WearLock: Unlocking Your Phone via Acoustics using Smartwatch

Shanhe Yi, Zhengrui Qin[†], Nancy Carter, Qun Li
College of William and Mary, [†]Northwest Missouri State University
{syi,njcarter,liqun}@cs.wm.edu, [†]zqin@nwmissouri.edu

Abstract—**Smartphone lock screens are implemented to reduce the risk of data loss or compromise given the fact that increasing amount of person data are accessible on smartphones nowadays. Unfortunately, many smartphone users abandon lock screens due to the inconvenience of unlocking their phones many times a day. With the wide adoption of wearables, token-based approaches have gained popularity in simplifying unlocking and retaining security at the same time. To this end, we propose to take advantage of the smartwatch for easy smartphone unlocking. In this paper, we have designed WearLock, a system that uses acoustic tones as tokens to automate the unlocking securely. We build a sub-channel selection and an adaptive modulation in the acoustic modem to maximize unlocking success rate against ambient noise only when those two devices are nearby. We leverage the motion sensor on the smartwatch to reduce the unlock frequency. We offload smartwatch tasks to the smartphone to speed up computation and save energy. We have implemented the WearLock prototype and conducted extensive evaluations. Results achieved a low average bit error rate (BER) as 8% in various experiments. Compared to traditional manual personal identification numbers (PINs) entry, WearLock achieves at least 18% unlock speedup without any manual effort.**

I. Introduction

As smartphone stores a wide variety of sensitive information of the owner, it is critical to provide effective protection for smartphone data. Currently, every smartphone operating system has a built-in screen lock application, which enables users to unlock their smartphones via PINs, passwords, patterns, etc. However, the reality is that a significant portion of users never lock their smartphones. A recent study [1] indicated that 53 out of 150 (35%) of participants have never enabled any sort of screen lock and the primary reason was due to the inconvenient input methods of screen locks. In another study [2], a large portion of participants (57.1%) indicated that they use none or naive screen lock (e.g. slide-to-unlock) while lots of participants (46.8%) agreed that unlocking their phones can be annoying and many of them (25.5%) admitted that they want a way to unlock their phone much easier. Therefore, the problem of user authentication on mobile devices is how to balance the security and the user experience [3].

To address this problem, one direction is to reduce the number of unlocks upon existing authentication mechanisms. There are two common approaches. One is to provide partial functionality on lock screens, enabling smartphone interactions before unlocking. This technique potentially reduces the number of unlocks, thus easing the unlock burden on users but at the cost of information security. For example, this approach may display several lines of an incoming email on a locked screen for user. However, those few lines may contain sensitive data. Further judgement from users is needed to determine what functionality or information is safe on the locked screen. The other approach is to choose the right moment to surface the authentication instead of enforcing it at each user session [4], which eventually involves some sort of implicit authentication methods. This scheme is not suitable for use in screen lock due to noticeable delays [2].

Another more promising direction to solve this problem without security tradeoff is to find the most suitable authentication method for mobile devices. The commonly seen authenticators on smartphone can be categorized into *passwords* (“what you know”), *biometrics* (“who you are”), and *tokens* (“what you have”) [5]. The term *password* in this paper includes words, phrases, patterns, PINs, or their combinations, which are used as secrets for authentication. However, this approach is problematic for mobile devices for several reasons. First, simple passwords are easy to guess while strong passwords are hard to remember. Second, the input environments on mobile devices introduce difficulties for users to enter passwords consisting of characters, digits, and symbols. Third, even though pattern or graphical passwords are much easier to input but all those passwords including previously mentioned are susceptible to shoulder surfing attacks or smudge attacks. Alternatively, *biometric*-based authentication uses unique features (e.g. fingerprints, eye iris, faces, voices, etc.) extracted from the human body and is considered convenience and secure. Recent work has also considered various gestures and inputting habits [6]–[9] as the sources for biometric extraction. However, one big disadvantage of biometric-based authentication is that those biometrics are uniquely tied to human body and are not as replaceable as passwords or tokens when being compromised or disclosed [10]–[12]. The *token*-based authentication usually includes contact-less proximity card, smart card with static or dynamic tokens. The advantages of token are easy-to-use, no need to memorize passwords, while the disadvantage is the cost of additional hardware.

