sum_{k=1}^{K} \frac{I(\rho(k) \geq \tau)}{K} \stackrel{?}{\geq} \alpha,$$

where α, K, and τ are preselected test parameters, then $\mathcal{V}$ ACCEPTS and SUCCESS is displayed on $\mathcal{D}$ and $\mathcal{S}$.

5) **Request to PUSH:** If $\mathcal{V}$ REJECTS in the previous steps, the verifier sends an authenticated and encrypted request-to-push (RTP) message to $\mathcal{S}$, requiring the user's approval via the push of a button. If user approval is received within a preset time limit, $\mathcal{V}$ ACCEPTS. Both $\mathcal{D}$ and $\mathcal{S}$ (interface permitting) display SUCCESS. If the verifier REJECTS, it sends a reject notice to $\mathcal{S}$.

Fig. 5 presents the formal protocol description of the second factor. To pass authentication, the login device $\mathcal{D}$ must prove knowledge of the contact trace (i.e., $Tr_D = Tr_S$). Note that this step is performed without revealing the traces to the verifier. Instead, $\mathcal{D}$ and $\mathcal{S}$ generate two tokens using their traces and the random nonces provided by $\mathcal{V}$'s challenge. Upon verification of the contact trace the proximity verification test is performed.

Second-factor authentication from an unregistered device. If the user attempts to log in from an unregistered device $\mathcal{T}$, TFA is always implemented with a push of a button at $\mathcal{S}$. The differentiation between $\mathcal{T}$ and $\mathcal{D}$ is based on the contact trace generated between $\mathcal{D}$ and $\mathcal{S}$. As $\mathcal{T}$ has not generated a contact trace with $\mathcal{S}$, the verifier $\mathcal{V}$ can recognize that the login is attempted from an unregistered device and skip the proximity verification. A request to push a button is sent to the secondary device to complete TFA.

5 Security Analysis

In this section, we analyze the security of ZITA under the threat model of Section 3.2. We focus on the security of the second-factor authentication, as the first factor is assumed to be compromised. We analyze the attacks defined by the threat model and also perform a formal analysis of ZITA under the same threat model on ProVerif [54]. The results of the formal analysis are presented in Appendix A.

Using an unregistered device $\mathcal{T}_M$. An adversary that has compromised the first factor may try to authenticate to the web service from an unregistered device $\mathcal{T}_M$. This represents the most common attack scenario. Upon completing the first factor, the verifier establishes a pairwise session key K_{VT_M} with $\mathcal{T}_M$. He then triggers the second-factor authentication with the transmission of $m_V(1)$ and $m_V(2)$ to $\mathcal{T}_M$ and $\mathcal{S}$, respectively. The terminal $\mathcal{T}_M$ will decrypt the message and will be required to send the token generated using contact trace Tr_D. As $\mathcal{T}_M$ does not have access to the contact trace Tr_D, the second-factor authentication cannot proceed further. The verifier aborts the proximity test and defaults to request the push of a button at $\mathcal{S}$. Without access to $\mathcal{S}$, the second-factor authentication fails, and the user is notified of the failed login attempt.

Physical access to login device $\mathcal{D}$. Let us now consider the case where the adversary has gained physical access to the designated login device $\mathcal{D}$. This scenario can occur when a close contact (e.g. friend) with physical access to the user's space uses the login device while the legitimate user who controls the secondary device is away [55]. The close contact may also be aware of the user's first-factor credentials.

Upon completing the first factor, the verifier triggers the second-factor authentication. Given that the adversary uses $\mathcal{D}$ to log in, the device will provide the same contact trace token as $\mathcal{S}$ and the contact trace verification will succeed. The second factor will proceed to the proximity verification phase with the collection of RSS at $\mathcal{D}$ and $\mathcal{S}$. However, because $\mathcal{D}$ and $\mathcal{S}$ are not in proximity, they are unable to communicate to select the sampled frequency, the start time,

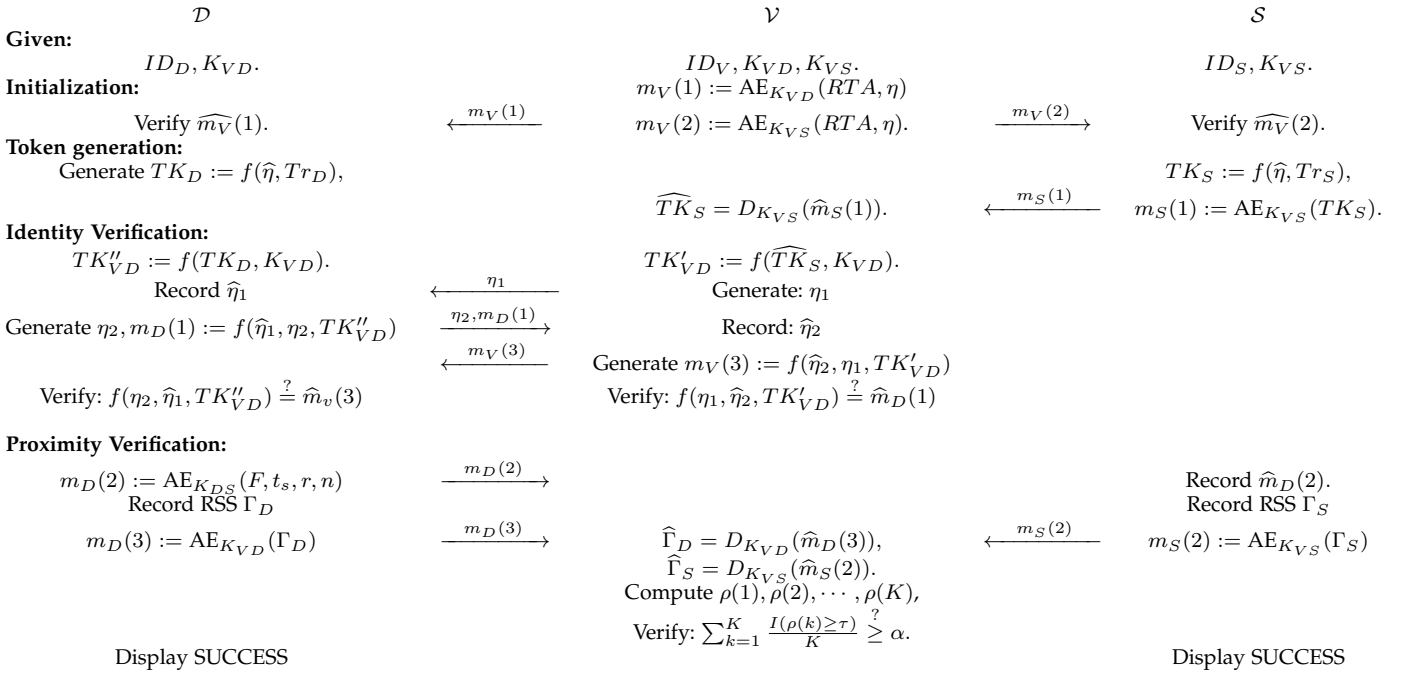


Fig. 5. The second factor authentication phase of ZITA.

