UWB-Auth: A UWB-based Two Factor Authentication Platform

Yifeng Cao
ycao361@gatech.edu
Georgia Institute of Technology
USA

Ashutosh Dhekne
dhekne@gatech.edu
Georgia Institute of Technology
USA

Mostafa Ammar
ammar@cc.gatech.edu
Georgia Institute of Technology
USA

ABSTRACT

This paper presents an ultra-wideband (UWB) based two-factor authentication (2FA) platform, called UWB-Auth, designed as carriable or wearable devices. UWB-Auth eliminates various social engineering attacks, including phishing attack, 2FA-fatigue attack, co-located attack etc., on existing 2FA solutions like Duo and reveals simple and fast user interaction. The key innovation of UWB-Auth is a novel combination of location authentication via UWB, checking whether a legitimate token is in the vicinity of the login device with centimeter-level accuracy, followed by an abstraction layer allowing different knowledge-based or biometric-based authentication, ensuring the user's identity and intent to login. Moreover, UWB-Auth reverses the sequence in which the two factors are verified, providing robust defences against data breach. We develop 3 UWB-Auth prototypes: a key-chain token, a smartwatch with commercial knowledge/biometric factor, and a smartring with customized knowledge/biometric authentication algorithm to demonstrate the effectiveness of UWB-Auth. Overall, UWB-Auth completes the whole authentication process in 4 seconds, and completely rejects malicious requests when the token is 20cm or 10° outside a small valid physical area near the login device. Even when a malicious entity gains physical access to the token, UWB-Auth stops attack attempts via knowledge and biometric authentication.

CCS CONCEPTS

• **Security and privacy** → **Multi-factor authentication**; • **Networks** → *Mobile networks*; • **Human-centered computing** → *Ubiquitous and mobile computing*.

KEYWORDS

2FA IoT Token, UWB, Wearables

ACM Reference Format:
Yifeng Cao, Ashutosh Dhekne, and Mostafa Ammar. 2024. UWB-Auth: A UWB-based Two Factor Authentication Platform. In *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '24), May 27–30, 2024, Seoul, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3643833.3656113

1 INTRODUCTION

Today, many online accounts are secured by using not just passwords (called the know-factor since it is knowledge-based) but also a second factor (called the have-factor) that requires the user to prove access to a device such as a smartphone or a physical token. While two-factor authentication (2FA) is increasingly becoming common practice, social engineering attacks, such as phishing, have exposed some of the limitations of today's 2FA systems. One solution to counter phishing attacks is to perform the second authentication factor locally between the login device and the token, rather than allowing the second-factor device to have its own independent communication path to the server through the Internet. Local Proximity-based solutions have used either NFC [20] or Bluetooth [50], and these solutions avoid the problems faced by Duo [6] and similar mechanisms that allow direct token-server communication. However, NFC requires the user to bring a token *very close* to the NFC reader and as such increases user interaction. Bluetooth based proximity checks can be quite erroneous and signal strength can be artificially boosted to fool the system into thinking that the authenticator device is close-by when it is actually *too far*.

When searching for a solution that will solve this NFC-Bluetooth too-close-too-far Goldilocks problem, we discovered that ultra-wideband (UWB) radios, that are now becoming popular for precise localization and object finding, could provide the "just-right" solution. UWB would allow us to define a small angular area in front of the login device, with tuneable size and angle, to be treated as the authentication zone. The UWB-enabled token must be present within this authentication zone for a login attempt to succeed.

In developing a UWB-based localization authentication solution, we discovered a *fundamental new opportunity that can keep passwords secure even during phishing attempts* that is available to any proximity based 2FA system.

The fundamental reason why passwords can easily be garnered during a phishing attack is because the second factor that verifies the possession of a token or authenticator device occurs *after* the credentials have been provided by the user. This order of verification is necessary on traditional systems since otherwise an adversary could trigger a large number of second-factor notifications creating a nuisance for users, and draining organizational resources. However, when the second factor is locally verified, the server can first verify the proximity of the token to the login device over the same secure connection that the login device established for login, and only then ask for password. UWB scans issued by the attacker's login device are not over the Internet, but locally in the vicinity of the login device. Asking for passwords after proximity check eliminates the possibility of any information leaking as a result of phishing attacks. Indeed, flipping the order of the two authentication factors is available to any system that relies on local verification of proximity. To the best of our knowledge, no other 2FA system provides this functionality of keeping passwords safe during a phishing attack. We combine our own UWB proximity solution with the flipped order of verification described above in a system we call UWB-Auth.

Our contributions in UWB-Auth are two-fold:

WiSec '24, May 27–30, 2024, Seoul, Republic of Korea

ACM ISBN 979-8-4007-0582-3/24/05.
https://doi.org/10.1145/3643833.3656113

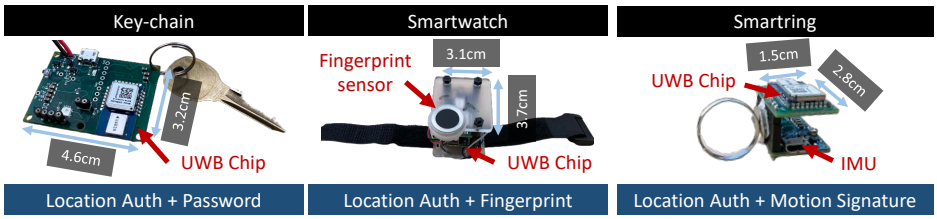

Figure 1: UWB-Auth prototypes in different forms. Location of the token is verified as the *first factor*, followed by several options for the second factor, differentiating our three prototypes, and demonstrating the UWB-Auth platform capabilities.

• We develop a generic platform enabling accurate UWB distance and angle measurements between a carriable or wearable authentication token and the login device.

• We design a two-factor authentication protocol leveraging accurate UWB localization for the have-factor executed via the login device as a conduit. By first verifying the have-factor, our protocol is resilient to several social engineering attacks.

As a demonstration of the wide range of use-cases where UWB-Auth can be applicable, we have developed three separate UWB-Auth prototypes in three different form-factors: a key-chain, a smartwatch, and a smart ring, all shown in Fig. 1. While the have-factor is demonstrated via UWB, the know-factor varies between these prototypes. The key-chain allows the user to continue using traditional passwords by typing them into the browser, albeit after the have-factor has been verified. The smartwatch holds the user's passwords protected by the user's fingerprints, making it simple to access online accounts using fingerprint and strong random passwords. Finally, the smart ring uses specific motion, such as the user's signature drawn while wearing the ring, to authenticate the user. The underlying protocol includes a handshake between the login device and the server, then a location authentication step, and then an optional password-from-token step. While we describe the rest of this paper assuming an inverted order of the two factors, the traditional order of the two factors is also available to UWB-Auth.

UWB-Auth mitigates: (a) Phishing attacks [10, 29], (b) phantom-login attacks or 2FA fatigue attacks [5, 11], (c) lost-device attacks, (d) co-located attacks, and (e) shoulder-surfing attacks [39]. A detailed security analysis is covered in Section 5. To the best of our knowledge, no other authentication solution is robust to all of these potential attacks simultaneously while maintaining simple user interaction. A quick comparison is enumerated in Table 4. Finally, while UWB chips are required to reduce the attack-surface, considering that UWB is being widely embedded in IoT devices [3], we expect this requirement to be easily satisfied in the future.

We evaluate UWB-Auth with real-world prototypes demonstrating strong security when UWB-based location authentication is used jointly with traditional passwords or with a fingerprint reader or motion sensor. Phishing will not be possible when the token is more than 20cm away or 10° away in angle from the valid sector region in all prototypes. We also evaluate the effectiveness of our model of verifying motion signatures through a small-scale user study and observe that it can achieve a low false positive rate (adversary gaining access to the online account, *after* obtaining the token) of 0.1% while maintaining false negative rate (a legitimate user is unable to access at the first attempt) as low as 4.8%. All methods have an end-to-end authentication time of less than 5 seconds (including the time for user interaction, e.g., password typing, fingerprint authentication, etc.), in which only half second authentication time results from location authentication.