In this paper, we seek a smartphone authentication solution in line with token-based method. Ideally, we want a secure screen lock that 1) authenticates user on each interaction; 2) is resistance to malicious observers; 3) requires minimal effort from user. Originally, the token-based solution is less favored due to the cost of additional hardware as mentioned. However, due to the increasing popularity of smart things and wearables, this solution has re-gained attentions [13]–[15]. Based on a market research of Kantar Wearable Technology [16] and Morgan Stanley [17], 12% of US consumers own at least one wearable device while 55% of consumers have intentions to buy at least one wearable devices. Hence, we envision that many smartphone users will possess at least one peripheral wearable device, such as a smartwatch or smartband, in the near future. Therefore, we investigate the solution which leverages pervasively co-located trusted devices for token-based authentication to create an automated and secure screen lock approach. Nevertheless, it is not easy to find a proper channel to conveniently establish a secure range to associate smartphone with co-located trusted devices (e.g. a smartwatch in our system). Solutions utilizing Near Field Communications (NFC) tags as trusted devices require users to manually attach a tag close to the phone's NFC antenna to achieve proximity of 10 cm or less. Solutions based on Bluetooth-enabled wearables, speakers and cars, can constantly connect to smartphones, but the connection range of Bluetooth cannot be securely guaranteed. Variants such as device model, paired device, and local environment may sustain a Bluetooth connection up to 100 meters in distance [18]. In our preliminary experiment, we have confirmed that android trusted devices based on Bluetooth do not lock one's phone until the trusted devices are 10-15 meters away in line-of-sight or 2-3 rooms away in none-line-of-sight. If someone takes your smartphone and stay not too far away from your trusted device, he may access your unlocked phone since your trusted device is still connected via Bluetooth.

To address those concerns, in this paper, we propose to exploit the acoustic channel to build the trusted relationship between a smartphone and its associated smartwatch and automatically unlock the phone when the smartwatch is nearby. To this end, we build WearLock, a system to automatically unlock smartphones via an acoustic channel between a smartphone and its associated wearable, i.e., a smartwatch in this paper. To be noted that our system is not meant to replace current smartphone authentication schemes (password or biometric based authentications), but provides a secure and efficient alternative which can significant reduce authentication effort on users. The assumption is that with a given noise level, we can maintain secure acoustic channel within roughly 1m distance between two devices using speaker and microphone, which acts as the secure boundary. Microphones and speakers are commonly available on these devices, eliminating the need for extra hardware additions. The communication range of acoustic channel is shorter than the Bluetooth and longer than NFC or magnetic-based channel [19], which is more desirable for the purpose of unlocking smartphones. One challenge is to build a robust and reliable acoustic modem scheme to secure the acoustic channel when devices are nearby. The other is to carefully design a system to accommodate the limited battery capacity and computation power of wearable hardware.

In summary, we make the following contributions:

- We proposed a novel automated and secure unlocking scheme for smartphone via a trusted wearable device. It requires minimal amount of effort from user.
- We are the first, to the best of our knowledge, to exploit the adaptive modulation of acoustics on common of-the-shelf (COTS) mobile devices for robust data transmission. The acoustic modem can adapt to ambient noise levels and interfering signals.
- We built WearLock on unmodified COTS smartphone and smartwatch devices and evaluated the system extensively.

II. System Overview

In this section, we describe the system architecture of WearLock and the smartwatch-assisted unlocking protocol.

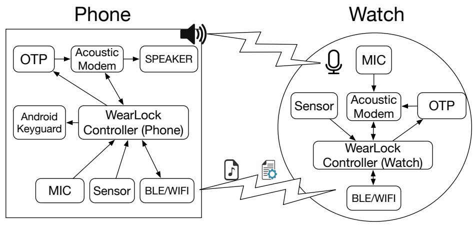

Fig. 1: The architecture of WearLock.