and the period, and therefore the verification fails. Note that the case where $\mathcal{M}$ has simultaneous physical access to $\mathcal{D}$ and $\mathcal{S}$ is in close proximity is not considered part of our threat model. It is assumed that the user is in possession of $\mathcal{S}$ and could visually detect the presence of an adversary at $\mathcal{D}$.

5.1 Resistance to MiTM Attacks

In a MiTM attack, $\mathcal{M}$ is assumed to have physical access to the designated login device $\mathcal{D}$ and is also co-located with $\mathcal{S}$, as shown in Fig. 2(a). The adversary uses $\mathcal{D}$ to pass the first-factor authentication, which automatically triggers the second factor. Given that the service is accessed from $\mathcal{D}$, the contact trace verification is successful, as both $\mathcal{D}$ and $\mathcal{S}$ provide tokens from the same trace. The adversary could attempt to defeat the RSS correlation test by replaying ambient signals collected around the primary device $\mathcal{D}$ to the location of $\mathcal{S}$ and vice versa. Although this attack seems plausible, we emphasize that it is quite difficult to launch because of the unpredictability of wireless channels. The adversary requires knowledge of all the channels between $\mathcal{D}$, $\mathcal{S}$, and the respective access points operating in each vicinity. As this is particularly difficult to obtain or predict, we consider an overshadowing attack where the adversary attempts to diminish the randomness induced by the ambient sources by transmitting a strong signal in the line-of-sight (LoS) path to the primary and secondary devices.

The second-factor authentication proceeds to the proximity verification. When $\mathcal{D}$ and $\mathcal{S}$ are not in proximity, the only way for the adversary to pass the proximity verification test is if the two devices measure highly correlated RSS values. A MiTM adversary can influence the RSS measured by each device as follows. The adversary deploys two rogue devices $\mathcal{M}_D$ and $\mathcal{M}_S$ in the proximity of $\mathcal{D}$ and $\mathcal{S}$, respectively, as shown in Fig. 2(a). He then deploys a fast link between $\mathcal{M}_D$ and $\mathcal{M}_S$ using the Internet infrastructure. The adversary relays the message $\mathrm{AE}_{K_{DS}}(F, t_s, r, n)$ between $\mathcal{D}$ and $\mathcal{S}$ to initiate the RSS collection process. Finally, because the two devices are in different environments, the adversary launches an active attack by synchronously transmitting the same signal at the two ends of the high-speed link.

Resistance to this active attack is drawn from three factors: (a) the sampled frequency band is unknown to the adversary. Although this provides some protection, the adversary can transmit in all relevant bands (e.g., all Wi-Fi bands) to ensure that the two devices sample the adversarial signal, (b) the randomness of the wireless channel when $\mathcal{D}$ and $\mathcal{S}$ are in different environments, and (c) other ambient transmissions on the same frequency. Despite the injection of the same waveform, $\mathcal{D}$ and $\mathcal{S}$ will record low correlated RSS values even if the distance to each rogue transmitter is kept the same. This is due to the different multipath and diffraction components in each environment. Finally, other devices operating on the same band randomize the received RSS. We experimentally demonstrate resistance to an active MiTM attack in Section 6.

5.2 Resistance to Co-located Attacks

In a co-located attack, the adversary is in the vicinity of $\mathcal{D}$ and $\mathcal{S}$, as shown in Fig. 2(b), but does not have physical access to either [31]. The adversary uses an unregistered device $\mathcal{T}_M$ to log in to $\mathcal{V}$. This scenario is relevant when the user is in a public space such as a coffee shop, or an airport terminal and the adversary is nearby. Similar to

the previous attacks, the adversary has access to the first-factor credentials and has completed it successfully from $\mathcal{T}_M$. Upon successful completion of the first factor, the verifier establishes a key K_{VT_M} with $\mathcal{T}_M$, and the second factor is automatically triggered. The verifier challenges $\mathcal{T}_M$ and $\mathcal{S}$, to provide proof of the contact trace by sending $m_V(1)$ and $m_V(2)$ to each device, respectively. As $\mathcal{T}_M$ does not have the contact trace Tr_D, the second-factor authentication defaults to the push button request to the user's secondary device. Without access to $\mathcal{S}$, the second-factor authentication fails, and the user is notified of the failed login attempt.

5.3 Loss of the Secondary Device

We now examine the event of loss or theft of the secondary device $\mathcal{S}$. Although the adversary is in physical possession of $\mathcal{S}$, security methods such as numerical pins, face recognition, etc. prevent the adversary from unlocking the device. Without access to the device, the adversary can only move the device to a location of his own choosing. In this case, the following scenarios apply: (a) the contact trace verification will fail because $\mathcal{T}_M$ does not possess the trace Tr_D. The second-factor authentication will default to the request for a push button by the user. Without the ability to unlock the secondary device, the login attempt fails. (b) If the adversary can be in physical proximity to the primary device (co-located attack), the attack will still fail if the primary device does not attempt a log-in. (c) If the adversary gains physical access to $\mathcal{D}$ and is in possession of $\mathcal{S}$, then he can log in to the web service without unlocking $\mathcal{S}$. This is the intended usability scenario of zero-effort authentication and cannot be prevented by any zero-effort method. (d) Finally, if the adversary can unlock $\mathcal{S}$ and access the push-button application, then she can use any public device $\mathcal{T}$ to log in to the service and pass the TFA at $\mathcal{V}$. This occurs regardless of the success of the contact trace and proximity verification tests. Even if the latter tests fail, the second factor will default to a push button. With unrestricted access to $\mathcal{S}$, the second factor will be successfully completed. This attack can be thwarted if access from non-designated devices is prohibited and the push-button operation at $\mathcal{S}$ is disabled.