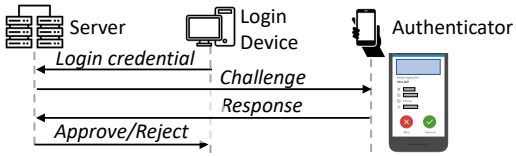

Figure 2: Typical 2FA protocol followed by Duo [6].

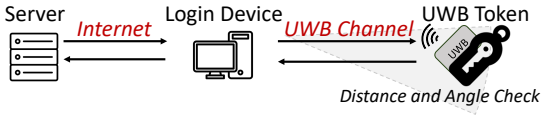

Figure 3: In contrast to Fig. 2, in UWB-Auth, the web server interacts with the token via the login device, and not directly over the Internet.

2 UWB-AUTH OVERVIEW

We now first introduce the scheme of traditional 2FA solutions. Then we present a high-level overview of UWB-Auth, with the interactions between the various entities presented in Fig. 3.

2.1 Background on Traditional 2FA

Existing 2FA solutions verify the possession of a token via a challenge-response protocol. For instance, after credential verification, Duo [6] will send a challenge to the user's registered smartphone to request the user's approval, affirming the legitimacy of the login attempt. This work-flow is shown in Fig. 2. However, this mechanism is vulnerable to phishing attacks as follows. An adversary creates a malicious website A' which resembles in appearance to the authentic website, tricking the victim in to submitting their credentials (username and password). The adversary then uses the credentials to log into the authentic website. Indeed the authentic website would trigger a second factor approval process, however, the victim will happily approve this illegitimate login request since the victim believes that it is their own request that is triggering this notification. Apart from phishing, phantom-login [36] and 2FA-fatigue attack [4], where victims are tricked into approving the login request from the adversary out of habit, are also feasible attacks on existing 2FA solutions. Such attacks have successfully fooled even IT professionals in the tech industry (Cisco [15], Uber [23]).

2.2 Physical Components in UWB-Auth

Similar to existing 2FA solutions, UWB-Auth involves message exchange between three main components: (a) a UWB token, (b) the login device, and (c) the web server to which the user wishes to

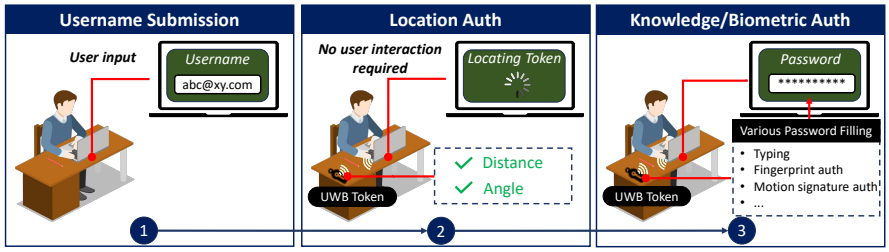

Figure 4: The user interactions in UWB-Auth during regular login.

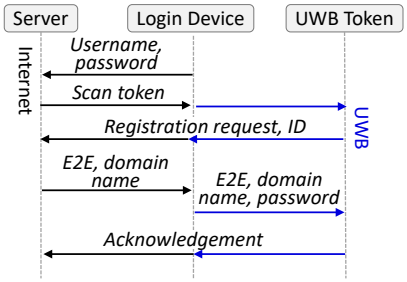

Figure 5: Registration protocol details.

login. However, as depicted in Fig. 3, the communication between the web server and the token is facilitated via the login device acting as a conduit, with UWB facilitating precise localization.

UWB token: The token proves to the website that it is within a small circular sector near the login device using UWB ranging and signal angle of arrival (AoA). This token can take any form-factor, e.g., a smartphone, smartwatch, a key-chain, or an AirTag-like carry-able hardware [2], so long as they are equipped with a UWB chip. In addition to demonstrating have-factor, the token may also be used for proof of know-factor or biometric-factor, depending on the needs of the application. Note that the token is not connected to the Internet, and rather communicates only over UWB using secure communication with the login device.

Login device: The login device functions as an intermediary between the website and the token. It performs UWB ranging and angle measurements with the token, and also serves as a conduit for data transfer between the server and the token.

Web server: The web-sever maintains a per-user shared secret (established during account setup) which it uses to encrypt messages intended for the UWB token, so that while all messages pass through the login device, they cannot be read or tampered with, by the login device. The server validates the trustworthiness of the distance measurement software on the login device and only then, proceeds with token based authentication.

2.3 Authentication Procedures

Account Signup. We assume that account signup or registration, where a user opens a new account at a website, occurs from a secure location with no eavesdroppers or malicious actors. This requirement exists for almost all authentication systems. The user selects a username and a password which will be stored in the server's database, as usual. Then the user registers the token with the website by requesting the login device to scan for nearby tokens. Henceforth, the token will be used as a have-factor, for all logins.

Additionally, to support different knowledge/biometric authentication methods, UWB-Auth allows the token to securely store the user's password, if the user plans to use the token as a password manager, like OnlyKey [13]. The stored password may be retrieved from the token only using the user's predefined unlocking method, such as a valid fingerprint or a motion signature. This unlocking method actually functions as the knowledge/biometric authentication factor as an alternative to remembering passwords. If the user opts in, the user also registers a valid fingerprint, or draws a motion pattern at the token.

Account Login. Regular account logins in UWB-Auth, depicted in Fig. 4, are performed with a marked difference from traditional methods: after the user supplies the username (1), the website first verifies proximity of the login device and the user's token (the have-factor) (2). After successful verification of the have-factor, the website requests the password, which can be either supplied by typing the password, or by the token using biometric or motion pattern-based authentication, actively performed by the user (3).

3 UWB-AUTH DETAILS

We now delve deeper in to the technical details allowing UWB-Auth to function. UWB-Auth securely exchanges messages between the web server, the login device, and the token. A long-term secure key, *E2E*, is established between the web server and the token at the registration stage, and a new per-session secure key, *SK*, to secure UWB communication between the login device and the token, is provided by the server during every login attempt. All communication between server and the login-device is secured using TLS, as usual. We describe next the protocols that governs UWB-Auth, including location verification and credential exchange.

3.1 Account signup (one-time)

The process of account sign up includes registering the token with a new website (creating a shared secret, *E2E*, and associating it with the token's ID). Fig. 5 depicts the message exchange in the registration protocol. When the user creates a username with a password on a new website, the website requests the login device to scan for a token nearby (over UWB). The correct token in the valid region responds to this scan request with its ID. The server creates a shared secret, called *E2E*, for this token. Then, the server sends the *E2E* key to the token together with its website's domain name, through the login device acting as an intermediary. We assume a secure end-to-end channel during account signup: (1) a secure HTTPS connection between the login device and the web server, and (2) confidence that there are no malicious UWB eavesdroppers in the vicinity of the login device at this stage.

3.2 Logins: Website-Token Handshake

The authentication starts with the user entering the username on a website loaded over a secure HTTPS connection. Once the server receives the login request, the handshake phase commences. The main function of this phase is to convince the token that it is communicating with a registered website, and convince the website that it is communicating with a live token. The server uses its long term shared secret *E2E* to send data to the token via the login device, and also generates a session key *SK* for securing communication between the login device and the token. This interaction is detailed in the first part of the Fig. 6.