System Architecture. Figure 1 illustrates the architecture of WearLock, which consists of a smartphone and a smartwatch. The smartphone usually has a speaker and microphone, a wireless interfaces (Bluetooth or WiFi), and optionally motion sensors. The smartwatch usually has a microphone, a wireless interface and optionally motion sensors. Both devices run an instance of WearLock Controller, as the agent executing our proposed unlocking protocol, which takes input from underlying hardware and controls the the output channels such as speaker for emitting acoustics, wireless radio for sending configurations, and Android Keyguard for enabling or disabling lock screen. The one time password (OTP) module is responsible for the one time password generation and verification. The acoustic modem is an OFDM modem which enables data such as OTP to be transmitted over the acoustic channel using proper modulation schemes.

The smartphone and the smartwatch communicate with each other through both the wireless and the acoustic channels. The wireless channel serves as the secure control channel, transmitting acoustic channel configuration information, including the pilot sub-channel, the null sub-channel, and the data sub-channel. The acoustic channel conveys data payload in data sub-channels along with pilot sub-channels. The motion sensor will be used to construct a pre-filter to skip unnecessary unlocking requests by matching the motion pattern. In the following sections, we will provide further details on the acoustic OFDM modem design, the secure unlocking scheme, and several system optimizations.

Smartwatch-assisted Unlocking Protocol. Figure 2 illustrates the overall protocol of WearLock between the smartphone and the smartwatch. The protocol has two phases: 1) Phase 1 is Request-to-Send/Clear-to-Send (RTS/CTS) phase for channel probing; and 2) Phase 2 is the data transmission phase for OFDM modulated OTP token.

Smartphone's view: To avoid continuous probing and monitoring, we design to start our protocol when the user clicks the power button. The smartphone detects the presence or absence of the wireless link with the smartwatch. When the wireless link is present, the smartphone continues to evaluate the motion patterns of the smartphone and the smartwatch, respectively. If the motion patterns match, it is assumed that both are co-located and the smartphone continues to operate by verifying recorded audio token from the smartwatch. If the token is validated, then the Android Keyguard service will maintain the smartphone in screen unlocked state. During this process, if the wireless link, or the motion pattern, or the token validation fails, subsequent computations will be skipped and the Android Keyguard will remain the smartphone locked.

Smartwatch's view: The smartwatch runs a thin client, which cooperates with the smartphone controller. The smartwatch transmits information such as Bluetooth/WiFi status, sensor data, and recorded acoustics over the wireless channel.

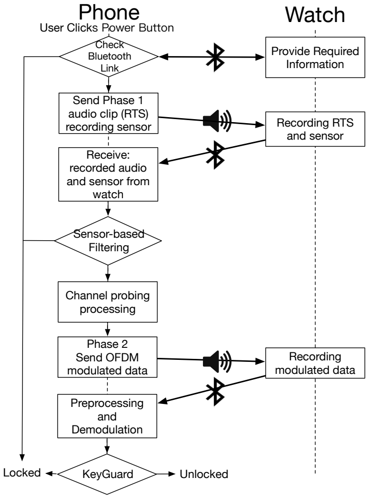
Phone
Watch
User Clicks Power Button
Check Bluetooth Link
Provide Required Information
Send Phase 1 audio clip (RTS) recording sensor
Recording RTS and sensor
Receive: recorded audio and sensor from watch
Sensor-based Filtering
Channel probing processing
Phase 2 Send OFDM modulated data
Recording modulated data
Preprocessing and Demodulation
Locked
KeyGuard
Unlocked

Fig. 2: The Protocol of WearLock.

III. Acoustic Modem Design

We designed and implemented a pure software modem for reliable data transmission over the acoustic channel. The goal is to meet the challenge of achieving robust communication under different ambient noise environments. We first discuss important characteristics of the acoustic channel. Then, we will describe our modem design, which includes signal detection using preamble identification, time synchronization using preamble and cyclic prefix, channel estimation and equalization with pilot tone, and signal modulation/demodulation. Figure 3 shows the block diagram of OFDM modem design.

The Acoustic Channel. Before diving into the design of acoustic modem, it is necessary to understand the important aspects of the acoustic channel, which significantly impact our design decisions. Next, we will discuss details of our OFDM modem design followed by practical considerations of our implementation.