5.4 Compromise of the Login Device

Let the login device $\mathcal{D}$ be compromised. This can occur by remotely installing malware on $\mathcal{D}$ and gaining access to the session keys K_{VD}, K_{DS}, and the contact token $Tr_D(i)$ collected up to the current round i. The adversary can attempt to log in from $\mathcal{D}$ itself or use an unregistered device $\mathcal{T}_M$. For all practical purposes, $\mathcal{D}$ and $\mathcal{T}_M$ are indistinguishable, as they hold the same secrets. We analyze the resistance to the compromise of $\mathcal{D}$ in two scenarios: (a) $\mathcal{D}$ and $\mathcal{S}$ are far apart and (b) $\mathcal{D}$ and $\mathcal{S}$ are co-located.

Scenario 1: Secondary device is away. With the knowledge of K_{VD}, K_{DS}, and the contact token $Tr_D(i)$, the adversary device $\mathcal{T}_M$ can pass the contact trace verification that requires proof of knowledge of $Tr_D(i)$. Upon completion of this phase, the second-factor authentication proceeds with the proximity verification. Because $\mathcal{T}_M$ and $\mathcal{S}$ are far apart (the user carrying the secondary device is away from the adversary's device), the proximity verification test will fail, as the two devices will record uncorrelated ambient RSS values. In this case, the user will receive a notification of a login attempt and will recognize the unauthorized access. The adversary could use $\mathcal{D}$ instead of $\mathcal{T}_M$ to log in to the web service. Similarly, if the $\mathcal{D}$ and $\mathcal{S}$ are far apart, the proximity verification test will fail and the push-button process of the second-factor authentication will be triggered.

Scenario 2: Login and secondary device are in proximity. If the compromised login device $\mathcal{D}$ and the secondary device are in proximity when the adversary compromises $\mathcal{D}$, both the contact trace verification and the proximity verification will be successful, as the adversary knows $Tr_D(i)$ and also records correlated ambient RSS with $\mathcal{S}$. Note that this adversary model is quite strong as it requires the remote compromise of $\mathcal{D}$ and physically tracking down the user. The only possibility for detecting the attack is the notification of the successful login on the secondary device.

Automated recovery. Whereas ZITA does not prevent the successful second-factor authentication under the strong model of compromising the first-factor and $\mathcal{D}$ and being co-located with $\mathcal{S}$, it offers an automated recovery mechanism once the malware that is used to control $\mathcal{D}$ has been removed (e.g., using nightly malware scans). Because of the heartbeat exchange, the trace $Tr_D(i)$ is periodically updated, creating a rolling secret that can be used to renew trust once $\mathcal{D}$ has been secured. That is, without access to the heartbeats exchanged at round $i+i$, the trace $Tr_D(i+1)$ remains secret even if $Tr_D(i)$ is exposed. This is because $Tr_D(i+1)$ is generated by applying a key derivation function with $Tr_D(i)$, $\eta_D(i+1)$, and $\eta_S(i+1)$ as inputs.

6 Evaluation

In this section, we perform a two-fold evaluation of ZITA. Initially, we evaluated the feasibility and security of the proximity verification method presented in Section 4.3. We further evaluated the usability of ZITA by developing a prototype application on the Android platform. The implementation on the Android 10 platform of the primary/secondary device app as well as the verifier are available at https://github.com/kaustubh-gupta/ZITA. The details of the ZITA application and user study using the application are provided in Appendix B and C, respectively.

6.1 Proximity Verification

6.1.1 RSS Correlation with Distance.

We first evaluated the correlation of ambient RSS with distance. We used two USRP N200 radios equipped with VERT2450 antennas acting as $\mathcal{D}$ and $\mathcal{S}$. The two USRPs simultaneously sampled ambient Wi-Fi signals at 2.437GHz with a 20MHz bandwidth, which corresponds to Wi-Fi channel 6. We set the gain of the antenna to 20dB and the sample rate to 20Hz. A laptop was connected to the USRP for

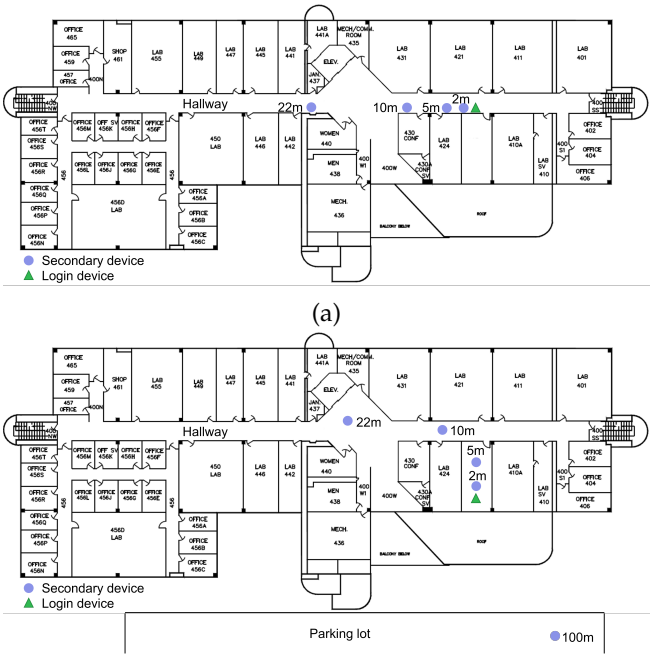


Fig. 6. The indoor environment for the proximity verification experiments with (a) Setup 1: LoS channel, and (b) Setup 2: NLoS and LoS channels.