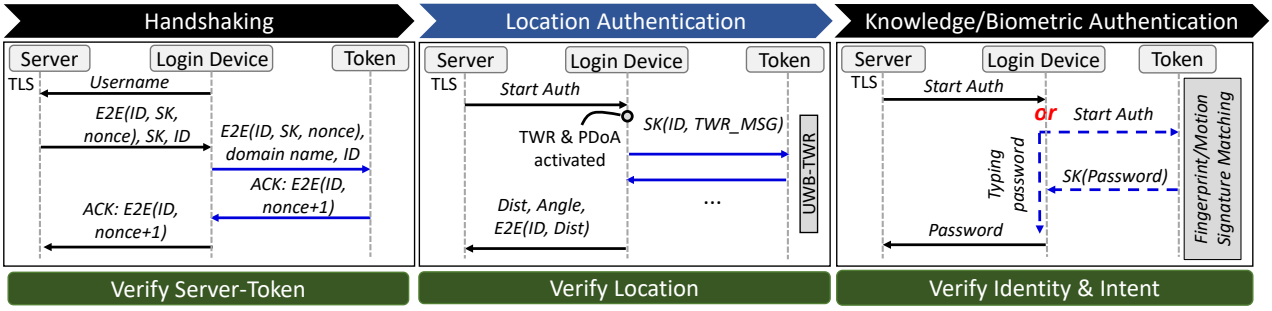

Figure 6: UWB-Auth authentication protocol verifies the legitimacy of the token and the server, verifies that the token is located physically nearby, and verifies that the user can prove their identity. UWB-Auth thwarts several 2FA attacks.

The server sends SK and the token's ID to the login device along with an encrypted message for the token: $E2E(SK, ID, nonce)$. The login device initiates UWB communication with the nearby token identified by ID, and sends the server's encrypted message to the token, along with the domain name of the website it is connected to[1]. The token looks up the correct $E2E$ secret based on the domain name, decrypts the message to verify the server, since only the correct server would be able to correctly encode the token's ID in the encrypted message, obtains the SK, and sends an SK encrypted acknowledgement which includes an $E2E$ encrypted incremented nonce. When the server receives this $E2E$ encrypted acknowledgement, it knows that it is communicating with the authentic token since only the authentic live token would be able to decrypt the nonce and then increment it. All subsequent communication between the login device and the token is encrypted by SK.

Note that since the domain name is included by the login device, remote phishing attacks are caught by the token since the website name it obtains does not match the registered domain. Malicious tokens cannot intercept the UWB messages and generate their own replies since they will not be able to decrypt and increment the $E2E$ encrypted nonce supplied by the server.

3.3 Logins: Location Authentication

Once the handshake phase is complete, the server requests the login device to start location authentication phase. Localization itself is performed using standard ranging and angle measurement methods. However, all UWB messages are encrypted using SK connecting this phase with the handshake phase. Description of the ranging and angle measurement protocols is included here for completeness. UWB-Auth employs asymmetric double-sided two way ranging (TWR) as specified in the IEEE 802.15.4z standard [21] for measuring distances, and a phase-difference-of-arrival (PDoA) technique to measure the relative angles of the token to the login device. The messages exchanged in TWR and PDoA are depicted in Fig. 7. UWB-Auth performs both these operations using the same TWR messages (since PDoA is a passive operation), and the message exchange is encrypted using the session key SK to prevent eavesdropping.

Underlying TWR Protocol. The TWR protocol for ranging comprises a POLL message initiated by the login, followed by a RESP message from the token and finally a FINAL message being sent by the login device. Each message pair, POLL-RESP and RESP-FINAL is seen as two rounds of message exchanges (with the RESP being common between the two), with corresponding round trip delays denoted as R_A and R_B and turn-around delays denoted as D_A, and D_B. The distance between the login device and the token is calculated using the standard formulation shown at the bottom of Fig. 7(a) derived in [40], which eliminates clock offset and drift.

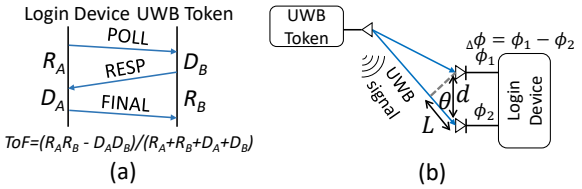

Figure 7: (a) Two way ranging protocol for UWB devices; (b) Angle of arrival measurement with phase difference.

Underlying PDoA Protocol. The login device, equipped with an antenna array, measures the angle of arrival (AoA) of the messages from the token using phase difference of arrival (PDoA). The logic behind PDoA is shown in Fig. 7 (b): the token is sending UWB signals to the login device equipped with two antennas placed d distance apart. This signal will be received by both antennas, but with slightly different path-lengths (differing by L). This path difference manifests as a phase difference $\Delta\phi$ between the received signals at the login device. When the token is a few wavelengths away from the login device, the angle of arrival θ, air-path difference L, and the antenna space d conforms to geometric constrain $L = d\cos(\theta)$. Replacing L with phase difference θ, the angle of arrival is calculated by $\cos\theta = \frac{\Delta\phi\lambda}{2\pi d}$, where λ is the wavelength of the UWB signal.

The angle and distance measurements provide the location of the token. Post location authentication phase, the token sends an $E2E$-encrypted message, including the distance measurement, to the login device. The login device packs the observed distance, the observed angle, and the token's $E2E$-encrypted message, and sends to the server. The server will check: (i) if the token is legitimate by decrypting the token ID and incremented nonce, and (ii) if the token is physically located within the valid area in the vicinity of the login device based on the distance and angles reported by the token and the login device. These checks prevent co-located phishing and phantom-login attacks.

Defining valid authentication region. Using TWR and PDoA, UWB-Auth measures the angle and distance of the token. We define a valid region around the login device in which the server will

[1]Practically, techniques explored in [50] can be used for identifying the correct domain, however, we only focus in the higher-level idea here.

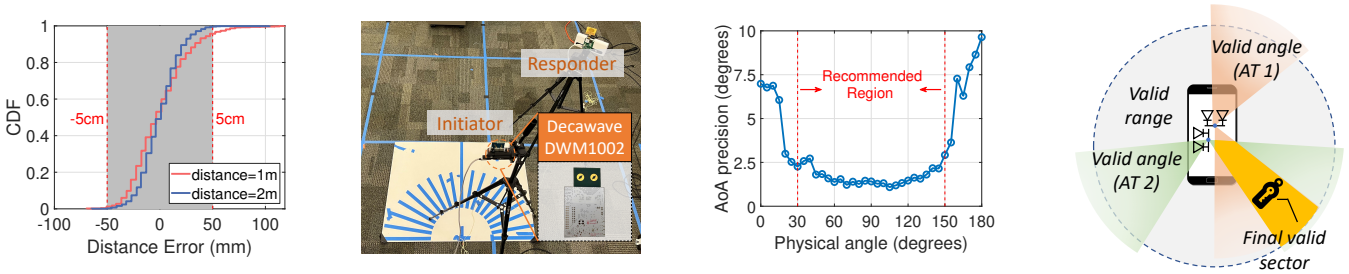

Figure 8: (a) Stability of distance measurements with UWB; (b) PDoA testbed; (c) Measured AoA precision at different angles; (d) Valid sector (yellow with token) defined by two overlapping valid angles from orthogonal PDoA setups.

approve the login request. The capabilities and limitations of the UWB hardware will guide the size and shape of this region.

Figure 8(a) shows the stability of the token's distance measurement (mean subtracted values) when the token is placed about 1 m and around 2 m from the login device. Observe that the central 95^{th} percentile precision is $\pm 5\,cm$ in both cases providing us with an approximate margin of error in the distance measurements. We use an upper bound of 1 m in our evaluations, with an expectation that the token will actually be within 2 feet (60 cm) of the login device, based on ergonomic recommendations for laptop usage [7]. The actual authentication distance desired by the user and the website can be a tunable parameter.

Angle measurement precision is more complex due to the non-linear mapping between the measured phase difference and the physical angle ($\cos\theta = \frac{\Delta\phi\lambda}{2\pi d}$). To study the real-world effects of this non-linear mapping, we placed the token at 37 different locations along a circle 1 m in radius spanning $[0°, 180°]$ angles. Fig. 8 (b) shows our marked testbed with the test locations $10°$ apart, and small markings every $5°$. The results of the angle measurement obtained by averaging 500 readings at each location show varying precision as the physical angle changes. These results are captured in Fig. 8 (c). We observe improved AoA precision as θ approaches $90°$, leading to our recommended setting: θ within $[30°, 150°]$.