1) Ambient Noise: Ambient noise directly affects the Signal-Noise-Ratio (SNR) at the receiver side. While ambient noise introduces many challenges, it also provides opportunities for co-location detection [20]. In order to measure the sound or noise power, we use the sound pressure level (SPL), which is defined as $\mathrm{SPL} = 20\log_{10}\frac{p}{p_{\mathrm{ref}}}$, where p is the root mean square (RMS) power and p_{ref} is a reference value.

2) Sound propagation and attenuation: In open air, the sound attenuation is mainly due to spreading loss. Assuming that SPL_{tx} and SPL_{rx} are the sound pressure levels at the transmitter and the receiver, respectively, and the distance between the transmitter and the receiver is d, then the sound attenuation in open air is defined as: $\mathrm{SPL_{tx}} - \mathrm{SPL_{rx}} = 20g\log_{10}(\frac{d}{d_0})$ where g is a geometric constant, with $g = 1$ for spherical propagation from a point source, and d_0 is a reference distance, i.e., the distance between transmitter's microphone and speaker [21].

In WearLock, we control the propagation range of acoustic signal by adjusting the speaker volume. We have measured the SPL at the receiver under line-of-sight (LOS) scenarios with different distances and volume settings, and the results are shown in Figure 4. From the figure, we can see that SPL attenuation match well with the theoretical value in spherical propagation, decreasing by about 6 dB when distance is doubled. Therefore, the Signal-to-Noise (SNR) at the receiver side can be estimated by $\mathrm{SNR_{rx}} = \mathrm{SPL_{rx}} - \mathrm{SPL_{noise}}$ where $\mathrm{SPL_{noise}}$ is the SPL of ambient noise.

3) Microphone and Speaker Characteristics: Ringing effect and *rise effect* adversely affect speaker and microphone performance [22]. Ringing is the effect that the speaker generates a longer output than the real length of input with a reverberation tail slowly reducing to 0. Similarly, rise effect is due to the fact that the speaker unit cannot reach to its highest power instantly. To overcome these effects, we define a zero-padding symbol guard interval T_g larger than the largest reverberation length to reduce the inter-symbol interference (ISI), and we also apply fading at the beginning of the signal.

OFDM Design. WearLock leverage orthogonal frequency division multiplexing (OFDM) modulation to modulate our token information. OFDM efficiently utilizes spectrum by allowing overlap in the frequency domain. It is also more resistant to frequency selective fading by enabling sub-channel selection and equalization techniques.

1) Modulation and Demodulation: The OFDM modulation and demodulation are simply implemented through Fast Fourier's Transformation (FFT) algorithms.

Considering a data sequence input to the IFFT, $X = [X_0, X_1, \cdots, X_k, \cdots, X_{N-2}, X_{N-1}]$, where $X_k = X_I(k) + jX_Q(k)$, which is in the form of quadrature amplitude modulation (QAM). Usually, the conversion back and forth between a binary data and the QAM-represented data input is done through a constellation mapping/de-mapping. To get the baseband modulated time-domain signal, we apply the IFFT:

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_A(k)e^{j(\frac{2\pi}{N}f_k t_n + X_P(k))} \tag{1}$$

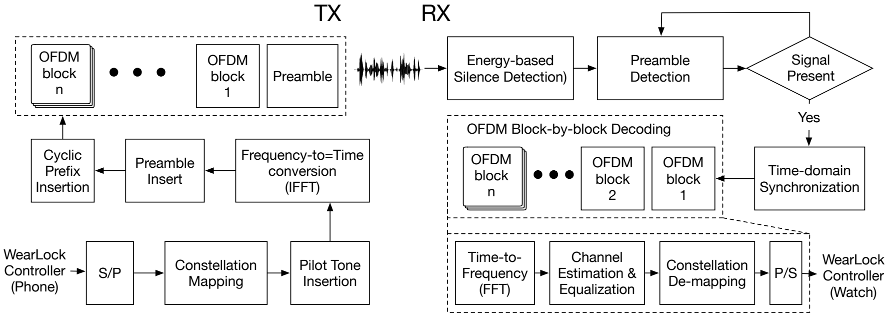

Fig. 3: The OFDM modem of WearLock.