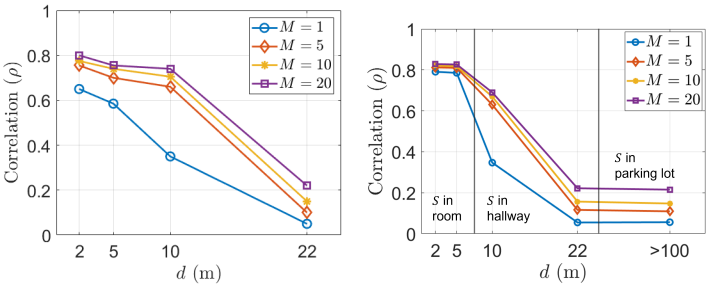


Fig. 7. (a) RSS correlation in Setup 1 for different moving average window M and (b) RSS correlation in Setup 2.

recording and processing the RSS data acting as $\mathcal{V}$. Initially, we placed $\mathcal{D}$ and $\mathcal{S}$ in the hallway of the fourth floor of the UArizona ECE building shown in Fig. 6(a) (Setup 1). The distance between the two devices varied from 2m to 22m. We also placed $\mathcal{S}$ at the parking lot outside the ECE building at an approximate distance of 100m to study a scenario where the user is away from $\mathcal{D}$. In Setup 2 Fig. 6(b), $\mathcal{D}$ was inside a lab and $\mathcal{S}$ was placed (a) inside the same lab at distances of 2m and 5m, (b) at the hallway at distances 10m and 22m, and (c) at the parking lot at an approximate distance of 100m.

Selecting the moving average window. We first experimented with the moving average window to eliminate the small scaling effect. Fig. 7(a) shows the correlation ρ as a function of the distance d for different moving average window sizes M. A total of 36,000 samples were collected for each distance (30 mins). The samples were organized into subsets of 200 samples (10-sec duration) and the correlation was computed over each subset, applying different moving average windows. We observe an increase in correlation with M at short distances, whereas M's impact diminishes at long

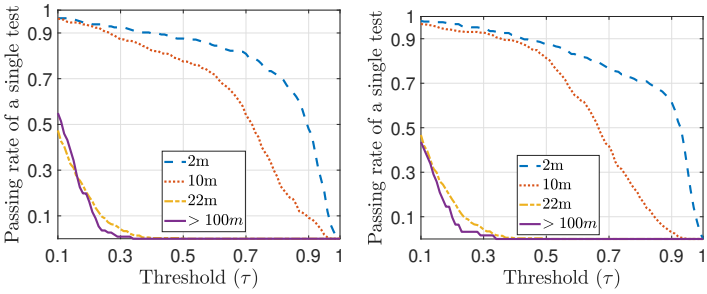


Fig. 8. Single test passing rate as a function of τ for different separations between $\mathcal{D}$ and $\mathcal{S}$.

distances because the ambient RSS sequences sampled by the devices are uncorrelated. Moreover, the correlation gains diminish after $M = 5$. We set $M = 5$ for the remaining of the experiments.

Selecting the multi-test parameters. The proximity verification test as expressed in (3) depends on the single correlation test threshold τ, the number of correlation tests K, and the fraction of single correlation tests α that must be passed. For a total of K tests, authentication is passed if $\sum_{k=1}^{K} \frac{I(\rho(k)\geq\tau)}{K} \geq \alpha$ where $\rho(k)$ denotes the correlation value for each subset k. Assuming independent tests due to the use of different subsets, the probability of passing proximity verification is

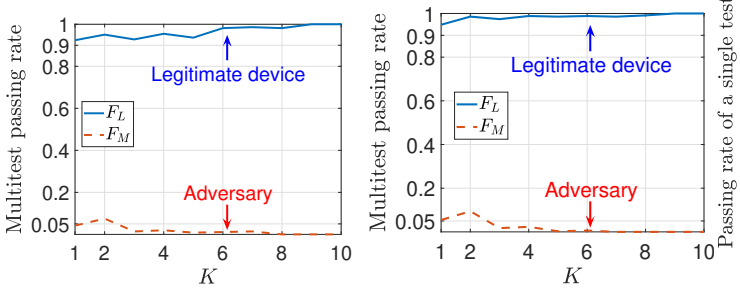
$$F = \sum_{x=\lceil \alpha \cdot K \rceil}^{K} \binom{k}{x} (f)^x \cdot (1-f)^{K-x}. \tag{16}$$

where $\binom{k}{x}$ is the binomial coefficient. Let F_L denote the probability of passing verification for the secondary device and F_M be the passing probability for the adversary $\mathcal{M}$. The probabilities F_L and F_M are derived from Eq. (16) by substituting the probability f_L and f_M of passing a single correlation test, given the selection of τ and the locations of $\mathcal{D}$ and $\mathcal{M}$. We use the equal error rate (EER) criterion to select the optimal threshold τ^*. The EER is defined as

$$EER: \quad 1 - F_L = F_M. \tag{17}$$

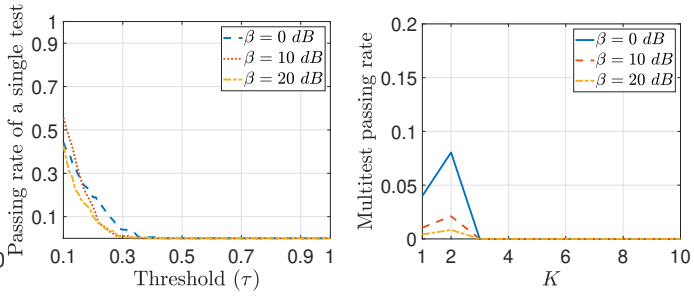
Fig. 8(a) shows the single test passing rate as a function of τ when Setup 1 is used. For each τ, we performed an exhaustive search over the two remaining free variables K and α to minimize the EER. Here, we limited K to 10 to reduce the authentication delay. The EER was minimized when $\tau^* = 0.3$ with the corresponding $\alpha^* = 0.4$ and $K^* = 10$. To compute the EER, $\mathcal{S}$ was assumed to be at 2m from $\mathcal{D}$, whereas $\mathcal{M}$ was placed in the hallway at 22m. The results for Setup 2 are shown in Fig. 7(b). The optimal threshold of $\tau^* = 0.29$ was selected with the corresponding $\alpha^* = 0.44$ and $K^* = 9$ to minimize the EER.

Test Passing Rate. Using the EER criterion, we evaluated the proximity test passing rate for the legitimate device and the adversary. Fig. 9(a) shows the passing rate as a function of the number of tests K, when the secondary device is in the same room as $\mathcal{D}$ at 2m, whereas $\mathcal{M}$ is at the hallway at



(a) Adversary in the hallway (b) Adversary at the parking lot

Fig. 9. Multi-test passing rate as a function of K.