The 1 m distance measurement and the $[30°, 150°]$ angle defines the valid region for authentication. To remove front-back angle ambiguity we recommend two orthogonal PDoA devices defining a small sector of valid authentication region (see Fig. 8 (d)).

3.4 Logins: Knowledge/Biometric Auth.

While a legitimate token's close proximity to the login device has been verified by the handshaking and location authentication phase, two important vulnerabilities still remain: (a) the user's intent to login is not verified, and (b) whether or not the token is with the legitimate user is not verified. As a consequence, the user account is vulnerable if the token is lost or stolen. Therefore, a robust second factor verifying the know-factor or biometric-factor is essential (to prove legitimacy of the user), coupled with some action voluntarily taken by the user (to indicate user intent). UWB-Auth designs a *general communication protocol* between the server, the login device, and the token, which supports diverse know-factor and biometric-factor authentication approaches. After the location authentication passes, the server expects to be presented with the password. Now, the user can either manually type in the password, or unlock the password stored in the token, treating the token as a hardware password manager. For manual password entry done on the laptop, the token is not involved in any further communication. If the user chooses fingerprint matching or motion signature matching, the login device directs the token to start this authentication. The token indicates to the user that it is ready to accept an "unlocking" input. In our implementation the user must use either a legal fingerprint, or move the token in the registered pattern. Once the fingerprint or the motion signature is verified, the token sends the SK-encrypted password to the login device. The login device will fill in the password input field automatically and then submit the password to the server. In this protocol, **the server benefits from the simplicity of a general interface**: Whether the password is typed-in manually or auto-filled using the token is transparent to the webserver. Note that using the token to store the password, it is feasible to use longer and randomized passwords strengthening the overall system.

4 PROTOTYPE IMPLEMENTATIONS

Our implementation of UWB-Auth includes hardware and software implemented at the login device and various token form-factors, as shown in Fig. 9.

4.1 Web server and Login Device

We emulate a login system with an authentication process running at the server, and a login page built with Flask framework at the client. Since laptops do not yet have embedded UWB, a UWB module is connected to the login device via USB to emulate a UWB-embedded login device. We use an HP ENVY, Intel Core i7-10510U laptop with 16GB RAM as the login device. We supplement the login device with two standalone UWB setups (mimicking a range-only and another range-and-angle setup) connected to the laptop over USB. In the first setup, we use one Trek1000 transceiver as the UWB communication module at the login device. Trek1000 is a one-antenna transceiver which can perform distance measurement using TWR with nearby UWB devices, but cannot perform AoA measurements. This setup is similar to most existing COTS mobile devices with UWB chips such as iPhone [12], and Samsung Galaxy [18]. In the second setup, we use two DWM1002 double-antenna devices as the UWB communication module which can simultaneously perform TWR and measure the AoA of a nearby UWB device. To create an unambiguous feasible area sector, the login device should be equipped with two vertically aligned antenna arrays. Currently, COTS mobile phones cannot perform sectoral detection, however, this space is rapidly improving. A strong need for antenna arrays, probably motivated by UWB-Auth, may propel manufacturers in that direction.

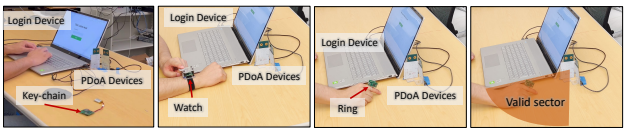

Figure 9: Implementation of UWB-Auth using the custom key-chain, watch and ring as tokens, and 2-PDoA UWB devices connected to a laptop enabling authentication only within the valid sector.

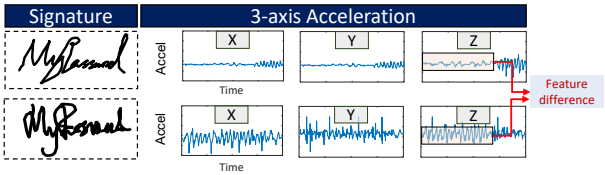

Figure 10: Two volunteers stylize the word "MyPassword". (Left) visualized movements; (Right) accelerometer readings.

4.2 UWB Tokens

We prototype tokens of three different form factors: (i) A UWB-chip embedded token with no other sensors (a key-chain form-factor inspired by AirTags [2]); (ii) A smartwatch with a UWB chip and a fingerprint sensor (form-factor inspired by PayPal-Samsung integration [43]); and, (iii) A smart ring (form factor inspired by [14] and [56]) with a UWB chip and an inertial sensor for authentication via motion signatures. These form-factors demonstrate the possibility of developing UWB-Auth on various COTS devices, while sharing the same authentication pipeline as specified in Sections 2 and 3. All three tokens use the DWM1000 UWB module [16] controlled by an ARM Cortex M0 ATSAMD21G18.

Key-chain token: We have designed a UWB token (3.2cm × 4.6cm) with a DWM1000 module and Cortex M0 chip. In this prototype, we expect the token will only be used for location authentication and the user will demonstrate the know-factor by manually typing the password on the website. Since UWB chips have been widely deployed in UWB location tags [2] and modern smartphones [12, 18], no extra hardware would be required to implement a UWB-only token today. While the simple key-chain token only provides location authentication, it is still resistant to phishing attacks since the location of the token is verified first and the token will detect a domain name mismatch *before* the system proceeds to prompting for the user's password.

Smartwatch with a fingerprint sensor: We designed a UWB-Auth token in a smartwatch form-factor (A stack of two 3.7cm×3.1cm PCBs) with a fingerprint sensor [1] in addition to the UWB. The fingerprint sensor is also controlled via the Cortex M0 microcontroller. The user authorizes access to the password by touching their finger to the fingerprint sensor. The fingerprint is stored and matched locally on the smartwatch, which unlocks a vault that contains the password. Once they pass the fingerprint authentication, the token sends the encrypted password to the login device which automatically fills in the password.

Smartring with an IMU: The smartring is an emerging smart device with appealing feature in fitness and vital sign tracking [14]. As a wearable device that is likely to be always carried by people, we find it a suitable form-factor for enabling 2FA. We have developed a ring form-factor hardware (a stack of two 2.8cm×1.5cm PCBs,

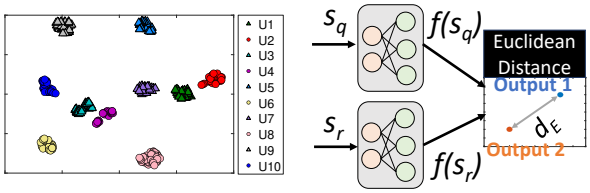

Figure 11: (Left) Feature visualization using t-SNE [51] of signatures written by 10 different users; (b) (Right) The forward process of Siamese network to compute signature similarity.

attached to the top of a ring) to demonstrate UWB-Auth's feasibility on this futuristic form factor. The PCB houses an LSM6DSO 6-DoF inertial measurement unit (IMU), in addition to the UWB and Cortex M0 microcontroller. The IMU collects movement data at 70Hz. The small form factor makes it challenging to place traditional biometric sensors such as fingerprint readers, in addition to the required UWB radio. Instead, we design a motion signature based authentication, which only relies on an inertial sensor commonly present in smartrings, to complete the knowledge/biometric authentication phase. We expect the user will "draw" a pattern while wearing the ring on their finger for authentication. If the movement pattern as seen by IMU matches the registered pattern, the password is provided to the login device to auto-fill.

The motivation of using motion as signatures comes from the fact that physical signatures have been long used for authentication on bank checks and contracts. An extension of physical signatures is motion signatures where an instrument (a pen or a ring) is moved in a certain predefined fashion and these movements are captured and compared with those stored in a database for authentication. Motion signatures can be changed at will, in contrast to fingerprints, and they capture an aspect of knowledge (in terms of what the pattern looks like) as well as biometrics (in terms of how the pattern is drawn with pauses and varying speeds). The user will perform certain hand or finger movements using the ring, when registering at a website, and then repeat these movements when logging in. The inertial sensor on the ring captures the movements and compares it with the pattern stored on the ring. If matched, the password is extracted and sent to the login device for authentication.