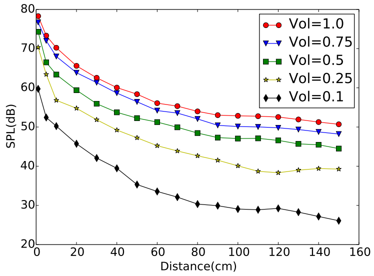

Fig. 4: Receiver's SPL in distance of different volume settings. Measured in a quiet room with the SPL of ambient noise about 15-20 dB in a line-of-sight scenario.

where $f_k = k/(N\Delta t)$, $F_s = 1/\Delta t$ is the sampling rate, and $t_n = n\Delta t$. And $X_A(k) = \sqrt{X_I(k)^2 + X_Q(k)^2}$, $X_P(k) = \arctan(X_Q(k)/X_I(k))$. Then the final representation of the signal is its real part $s_n = Re\,(x_n)$. We directly use this base-band signal as our output acoustic signal and send it through the speaker. To demodulate a received time-domain signal, we just apply the FFT and then look at the complex representation of X_k in the result, and de-map it according to the constellation diagram. However, due to the characteristics of the acoustic channels, which present delay, attenuation and phase distortion issues, we need to implement synchronization, sub-carrier selection, channel estimation and channel equalization.

2) *Sub-carrier Frequency Range:* Originally, we want to work on the near-ultrasound frequency ranged from 15kHz to 20kHz for the following reasons: 1) the frequency range of most ambient noise in our scenarios is below 15kHz; 2) humans are most sensitive to frequencies between 2,000 and 5,000 Hz; and 3) many new smart devices support native 44.1kHz or even higher sampling rate which indicates that the frequency response is acceptable below 20kHz. However, in real device experiment (A Moto 360 Android watch), we have found that there is a mandatory built-in low-pass filter, which limits the frequency range no higher than 7kHz, where the signal fades significantly from 5kHz to 7kHz[1]. Therefore, our design supports a smartphone-smartwatch pair utilizing audible acoustic signals (1kHz-6kHz) and an emulated smartphone-smartphone pair utilizing inaudible near-ultrasound acoustic signals (15kHz-20kHz).

3) *Preamble Design:* Existing preambles used in acoustic OFDM modems are usually based on PN-sequence or linearly frequency modulated (LFM) signals. The PN-sequence signal is a sequence of signal that has very strong auto-correlation output and weak cross-correlation output. The LFM signal is also known as Chirp signal or Sweep signal, which has nice Doppler-shift insensitivity and can be accurately detected in matched filtering. Therefore, we adopted a chirp signal for signal detection and coarse synchronization. The chirp signal increases from $f_{\min}$ to $f_{\max}$ in a time frame T_p.

4) *Silence Detection and Signal Detection:* The purpose of signal detection is to find the target signal in the recorded acoustic stream. First, we use an energy-based detector to filter out the section of silence. When there is a strong signal with SPL that surpasses our predefined noise level, we will perform the signal detection, relying on the detection of a known preamble. A cross-correlator calculates a normalized score and compares against a threshold value. Once we have detected a target signal, we will send this audio buffer to next processing block.

5) *Synchronization:* Finding the start of a frame is critical to all the follow-on processing and thus the system performance. Our synchronization has two steps: a coarse time-domain synchronization and a fine time-domain synchronization. The coarse synchronization in time-domain is done during the preamble detection through cross-correlation of the received signal and the known preamble. The preamble is a chirp signal, which correlates well with the original chirp even if there is a frequency shift. This characteristic ensures that we can always find a coarse start of the frame. During the processing of OFDM symbol, we perform the fine time-domain synchronization by leveraging the cyclic prefix. The cyclic prefix is a technique prefixing a symbol with a repetition of its end, which usually serves as a guard interval to eliminate ISI and is a technique to improve the robustness of multi-path propagation. For the purpose of fine time-domain synchronization, we use a window-based method, to iteratively find the best match of the head and tail of the signal after delay adjustment. Assume the time domain signal is $x(t)$, and the

[1] We deem the reason of filtering as the main microphone usage in Android wear is speech recognition. We are planning to test on more android wear models.

length of cyclic prefix is T_g, we have

$$\underset{t_f}{\operatorname{argmin}} \sum_{t=t_c+t_f}^{t_c+t_f+T_g} x(t)x(t+T_s), \quad \forall t_f \in [-\tau, \tau] \tag{2}$$

where T_s is length of symbol excluding the guard interval, t_c is the coarse delay, and τ is the searching range for t_f of a finer synchronization.