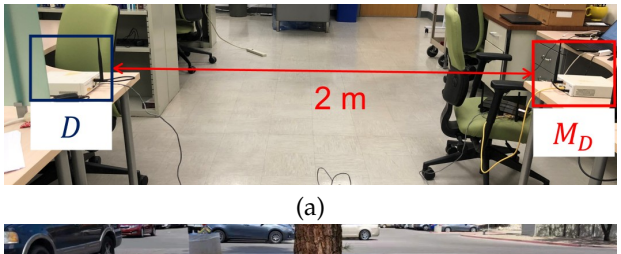


(a) Single test (b) Multi-test

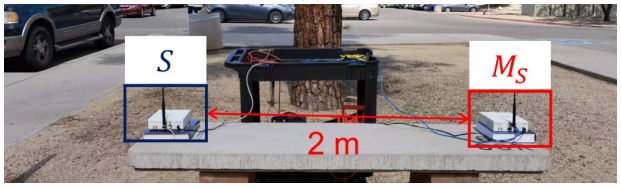
Fig. 11. Passing rate for a MiTM adversary.

22m. Fig. 9(b) shows the passing rate when $\mathcal{M}$ is moved to the parking lot, approximately 100m away. We observe that when $\mathcal{D}$ and $\mathcal{S}$ are in proximity, the proximity test is passed with very high probability. On the other hand, when $\mathcal{S}$ is away, the proximity test fails with almost certainty.

Resistance to MiTM Attacks. We further evaluated the resistance of the proximity test to active MiTM attacks. We replicated the topology shown in Fig. 2(a) by placing the login device $\mathcal{D}$ in the lab room whereas the secondary device $\mathcal{S}$ was placed in the parking lot approximately 100m away. We deployed two additional USRPs $\mathcal{M}_D$ and $\mathcal{M}_S$ to act as the MiTM attackers, as shown in Fig. 10. This scenario represents a user who is far away from the login device. We then placed two transmitters $\mathcal{M}_D$ and $\mathcal{M}_S$, 2m away from $\mathcal{D}$ and $\mathcal{S}$, respectively. Both $\mathcal{M}_S$ and $\mathcal{M}_D$ launched an overshadowing attack by synchronously transmitting the same waveform on Wi-Fi channel 6.



(a)



(b)

Fig. 10. (a) Adversarial transmitter $\mathcal{M}_D$ placed in proximity of $\mathcal{D}$ indoors, and (b) $\mathcal{M}_S$ placed in proximity of $\mathcal{S}$ outdoors.

The strength of the attack, measured by the gain β relative to the ambient RSS, was varied from 0 to 20 dB. Even when the attacker transmitted 20dB higher than the ambient RSS, the correlation of the RSS samples between $\mathcal{D}$ and $\mathcal{S}$ never exceeded 0.1. This is due to the different channels between $\mathcal{D} - \mathcal{M}_D$ and $\mathcal{S} - \mathcal{M}_S$, respectively, despite the symmetric and synchronous nature of the attack. The difference in ambient RSS also contributed to the low correlation.

Fig. 11(a) shows the passing rate for a single test under a MiTM attack for different overshadowing gains. Fig. 11(b) shows the passing rate for the multi-test under the same scenario. We observe that with careful selection of the test parameters, an overshadowing MiTM attack fails to defeat the proximity verification test. This is because the ambient RSS was measured on a particular channel where multiple APs operated (rather than measuring a single AP [56]) and contributed to the signal randomness. Table 2 shows the number of APs, the highest RSS, and the average RSS of all APs that were detected on channel six during our experiments and at the different experiment locations. Even if the adversary accurately applies power control, it is difficult to predict the contribution from co-existing APs.

TABLE 2
Activities observed for the ambient wireless environment

Location	Number of AP	Highest RSS	Average RSS
Outdoor	6	-54 dBm	-67.8 dBm
Indoor (Hallway)	11	-42 dBm	-55.45 dBm
Indoor (Lab)	7	-43 dBm	-52.86 dBm

Note that the average RSS is weak for the outdoor environment (close to -70dBm) which leaves the adversary in control of the RSS at the secondary device. However, security is harnessed by the fact that the indoor channel (where the default device is located) and the outdoor channel (where the secondary device is located) are different. A more elaborate MiTM attack would call for accurate channel estimation to the designated and secondary devices so that the injected signal can be crafted appropriately.

6.2 Performance Analysis

We evaluate the performance of ZITA in terms of the time to complete the TFA authentication and the energy usage on the secondary device which may be battery-constrained.

Delay of ZITA. We evaluated the delay of the ZITA application (presented in Appendix B), between pressing the login button on the primary device and completing the login process. In our experiments, both the primary and secondary devices were implemented using Samsung A9 Smartphones. The online service that performed the credential verification was implemented on a Dell Desktop with an Intel i7 octa-core processor @3.80GHz with 35 GB RAM.

The delay in executing the ZITA protocol consists of two factors: (a) the delay in communicating with the online server

and performing the necessary cryptographic checks and (b) the delay in collecting the RSS samples to perform the proximity verification. For the communication and computation delay, we executed 100 logins from the ZITA application and measured the average delay and standard deviation. This was found to be 536μs$\pm 68\mu$s. Further, the RSS-based proximity test required the collection of 2,000 RSS samples. The typical Wi-Fi sampling rate is about 20MHz which results in a delay of 100ms. The total delay for executing ZITA is therefore $100\text{ms} + 0.6\text{ms} \approx 0.1\text{s}$.

We note that to increase the entropy of the RSS data, one can opt to collect RSS data over a longer time period and still maintain an acceptable time performance. For instance, the two devices can collect samples over a period of 1-2 seconds, and then downsample to perform the proximity test.

Energy consumption of ZITA. Next, we evaluated the energy usage for executing ZITA for the secondary device. There are two components of energy consumption for the ZITA execution. The first component is the heartbeat exchange to maintain the contact trace. The second component is the message exchange during the ZITA execution.