The IMU captures spatio-temporal information about the movement including subtle pauses and quick movements of the hand, providing a rich set of features for motion signature matching. Fig. 10 shows how the same word, "MyPassword", drawn by two volunteers wearing the smartring, differs both visually, and as observed from the IMU sensor data. As a preliminary test to understand the diversity of motion patterns, we asked 10 users to draw their own signatures while wearing the smartring. Fig. 11 (left) visualizes the separation between personalized signatures of 10 different users converted to 2D space using t-SNE [51]. Signatures from different users are clearly separable, while different instances of the same user's signatures are closely clustered, demonstrating the feasibility of motion signature authentication. We adopt siamese neural network (SNN) [24], an architecture that takes two input vectors and produces a comparison output vector as shown in Fig. 11 (right). The authentication passes only when there is a small Euclidean distance between the feature vector of the input motion data and the registered motion signature.

5 SECURITY ANALYSIS

UWB-Auth achieves secure login through (i) location authentication, (ii) knowledge/biometric authentication, and (iii) through switching the sequence of the two, in contrast with traditional 2FA. We now discuss the resilience of UWB-Auth to some common attacks; Table 4, under introduction, presents a comparison with other 2FA schemes.

Remote phishing. In a phishing attack, the attacker entices a user to open a malicious website S' that looks similar to S. Using traditional 2FA, when such an attack is launched, if the user accesses the malicious website S', they volunteer the password to the website, assuming it is the legitimate website S and even provide their consent to the second factor. In the background the malicious website S' forwards the password and username to the legitimate website S, which generates the 2FA request that the user consents to. In UWB-Auth such a phishing attack is prevented at multiple levels: (i) For the have-factor to succeed, the token and the login device must be in physical proximity, since the token cannot be directly queried by the website S (token is not connected to the Internet). (ii) The malicious login device cannot send made-up messages as if they originated from the token since it does not know the token's E2E secret key. (iii) If the legitimate token receives a message from the malicious website S', since the user has not registered with S', the token does not have a corresponding $E2E$ entry for that website. Hence no password is exposed. Similar attacks launched remotely such as phantom-login attacks [5], and 2FA-fatigue attacks [11] are also prevented.

Co-located phishing. If a malicious login device D' is in communication range of the token, then D' can attempt to connect with the token directly at the same time when the user is attempting either a phished login or a legitimate login. However, such attacks will be stopped so long as the token is not within the small *valid area* of D'. UWB-Auth cannot protect against *malicious devices located within the valid sector*. However, since the valid sector is typically very small, it is easy for the user to ensure the integrity of the valid area.

Stolen-token attack. Traditionally, a stolen token makes the account vulnerable to the adversary, which is a crucial shortcoming in existing hardware-token based 2FAs like YubiKey [20]. This is particularly true if the account was ever subject to a phishing attack since the password might have been previously compromised without the user's knowledge. UWB-Auth makes it harder for attackers to garner the password since a phishing attack will never compromise the user's passwords. Further, when UWB-Auth uses fingerprint or motion signatures, stronger passwords can be stored at the token, increasing resilience to password guessing.

Malicious login device: timestamp spoofing attack. As elaborated in Section 3, UWB two-way ranging calculates distance based on timestamps reported by both the login device, and the token. If an adversary manipulates the reported timestamp at a malicious login device D', the calculated distance can be intentionally reduced. UWB-Auth prevents timestamp spoofing by assuming that a trustable software runs at the login device to report the UWB distance measurement. Since a trustable software is an essential requirement in all existing 2FA solutions (Duo [6], YubiKey [20], etc.), either in the form of a software, a backup service or a plugin, it is reasonable to assume UWB-Auth can report location through the trusted software, rather than reported by the rendered website or the login device directly.

Malicious token attack. An attacker could create a token which responds to any and all authentication requests. However, merely creating such a token does not suffice because the per-website encryption $E2E$ is not known to the malicious token. Therefore, it cannot perform nonce-based challenge-response authentication with the server.

Man-in-the middle attack. In the classic man-in-the middle attack, an attacker may intercept and manipulate communication between the login device and the token. However, the channel between the login device and the token is encrypted by session key SK, which is generated at the server and communicated to the login device via TLS (over HTTPS connection). When the login device starts handshake with the token, the session key SK is also encrypted by $E2E$, and never sent in cleartext. Since the attacker cannot retrieve session key, UWB-Auth is resilient to man-in-the-middle attack.

6 PROTOTYPE EVALUATION

UWB-Auth relies on location authentication followed by prototype-specific proof of knowledge or biometric information. Our evaluation of UWB-Auth reflects these two-stages. First, we focus on location authentication showing the accuracy we obtain using UWB in different physical environments. Then, we compare UWB based localization with Bluetooth demonstrating the reasons for choosing UWB radios over the more prevalent Bluetooth hardware. We then evaluate, in some detail, our prototype implementation of the ring form-factor, since motion signatures do not yet have a standardized interface, in contrast to fingerprints. We then perform end-to-end evaluations of our three prototypes, comparing the authentication time, and energy consumption.

6.1 Location Authentication Accuracy

We first evaluate the accuracy of UWB-Auth in location authentication. The evaluation includes two parts: distance authentication accuracy and angle measurement accuracy, performed at three locations with different levels of multipath (see photos of the three areas in Fig. 12): (i) An atrium, (ii) a small-size common area, and (iii) a cluttered lab. We desire a sharp cut-off at our set distance and angle thresholds.

Distance authentication accuracy. In this evaluation, we set the radius of the valid sector to be 1 m within which, login is approved. The token is successively moved $10cm$ at a time, from $0.6m$ to $1.4m$. If the measured distance is less than $1m$, the authentication is successful, otherwise, authentication fails. Fig. 13 shows the authentication success rate at different distances. We observe that the authentication success rate quickly drops to 0 in all locations when the distance is larger than $1.2m$, meaning attacks from beyond $1.2m$ will be blocked. The success rate remains over 99% when the distance is smaller than $0.8m$. Considering the user is typically very close to the login device (around $0.5m$), our 2FA is highly accurate in distance authentication, due to high precision of UWB-based TWR scheme. We also observe a ranging bias of about $10cm$ in a cluttered space (e.g., lab area) due to multipath, enlarging the 50% success rate region to $1.1m$.

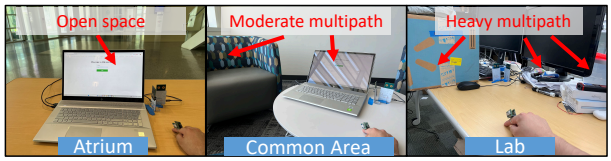

Figure 12: Location authentication evaluation at: (a) Atrium; (b) Building common area; (c) Lab.

Angle authentication accuracy. Angle authentication restricts the feasible region to a smaller circular sector area (instead of a circle with just distance measurements), and provides additional security to prevent attacks from a nearby adversary. We place two double-antenna PDoA devices orthogonal, to uniquely determine the AoA of the token in 2D. The authentication should succeed only when the token is in the valid angle range, set to $[-75°, -15°]$ in our evaluation. Fig. 14 (right) shows the authentication success rate at different angles in three locations. The success rate is approximately 0 when the angle is outside $[-80°, -10°]$, and is over 98% within $[-70°, -20°]$, demonstrating the effectiveness of using UWB PDoA algorithm to perform angle authentication.

6.2 UWB-Auth versus Bluetooth

Bluetooth is used for 2FA in [27, 50] which verifies the proximity of a token to the login device. We compare UWB-Auth with two types of Bluetooth-based 2FA solutions: proximity authentication via Bluetooth connection, and proximity authentication via RSSI. We will only evaluate Bluetooth using insecure RFCOMM sockets, as used in [27], which does not need pairing process.

Bluetooth communication range: The login device establishes a Bluetooth connection with the token as a proof of have-factor. Unfortunately, this approach is vulnerable to co-located attacks. Because the communication range of Bluetooth can be larger than $10m$, the attacker can hide nearby and load a phishing or phantom-login attack.

Threshold based on Bluetooth RSSI: The login device initiates a Bluetooth connection with the token, and uses thresholding on Bluetooth received signal strength indicator (RSSI) to infer the proximity of the token. To compare UWB-Auth's performance versus Bluetooth RSSI based solutions, we setup two Bluetooth devices and collect Bluetooth RSSI data when they are placed at different distances. In our experiment, we use a Google Pixel 6a as the login device, and a Google Pixel 4a as the token. The collected RSSI ranges from -100 to 0 where higher RSSI indicates larger received power. The observed RSSI variation at different distances are shown in Fig. 15. Bluetooth RSSI exhibits large fluctuations even when kept static at the same distance, making it difficult to establish a single threshold based on RSSI. For example, when proximity threshold is set to about $1.5m$, an attacker can easily load a co-located attack from a place which is $3m$ away. Worse still, the attacker can use a Bluetooth repeater/extender to significantly increase the observed RSSI. In contrast, UWB-Auth employs two-way ranging and angle of arrival to compute the exact relative position between the login device and the token, severely limiting co-located attacks.

Overall, UWB is a suitable technology for location authentication, and shows promise as one of the factors in 2FA.

Table 1: Performance of different signature matching models.

Model	BAC (%)	TPR (%)	TNR (%)
CTW [55]	92.4	90.6	94.1
OC-SVM [48]	92.6	89.5	95.7
SVM	94.6	97.9	91.3
Siamese (SNN)	**97.5**	**95.2**	**99.9**

6.3 Motion Signature Evaluation: Smartring

The password and biometric factors enabled by two of our token prototypes have been well-studied. Therefore, we do not evaluate their functioning. However, motion signatures, enabled by our smartring platform are a new approach. Our aim in this evaluation is to only understand the space better and is not to present an exhaustive evaluation of motion signatures, which we leave as an open problem for future research.

As a preliminary study, we recruited 18 volunteers[2] and asked them to design their own motion signatures and repeat multiple times. We collect 1,559 samples in total, covering various signature styles and levels of practice. All the volunteers are right-handed, with the ring worn on their index finger (a more detailed future evaluation should consider left-handed volunteers as well, and those that wear the ring on different fingers).

Understanding the space of motion signatures. When using movements of the finger as a signature, due to inherent variations in signatures drawn by a person we must allow a *margin of error*. However, this margin of error could also allow an attacker's signature to be accepted by UWB-Auth. Therefore, we must tighten bounds, and evaluate UWB-Auth under the following metrics, based on true positives (TP), false negatives (FN), true negatives (TN), false positives (FP), and understand the trade-offs between them:
(1) *True positive rate (TPR)*: $\frac{TP}{TP+FN}$. The probability that a legitimate signature is accepted by the server, which results in a higher *speed of authentication* and a better *user experience*.
(2) *True negative rate (TNR)*: $\frac{TN}{TN+FP}$. TNR measures the probability that a fake signature is rejected by the server (*security* of the signature verification algorithm).
(3) *Balanced accuracy (BAC)*: The accuracy when TPR is equal to TNR. This balance-metric measures the overall performance of signature verification.

As mentioned in Section 4, UWB-Auth extracts features from raw IMU data and employs SNN for signature authentication. To improve authentication robustness, we collect multiple instances of a user's signature during registration. A query signature is assumed legitimate if it matches with 80% of the registered signatures. The evaluation is performed via $k-$fold cross-validation (k is the number of volunteers): we iteratively train the model with $k-1$ users' data, and evaluate on the remaining user.

To understand the model's accuracy in comparison with other models which can also perform pattern matching on motion data, we compare the ring's authentication accuracy using our SNN-based model, using canonical time warping (CTW) [55], support vector machines (SVM), and one-class SVM (OC-SVM) [48]. Table. 1 presents the results in term of overall BAC, TPR, and TNR, of the same k-fold cross-validation set. Overall, SNN achieves the best performance among the models we tested, with 95.2% accuracy in approving a legitimate query and 99.9% accuracy in rejecting

[2]This study has been approved by the IRB at our institution.

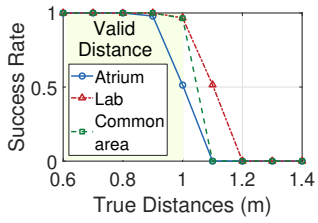

Figure 13: Success rate at different distances for UWB-Auth.

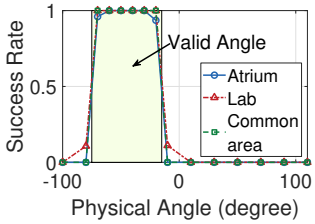

Figure 14: Success rate at different angles for UWB-Auth.

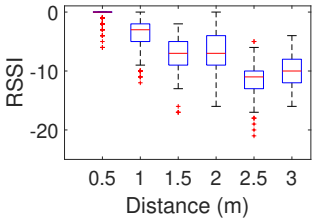

Figure 15: Bluetooth RSSI variation at different distances.

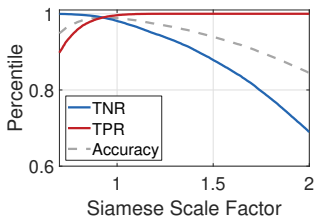

Figure 16: The tradeoff between TPR and TNR.

Table 2: UWB-Auth's end-to-end authentication time.

Prototype	Hand-shaking	Location Auth	Knowledge / Biometric Auth	Full Auth Time
UWB token	90.8ms	306.9ms	4721.1ms	5119.9ms
UWB watch			3552.6ms	3950.3ms
UWB ring			3715.0ms	4112.7ms

Table 3: UWB-Auth power consumption.

Prototype	Idle	AoA and Ranging	Knowledge / Biometric Auth	95% duty cycle
UWB Token	58mW	732mW	0	95mW
UWB Watch			210-350mW	108mW
UWB Ring			126mW	163mW

a fake query. This demonstrates that our SNN-based similarity comparison effectively encodes raw IMU data into a feature space, making signatures comparable with Euclidean distances.

Of course, as we set higher preference on rejecting fake queries, TNR is slightly higher than TPR. Based on the application use-case and user preferences, the weight of TNR and TPR is adjustable by adding a punishing factor α to false negatives. Fig. 16 shows the trade-off between TPR and TNR when different values of *alpha* are used. As α increases from 0.7 to 1.1, TNR/TPR varies from 99.9%/91.9% to 95.2%/99.9%. Observe that TPR and TNR are balanced when $\alpha = 0.91$ rather than 1, which results from differences between the training and test datasets.

Overall, motion signatures offer unique advantages and we believe our work will propel further research in this space.

6.4 System Property: Authentication Time

The average authentication time is a quantitative metric evaluating the usability of an authentication approach. We measure UWB-Auth's authentication time by including the time spent in three phases: handshake phase, location authentication phase, and knowledge/biometric authentication phase. We recruit volunteers to perform authentication with different prototypes and record the average time spent in each phase. For UWB token prototype, we set the required password strength of "3class12" [41], where the password should include 3 different classes of characters with at least 12 characters. For the ring prototype, the volunteers are asked to wear the ring to write (draw) a string with at least 8 characters. Table 2 shows the login delay of each prototype, as seen in our end-to-end implementation. We observe that time spent in handshake and location authentication is at sub-second level, which is almost unperceivable to the user. The major portion of time consumption is the user interaction in the knowledge/biometric authentication phase. Compared to password typing, providing a fingerprint by touching the watch, or drawing a pattern with the ring is more time efficient, but suffers from the overhead of wearing the watch or ring. Overall, the login delay with all our prototypes ranges from 4-5 seconds, comparable to performing one-factor password-only authentication on a laptop/PC. Also, when compared with typing a password on a smartphone, which frequently takes more than 10 seconds [41], our watch/ring prototype is significantly faster.