Energy consumption of the heartbeat exchange. We implemented the heartbeat exchange between two smartphones over Bluetooth 5.0. Initially, we used existing measurements to model the per-bit transmission energy and set it to 8nJ/bit [57]. Each heartbeat is 1,024 bits and requires a 24-bit header to be transmitted. yielding an overhead of $8nJ * (24 + 1,024)bits = 8.4\mu J$ per heartbeat.

Energy consumption of the authentication. When ZITA is executed, the secondary device needs to send two messages $m_S(1)$ and $m_S(2)$ to the verifier (see Fig. 5 of the manuscript that outlines the ZITA second factor). Message $m_S(1)$ is the trace of the collected heartbeats whereas is the 2,000 collected RSS samples. In our application, this communication is performed over Wi-Fi, since the verifier is an online server, but can also be implemented over Bluetooth if a smartwatch is using a smartphone as a network gateway. We use a 512-bit hash to generate the trace from the heartbeats. Moreover, each RSS sample was stored with 8-bit accuracy leading to a total of $8 \cdot 2000 = 16000$ bits per proximity verification (2,000 samples collected per verification). Wi-Fi headers are equal to 64 bits and each Wi-Fi packet can accommodate up to 1,500 bytes. In total, the trace requires the transmission of one packet of length 572 bits, and the RSS values require the transmission of one packet of length 12,000 bits and one packet of length 4,128 bits, tallying to a total bit count of 16,700 bits. At 2nJ per bit, the energy cost to execute ZITA at the secondary device becomes $33.4\mu J$.

Experimental evaluation. We further evaluated the energy consumption of ZITA on a Samsung A9 Smartphone. We first established the baseline by leaving the smartphone on for a day with the battery charged at 100%. We then varied the frequency of the heartbeat exchange and the number of TFA executions. Fig. 12, shows the remaining battery percentage after 24 hours compared to the baseline battery drainage (55% remaining battery) with different login attempts. At the suggested rate of 1 heartbeat per 30 minutes, the impact of ZITA is less than 5% and the impact remains low for other frequencies as well.

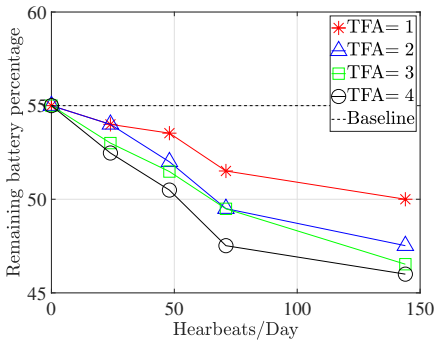


Fig. 12. Remaining battery in $\mathcal{S}$ at the end of the day as a function of the heartbeat rate and the ZITA TFAs executed.

7 Conclusions

We proposed the Zero-Interaction Two-factor Authentication (ZITA) that achieves a new form of in-band second-factor authentication. In ZITA, the first factor is implemented using the conventional username and password methods. The second factor is completed without any human effort provided that the user is not accessing the service from an unregistered public device and a designated secondary device is physically co-present. To automate the second factor, ZITA exploits the long-term contact between the login device and the secondary device such as a smartphone. Moreover, to thwart man-in-the-middle and co-located attacks, ZITA incorporates a proximity verification test that relies on the randomness of ambient RF signals. Moreover, ZITA leverages the prolonged co-presence of mobile devices to refresh the second factor. As long as any part of the proximity trace remains secret, rogue devices with knowledge of the first-factor secrets are eventually identified and revoked. We verified the security properties of ZITA and evaluated the in-band proximity verification with extensive USRP testbed experiments in indoor and outdoor environments.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. This work was supported by ARO grant W911NF-19-1-0050 and NCESR Cycle 17 grant from NPPD. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of the ARO or NPPD.

References

[1] T. Armerding, "The 18 biggest data breaches of the 21st century," https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html, 2018.

[2] M. Javed, "Detecting credential compromise in enterprise networks," Ph.D. dissertation, UC Berkeley, 2016.

[3] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce reuse in wpa2," in *Proc. of the ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1313–1328.

[4] S. Havron, D. Freed, R. Chatterjee, D. McCoy, N. Dell, and T. Ristenpart, "Clinical computer security for victims of intimate partner violence," in *Proc. of USENIX Security*, 2019, pp. 105–122.

[5] C. Wang, S. T. K. Jan, H. Hu, D. Bossart, and G. Wang, "The next domino to fall: Empirical analysis of user passwords across online services," in *Proc. of CODASPY*, 2018.

[6] F. Aloul, S. Zahidi, and W. El-Hajj, "Two factor authentication using mobile phones," in *Proc. of IEEE/ACS International Conference on Computer Systems and Applications*, 2009, pp. 641–644.

[7] J. Lang, A. Czeskis, D. Balfanz, M. Schilder, and S. Srinivas, "Security keys: Practical cryptographic second factors for the modern web," in *Proc. of International Conference on Financial Cryptography and Data Security*, 2016, pp. 422–440.

[8] D. M'Raihi, S. Machani, M. Pei, and J. Rydell, "Totp: Time-based one-time password algorithm," *Internet Request for Comments*, 2011.

[9] All Things Auth, "SMS: The most popular and least secure 2fa method," https://www.allthingsauth.com/2018/02/27/sms-the-most-popular-and-least-secure-2fa-method/, 2018.

[10] K. D. Bowers, A. Juels, R. L. Rivest, and E. Shen, "Drifting keys: Impersonation detection for constrained devices," in *Proc. of IEEE INFOCOM*, 2013, pp. 1025–1033.

[11] "Duo security two-factor authentication," https://duo.com/product/multi-factor-authentication-mfa/authentication-methods/duo-push, 2019.

[12] J. Abbott and S. Patil, "How mandatory second factor affects the authentication user experience," in *Proc. of CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[13] S. Das, A. Kim, B. Jelen, J. Streiff, L. J. Camp, and L. Huber, "Why don't older adults adopt two-factor authentication?" in *Proc. of SIGCHI Workshop on Designing Interactions for the Ageing Populations-Addressing Global Challenges*, 2020, pp. 1–5.

[14] T. J. Pierson, X. Liang, R. Peterson, and D. Kotz, "Wanda: Securely introducing mobile devices," in *Proc. of INFOCOM*, 2016, pp. 1–9.