6.5 System Property: Power Consumption

Authentication is a user-initiated process. Therefore, it is possible to incorporate a miniature physical button to turn the token on and off, which would allow several days of battery life. However, for simplicity, we tested energy consumption of UWB-Auth's prototype tokens in an always-on mode; when idle, when actively running location authentication, and when running knowledge/biometric authentication. The results are shown in Table 3. Note that since UWB-based message transmission is short compared to the whole authentication process, the UWB radio and the fingerprint sensor can be heavily duty-cycled. Table 3 also shows the power consumption with 95% duty cycle is 95mW, 108mW and 163mW for UWB token, watch, and the ring, respectively. Even with the token kept powered on, this power consumption indicates a day-long charge with just a 3.7V/400mAh battery.

7 RELATED WORK

UWB applications. With the capability of measuring distance at centimeter-level accuracy, UWB is appealing in various mobile computing and IoT applications, including indoor localization [26, 30, 32], motion tracking [25], material sensing [31, 54], etc. UWB-Auth demonstrates the feasibility of employing UWB for accurate location authentication with the resilience to co-located attack in a 2FA application, thereby significantly expanding UWB's utility.

Commercial 2FA solutions. The mainstream 2FA solutions are hardware based or software based. While the first factor is still traditional credentials (password), the second factor typically demonstrates the presence of a registered token [6, 9, 17, 19, 20]. YubiKey [20] is a hardware token providing near-field authentication via USB connection or NFC communication with the login device. It complies with FIDO2 standard [8]. However, YubiKey is vulnerable to stolen-token attack when the password was previously compromised using phishing, which is rather common when the credentials are authenticated before YubiKey. Frequent user interaction also limits the usability of YubiKey [28]. Software tokens such as Duo [6], signicat [19], and google 2-step verification [9] require a dynamic code, or a phone call permission, or the user needs to tap a notification option in a registered smartphone, which is resilient to stolen-token attack. However, software tokens are

Table 4: Comparison of 2FA solutions under various attack scenarios. ✓ indicates resilience, ✗ indicates vulnerability.

2FA solutions	Phishing Attack	Credential Garnering	Co-located Attack	Authentication Time	Hardware Used
YubiKey (hardware-based) [20]	✓	✗	✓	∼9.1s [45]	Key chain
Duo (software-based) [6]	✗	✗	N/A	∼11.8s [45]	Smartphone
2FA-PP [50]	✓	✗	✓	∼ 10.3*s* [27]	Smartphone
UWB-Auth	✓	✓	✓	∼3-5s	Multiple options

vulnerable to fatigue attacks [4, 42] and phishing [49]. Compared to existing 2FA solutions, UWB-Auth is resilient to fatigue and phishing attack through precise location authentication, and resists lost-token attack by switching the authentication order of the two factors. Furthermore, UWB-Auth requires no user interaction to authenticate location.

Location-based authentication. Location-based authentication is proposed as a solution to remotely launched phishing attacks. It requires a legitimate token physically close to the login device. Such location authentication can be achieved via GPS [22], Wi-Fi [37, 53], Bluetooth [27, 50], NFC [33, 47], customized RF signals [35, 44], or acoustic signals [34, 38, 46, 52]. The major attack on existing solutions is the co-located attacks when the attacker is physically present at several meters from the victim. UWB-Auth uses UWB for high-precision distance and angle bounding, that resists co-located attacks.

Table 4 summarizes the difference between UWB-Auth and other 2FA solutions. In comparison, UWB-Auth is resilient to phishing attack and stolen-token while maintaining a superior level of usability, with no or lightweight extra burden.

8 DISCUSSION AND CONCLUDING REMARKS

Why use UWB when other wireless technologies exist? Ultra-wideband (UWB) radios have started to become main-stream with their introduction in wearable and carry-able devices such as the Apple Watch [3], and smartphones and finder tags [2]. UWB is preferred over existing wireless technologies including Bluetooth and WiFi for localization-specific applications because of their higher accuracy and resilience to wireless multi-path allowing their use for locating lost objects (with preinstalled UWB tags) even in cluttered real-world environments. In this work, we leverage these properties of precise UWB localization to develop a two-factor authentication system, that overcomes many of the challenges faced by today's two-factor authentication systems, including the vulnerability to phishing attacks, multi-factor fatigue attacks and poor usability resulting from complicated user interaction.

What additional infrastructure does UWB-Auth require? UWB-Auth requires users to carry a UWB-enabled token and use a UWB-enabled login device (such as a laptop or smartphone). The server, being aware of UWB-Auth requires storing an additional encryption key that it shares with the UWB-token, in addition to traditional password and username. Of course, in any 2FA scheme, the server requires to store some additional information, and in that sense, UWB-Auth is not too burdensome.

What additional burden or learning curve do users of UWB-Auth incur? Users need to carry the UWB-token with them at all times. In the future if UWB-Auth becomes integrated into smartwatches, this requirement is automatically met. A subscribing website performs localization authentication with the token automatically after username is entered, and then may provide the user an option to type in a password (key-chain form-factor), or prompt the user to input password directly on the token. The user presents a registered fingerprint on the token (smartwatch form-factor) or moves the token in a predefined fashion (smart ring form-factor) to provide the second factor of authentication to the server.

How does UWB-Auth compare with the state-of-the-art? Some form of IoT tokens have been used to bolster secure access to online accounts using two-factor authentication. This includes commercial products such as RSA tokens [17], physical tokens such as YubiKey [20], software tokens on mobile phones such as Duo[6], and several other token-based systems that have been proposed in academic research [22, 27, 38, 53]. Yet, despite decades of efforts, a secure and usable approach to verify online credentials still remains a challenge. Attackers have started utilizing social engineering methods such as multi-factor fatigue attacks to weaken the second factor. Traditional phishing attacks garner the user's password and in some cases succeed in compelling the user to approve the second factor notification prompt. These forms of attacks have troubled even employees of large technology corporations such as Cisco [15] and Uber [23]. While existing 2FA solutions are already under security attacks, enabling 2FA for all login accesses substantially increases the time to login to around (≥10s) [45], negatively affecting user experience.Table 4 presents a succinct comparison between existing protocols and UWB-Auth. In short, existing 2FA systems are deficient in both security as well as speed of access.

How to change the password and recover from lost tokens? When the fingerprint or motion signature authentication is used, the token functions as a password manager which stores the user credentials. In this case, changing the password on a website would require the user to possess the token and perform procedures similar to registration. For account recovery when a token is lost, fallback methods like recovery codes or email-based recovery can be used to enable password replacement.

In summary, UWB-Auth expands the available options for protecting access to online accounts. It flips the ordering of the traditional two factors by requiring proof of have-factor before the know-factor or biometric-factor, verifies token-to-login-device distance with UWB as proof of have-factor, and explores various know-factor and biometric-factor demonstrating solutions in different physical forms. We intend to develop UWB-Auth into a FIDO2 standards compliant authentication platform in the future, and encourage further scientific enquiry in this field.

REFERENCES

[1] Adafruit fingerprint sensor. https://www.adafruit.com/product/4750.

[2] Airtag. https://www.apple.com/airtag/.
[3] Apple watch. https://www.apple.com/watch/.
[4] Current mfa fatigue attack campaign targeting microsoft office 365 users. https://www.gosecure.net/blog/2022/02/14/current-mfa-fatigue-attack-campaign-targeting-microsoft-office-365-users/.
[5] Dealing with phantom mfa challenges in microsoft 365. https://duo.com/blog/mfa-fatigue-what-is-it-how-to-respond.
[6] Duo mobile app. https://duo.com/product/multi-factor-authentication-mfa/duo-mobile-app.
[7] Ergonomic monitor positioning. https://workerscomp.nm.gov/sites/default/files/documents/publications/Ergonomics/Info_Ergonomics_Computer_Monitor_Positioning.pdf.
[8] Fido2. https://fidoalliance.org/fido2/.
[9] Google 2-step verification. https://safety.google/authentication/.
[10] How does phishing work? https://www.yubico.com/resources/phishing/.
[11] Mfa fatigue: What it is and how to respond. https://duo.com/blog/mfa-fatigue-what-is-it-how-to-respond.
[12] Nearby interaction with uwb. https://developer.apple.com/nearby-interaction/.
[13] Onlykey. https://www.samsung.com/us/smartphones/galaxy-z-flip5/.
[14] Oura. https://www.ouraring.com.
[15] Phishing attack on cisco. https://www.cybersecuritydive.com/news/cisco-phishing-attack/629442/.
[16] Qorvo dw1000 chip. https://store.qorvo.com/products/detail/dw1000itr13-qorvo/681945/.
[17] Rsa securid. https://www.rsa.com/products/.
[18] Samsung uwb.
[19] Signicat. https://www.signicat.com/.
[20] Yubikey. https://www.yubico.com/.
[21] Ieee standard for low-rate wireless networks. *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pages 1–709, 2016.
[22] Usman Alhaji Abdurrahman, Mustafa Kaiiali, and Jawad Muhammad. A new mobile-based multi-factor authentication scheme using pre-shared number, gps location and time stamp. In *2013 International Conference on Electronics, Computer and Computation (ICECCO)*, pages 293–296. IEEE, 2013.
[23] Lawrence Abrams. Phishing attack on uber. https://tinyurl.com/34r7muz7.
[24] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.
[25] Yifeng Cao, Ashutosh Dhekne, and Mostafa Ammar. Itracku: Tracking a pen-like instrument via uwb-imu fusion. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '21, page 453–466, New York, NY, USA, 2021. Association for Computing Machinery.
[26] Haige Chen and Ashutosh Dhekne. Pnploc: Uwb based plug & play indoor localization. In *2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2022.
[27] Alexei Czeskis, Michael Dietz, Tadayoshi Kohno, Dan Wallach, and Dirk Balfanz. Strengthening user authentication through opportunistic cryptographic identity assertions. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 404–414, 2012.
[28] Sanchari Das, Gianpaolo Russo, Andrew C Dingman, Jayati Dev, Olivia Kenny, and L Jean Camp. A qualitative study on usability and acceptability of yubico security key. In *Proceedings of the 7th Workshop on Socio-Technical Aspects in Security and Trust*, pages 28–39, 2018.
[29] Rachna Dhamija, J Doug Tygar, and Marti Hearst. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590, 2006.
[30] Ashutosh Dhekne, Ayon Chakraborty, Karthikeyan Sundaresan, and Sampath Rangarajan. {TrackIO}: Tracking first responders {Inside-Out}. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 751–764, 2019.
[31] Ashutosh Dhekne, Mahanth Gowda, Yixuan Zhao, Haitham Hassanieh, and Romit Roy Choudhury. Liquid: A wireless liquid identifier. In *Proceedings of the 16th annual international conference on mobile systems, applications, and services*, pages 442–454, 2018.
[32] Bernhard Großwindhager, Michael Stocker, Michael Rath, Carlo Alberto Boano, and Kay Römer. Snaploc: An ultra-fast uwb-based indoor localization system for an unlimited number of tags. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, pages 61–72, 2019.
[33] Tzipora Halevi, Di Ma, Nitesh Saxena, and Tuo Xiang. Secure proximity detection for nfc devices based on ambient sensor data. In *Computer Security–ESORICS 2012: 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10-12, 2012. Proceedings 17*, pages 379–396. Springer, 2012.
[34] Dianqi Han, Yimin Chen, Tao Li, Rui Zhang, Yaochao Zhang, and Terri Hedgpeth. Proximity-proof: Secure and usable mobile two-factor authentication. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 401–415, 2018.
[35] Gerhard P Hancke and Markus G Kuhn. An rfid distance bounding protocol. In *First international conference on security and privacy for emerging areas in communications networks (SECURECOMM'05)*, pages 67–73. IEEE, 2005.
[36] Mohammed Jubur, Prakash Shrestha, Nitesh Saxena, and Jay Prakash. Bypassing push-based second factor and passwordless authentication with human-indistinguishable notifications. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 447–461, 2021.
[37] Andre Kalamandeen, Adin Scannell, Eyal de Lara, Anmol Sheth, and Anthony LaMarca. Ensemble: cooperative proximity-based authentication. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 331–344, 2010.
[38] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. {Sound-Proof}: Usable {Two-Factor} authentication based on ambient sound. In *24th USENIX security symposium (USENIX security 15)*, pages 483–498, 2015.
[39] Arash Habibi Lashkari, Samaneh Farmand, Dr Zakaria, Omar Bin, Dr Saleh, et al. Shoulder surfing attack in graphical password authentication. *arXiv preprint arXiv:0912.0951*, 2009.
[40] Michael McLaughlin and Billy Verso. Asymmetric double-sided two-way ranging in an ultrawideband communication system, November 26 2019. US Patent 10,488,509.
[41] William Melicher, Darya Kurilova, Sean M Segreti, Pranshu Kalvani, Richard Shay, Blase Ur, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Michelle L Mazurek. Usability and security of text passwords on mobile devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 527–539, 2016.
[42] Philip Polleit and Michael Spreitzenbarth. Defeating the secrets of otp apps. In *2018 11th International Conference on IT Security Incident Management & IT Forensics (IMF), IEEE*, pages 76–88, 2018.
[43] Leena Rao. Paypal debuts its biometrics and smartwatch integrations with samsung. https://techcrunch.com/2014/04/11/paypal-debuts-its-biometrics-and-smartwatch-integrations-with-samsung/.
[44] Kasper Bonne Rasmussen and Srdjan Capkun. Realization of {RF} distance bounding. In *19th USENIX Security Symposium (USENIX Security 10)*, 2010.
[45] Ken Reese, Trevor Smith, Jonathan Dutson, Jonathan Armknecht, Jacob Cameron, and Kent Seamons. A usability study of five two-factor authentication methods. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, pages 357–370, 2019.
[46] Yanzhi Ren, Chen Chen, Hongbo Liu, Jiadi Yu, Zhourong Zheng, Yingying Chen, Pengcheng Huang, and Hongwei Li. Secure mobile two-factor authentication leveraging active sound sensing. *IEEE Transactions on Mobile Computing*, 2022.
[47] Anthony Rosati. Two factor authentication using near field communications, March 14 2017. US Patent 9,594,896.
[48] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.
[49] Richard Shay, Iulia Ion, Robert W Reeder, and Sunny Consolvo. " my religious aunt asked why i was trying to sell her viagra" experiences with account hijacking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2657–2666, 2014.
[50] Enis Ulqinaku, Daniele Lain, and Srdjan Capkun. 2fa-pp: 2nd factor phishing prevention. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, pages 60–70, 2019.
[51] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
[52] Dimitra Zarafeta, Christina Katsini, George E Raptis, and Nikolaos M Avouris. Ultrasonic watch: Seamless two-factor authentication through ultrasound. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.
[53] Jiansong Zhang, Zeyu Wang, Zhice Yang, and Qian Zhang. Proximity based iot device authentication. In *IEEE INFOCOM 2017-IEEE conference on computer communications*, pages 1–9. IEEE, 2017.
[54] Tianyue Zheng, Zhe Chen, Jun Luo, Lin Ke, Chaoyang Zhao, and Yaowen Yang. Siwa: See into walls via deep uwb radar. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pages 323–336, 2021.
[55] Feng Zhou and Fernando Torre. Canonical time warping for alignment of human behavior. *Advances in neural information processing systems*, 22, 2009.
[56] Hao Zhou, Taiting Lu, Yilin Liu, Shijia Zhang, Runze Liu, and Mahanth Gowda. One ring to rule them all: An open source smartring platform for finger motion analytics and healthcare applications. In *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation*, pages 27–38, 2023.