[15] Y. Wu, A. Khisti, C. Xiao, G. Caire, K.-K. Wong, and X. Gao, "A survey of physical layer security techniques for 5G wireless networks and challenges ahead," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 4, pp. 679–695, 2018.

[16] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam, "ProxiMate: proximity-based secure pairing using ambient wireless signals," in *Proc. of MobiSys*, 2011, pp. 211–224.

[17] L. Shi, M. Li, S. Yu, and J. Yuan, "BANA: Body area network authentication exploiting channel characteristics," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 1803–1816, 2013.

[18] F. Puente, J. Sandoval, P. Hernandez, and C. Molina, "Improving online banking security with hardware devices," in *Proc. of International Carnahan Conference on Security Technology*, 2005, pp. 174–177.

[19] E. Ulqinaku, D. Lain, and S. Capkun, "2FA-PP: 2nd factor phishing prevention," in *Proc. of ACM WiSec*, 2019, pp. 60–70.

[20] S. Jarecki, H. Krawczyk, M. Shirvanian, and N. Saxena, "Two-factor authentication with end-to-end password security," in *Proc. of IACR International Workshop on Public Key Cryptography*, 2018, pp. 431–461.

[21] "Sound login two factor authentication," https://soundlogin.com/, 2019.

[22] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, "Soundproof: usable two-factor authentication based on ambient sound," in *Proc. of USENIX Security*, 2015, pp. 483–498.

[23] M. Wang, W.-T. Zhu, S. Yan, and Q. Wang, "SoundAuth: Secure zero-effort two-factor authentication based on audio signals," in *Proc. of IEEE Conference on Communications and Network Security (CNS)*, 2018, pp. 1–9.

[24] P. Shrestha and N. Saxena, "Listening watch: Wearable two-factor authentication using speech signals resilient to near-far attacks," in *Proc. of ACM conference on security & privacy in wireless and mobile networks*, 2018, pp. 99–110.

[25] S. W. Shah and S. S. Kanhere, "Wi-Auth: Wifi based second factor user authentication," in *Proc. of EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2017, pp. 393–402.

[26] A. Acar, W. Liu, R. Beyah, K. Akkaya, and A. S. Uluagac, "A privacy-preserving multifactor authentication system," *Security and Privacy*, vol. 2, no. 5, p. e88, 2019.

[27] A. Acar, H. Aksu, A. S. Uluagac, and K. Akkaya, "A usable and robust continuous authentication framework using wearables," *IEEE Transactions on Mobile Computing*, vol. 20, no. 6, pp. 2140–2153, 2020.

[28] A. Acar, S. Ali, K. Karabina, C. Kaygusuz, H. Aksu, K. Akkaya, and S. Uluagac, "A lightweight privacy-aware continuous authentication protocol-paca," *ACM Transactions on Privacy and Security (TOPS)*, vol. 24, no. 4, pp. 1–28, 2021.

[29] Y. Ren, P. Wen, H. Liu, Z. Zheng, Y. Chen, P. Huang, and H. Li, "Proximity-Echo: Secure two factor authentication using active sound sensing," in *Proc. of IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[30] Y. Cao, Q. Zhang, F. Li, S. Yang, and Y. Wang, "PPGPass: nonintrusive and secure mobile two-factor authentication via wearables," in *Proc. of IEEE Conference on Computer Communications*, 2020, pp. 1917–1926.

[31] D. Han, Y. Chen, T. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, "Proximity-proof: Secure and usable mobile two-factor authentication," in *Proc. of MobiCom*, 2018, pp. 401–415.

[32] M. Fomichev, J. Hesse, L. Almon, T. Lippert, J. Han, and M. Hollick, "Fastzip: Faster and more secure zero-interaction pairing," in *Proc. of Annual International Conference on Mobile Systems, Applications, and Services*, 2021, pp. 440–452.

[33] E. Yang, S. Fang, and D. Shen, "DASK: Driving-assisted secret key establishment," in *Proc. of IEEE Conference on Communications and Network Security (CNS)*, 2022, pp. 73–81.

[34] K. Lee, N. Klingensmith, D. He, S. Banerjee, and Y. Kim, "ivPair: context-based fast intra-vehicle device pairing for secure wireless connectivity," in *Proc. of ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2020, pp. 25–30.

[35] A. Czeskis, M. Dietz, T. Kohno, D. Wallach, and D. Balfanz, "Strengthening user authentication through opportunistic cryptographic identity assertions," in *Proc. of ACM CCS*, 2012, pp. 404–414.

[36] M. Shirvanian, S. Jarecki, N. Saxena, and N. Nathan, "Two-factor authentication resilient to server compromise using mix-bandwidth devices," in *Proc. of NDSS Symposium*, 2014, pp. 1–16.

[37] Y. Lu, F. Wu, Q. Huang, S. Tang, and G. Chen, "Telling secrets in the light: An efficient key extraction mechanism via ambient light," *IEEE Transactions on Wireless Communications*, vol. 20, no. 1, pp. 186–198, 2020.

[38] "Sign in faster with 2-step verification phone prompts," https://support.google.com/accounts/answer/7026266?co=GENIE.Platform, 2019.

[39] "FIDO universal 2nd factor (U2F) overview," https://fidoalliance.org/specs/fido-u2f-v1.0-nfc-bt-amendment-20150514/fido-u2f-overview.html, 2019.

[40] B. Shrestha, M. Shirvanian, P. Shrestha, and N. Saxena, "The sounds of the phones: dangers of zero-effort second factor login based on ambient audio," in *Proc. of ACM SIGSAC*, 2016, pp. 908–919.

[41] W. Choi, M. Seo, and D. H. Lee, "Sound-proximity: 2-factor authentication against relay attack on passive keyless entry and start system," *Journal of Advanced Transportation*, vol. 2018, pp. 1–13, 2018.

[42] M. Miettinen, N. Asokan, F. Koushanfar, T. D. Nguyen, J. Rios, A.-R. Sadeghi, M. Sobhani, and S. Yellapantula, "I know where you are: Proofs of presence resilient to malicious provers," in *Proc. of ACM CCS*, 2015, pp. 567–577.

[43] M. Conti and C. Lal, "A survey on context-based co-presence detection techniques," *arXiv preprint arXiv:1808.03320*, 2018.

[44] M. Hessar, V. Iyer, and S. Gollakota, "Enabling on-body transmissions with commodity devices: poster," in *Proc. of International Conference on Mobile Computing and Networking*, 2016, pp. 452–453.

[45] Z. Li, H. Wang, and H. Fang, "Group-based cooperation on symmetric key generation for wireless body area networks," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1955–1963, 2017.

[46] T. J. Pierson, T. Peters, R. Peterson, and D. Kotz, "Proximity detection with single-antenna IoT devices," in *Proc. of Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–15.

[47] Z. Luo, W. Wang, J. Qu, T. Jiang, and Q. Zhang, "ShieldScatter: Improving iot security with backscatter assistance," in *Proc. of ACM Conference on Embedded Networked Sensor Systems*, 2018, pp. 185–198.

[48] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," in *Proc. of CRYPTO*, 2000, pp. 531–545.

[49] N. F. Tanenbaum, Andrew S and D. J. Wetherall, *Computer Networks, sixth Edition*. Pearson, 2021.

[50] Z. Xu, J. Li, Y. Pan, L. Lazos, M. Li, and N. Ghose, "PoF: Proof-of-following for vehicle platoons," in *Proc. of NDSS Symposium*, 2022, pp. 1–18.

[51] H.-A. Wen, T.-F. Lee, and T. Hwang, "Provably secure three-party password-based authenticated key exchange protocol using weil pairing," *IEEE Proceedings-Communications*, vol. 152, no. 2, pp. 138–143, 2005.

[52] M. Vanhoef and E. Ronen, "Dragonblood: Analyzing the dragonfly handshake of WPA3 and EAP-pwd," in *Proc. of IEEE S&P*. IEEE, 2020, pp. 517–533.

[53] N. Ghose, L. Lazos, and M. Li, "SFIRE: Secret-free in-band trust establishment for COTS wireless devices," in *Proc. of IEEE INFOCOM*,

2018, pp. 1529–1537.
[54] B. Blanchet, B. Smyth, V. Cheval, and M. Sylvestre, "ProVerif: Cryptographic protocol verifier in the formal model," http://prosecco.gforge.inria.fr/personal/bblanche/proverif/, 2018.
[55] D. Freed, J. Palmer, D. Minchala, K. Levy, T. Ristenpart, and N. Dell, ""a stalker's paradise": How intimate partner abusers exploit technology," in *Proc. of CHI*, 2018, p. 667.
[56] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan, "Sensor-based proximity detection in the face of active adversaries," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 444–457, 2018.
[57] M. Rahman, "Energy efficiency evaluation of BLE 5 technology," *Master's Thesis*, p. 65, 2019.

Nirnimesh Ghose received his Ph.D. in the Electrical and Computer Engineering from the University of Arizona, Tucson in 2019. He is an Assistant Professor of Computer Science and Engineering at the University of Nebraska-Lincoln. He received his MS degree in Electrical and Computer Engineering from the Illinois Institute of Technology, Chicago in 2012, and his B.Tech. degree in Electronics and Communication Engineering from the Uttar Pradesh Technical University (now Dr. A.P.J. Abdul Kalam Technical University), Lucknow, India in 2010. His research focuses on network security and privacy with applications to emerging wireless networks, cyber-physical systems, the Internet of Things, aviation and transportation networks, and the interaction between cybersecurity and social networks. He has authored papers in flagship security conferences and journals like IEEE Security and Privacy, IEEE INFOCOM, USENIX Security, and IEEE Transactions on Mobile Communications. He has served as a web chair for IEEE CNS 2018 and as a reviewer for numerous conferences and journals.

Kaustubh Gupta received the M.S. and B.S. degrees in Computer Science from the University of Nebraska, Lincoln, in the years 2022 and 2020, respectively. He is currently working as a Security Engineer at a leading cloud computing company as his interests continue to lie in the domains of cybersecurity and cloud computing. During his academic tenure, he was actively involved in various research projects with a particular focus on cybersecurity.

Loukas Lazos received the Ph.D. degree in Electrical Engineering from the University of Washington in 2006. He is a Professor of Electrical and Computer Engineering at the University of Arizona. His research interests lie broadly in the areas of wireless network security, user privacy, and communications, with an emphasis on secure protocol design, trust management for emerging technologies, security of cyber-physical systems, and fair resource allocation. Recently, he has focused on projects related to the security of connected autonomous systems, the security of NextG Networks and mmWave communications, trust establishment for IoT, dynamic and fair spectrum access, and secure cloud storage. He was a recipient of the U.S. National Science Foundation Faculty Early CAREER Development Award in 2009 for his research in the security of multi-channel wireless networks. He has served as the Technical Program Chair for the IEEE CNS Conference, the IEEE GLOBECOM Symposium on Communications and Information Systems Security, and the IEEE DySPAN Workshop. He is a Senior Associate Editor of the IEEE Transactions on Information and Forensics Security.

Ming Li (M'11, SM'17) is an Associate Professor in the ECE Department at University of Arizona. He was an Assistant Professor in the Computer Science Department at Utah State University from 2011 to 2015. He received his Ph.D. in ECE from Worcester Polytechnic Institute, MA, in 2011. His research interests include wireless networks and security, privacy-enhancing technologies, and cyber-physical system security. He has published more than 135 journal and conference papers, with an h-index of 42. He received the NSF CAREER Award in 2014, the ONR YIP Award in 2016, and several paper awards, including the best paper award from ACM WiSec 2020. He served on the editorial boards of IEEE TMC and TDSC. He was a TPC Co-chair of IEEE CNS 2022. He is a senior member of IEEE and a member of ACM.

Ziqi Xu earned her MS degree in Electrical Engineering from Syracuse University in 2019, distinguished by the receipt of the Outstanding Achievement Award in Graduate Study. She is currently a Ph.D. candidate in electrical and computer engineering at the University of Arizona, Tucson. Her research focus centers on the domain of cyber-physical system security.

Jingcheng Li received the B.S. degree from the Beijing University of Posts and Telecommunications, China, in 2017, and the M.S. degree from Syracuse University in 2019. He is currently pursuing a Ph.D. degree with the Department of Electrical and Computer Engineering at the University of Arizona. His research interests include wireless physical layer security and millimeter-wave communication.