6) Channel Estimation and Equalization: Acoustic channel requires channel estimation and equalization techniques to overcome the distortions caused by fast fading, delay spreading, and multipath propagation. We insert equal-spaced unit-powered pilot tones for the purpose of equalization. To get the channel estimation, we extract pilot tones in frequency domain after proper synchronization as $z(k)$ where $k \in \mathbb{P}$, the pilot sub-channel set. Since it is equal-spaced in the frequency domain, we then apply a FFT-based interpolation with a proper interpolation length to expand it to estimate the data channel frequency response $H(k)$, $k \in \mathbb{P} \cup \mathbb{D}$, where $\mathbb{D}$ is the data sub-channel set. And $H(k) = z(k)$ when $k \in P$. Then, the equalization on the pilot and data channel is calculated as follows: $\hat{s}(k) = \frac{z(k)}{H(k)}, \quad k \in \mathbb{P} \cup \mathbb{D}$. By equalizing the known *a-priori* pilot sub-channel to unit-power, we equalize the data channel at the same time.

7) Adaptive Modulation: WearLock supports modulations such as BASK/QASK, BPSK/QPSK, 8PSK and 16QAM. We adopt an adaptive modulation scheme, which has a Request-to-Send/Clear-to-Send (RTS/CTS) phase before the data transmission phase. The motivation of adaptive modulation is that in every round, we want to make sure that the acoustic signal can be delivered reliably from smartphone to the nearby smartwatch in spite of the ambient noise and interfering signals. As is well known, the higher the order of modulation, the higher the date rate R. R can be calculated by $R = \frac{|D| r_c \log_2 M}{T_g + T_s}$, where M is the modulation order, $|D|$ is size of data sub-channel set, r_c is the coding rate for channel coding, and $r_c = 1$ if no channel coding is used. Higher order modulations are more vulnerable to ambient noise and interference. This usually requires a higher SNR to maintain the same error rate as a lower order modulation. Therefore, dynamically adaptive modulation are adopted by many communication systems, in which they sense the channel quality and adapt the modulation under certain constraints. Unlike traditional adaptive modulation for communication systems which seeks to maximize the system data rate, our design goal is to utilize the propagation loss in transmission to select a modulation mode to maintain a BER under a target BER. In the RTS/CTS phase, WearLock sends out a preamble with a block-based pilot symbol as a channel probing packet, which will serve the purpose of sub-channel selection and modulation selection.

Channel probing and sub-channel selection: It is important for WearLock to find the long-term or short-term noise which lasts for at least the time of transmission, like periodically-restarting air conditioner, which overlays certain frequencies for undefined duration. By sending a channel probing packet, WearLock can get an estimate of the channel state information and rank all the candidate sub-channels by the noise power. WearLock also chooses sub-channels in a priority order from low frequency to high frequency, and from low noise power to high noise power. We will assess the performance of sub-channel selection in our evaluation.

Pilot-based SNR indicator: From the channel probing result, we can also estimate the pilot signal SNR as an indicator for adaptive modulation. In order to measure and compare the performance of different modulation schemes, we use a normalized signal-to-noise ratio (SNR) as metric: E_b/N_0, which is the ratio of the energy per bit to noise power spectral density. It can be calculated as $\frac{E_b}{N_0} = \frac{C}{N} \cdot \frac{B}{R} \propto \mathrm{PSNR} \cdot \frac{B}{R}$ where B is the bandwidth, and R is the data rate, as we have discussed previously.

The $\frac{C}{N}$ is the carrier to noise power ratio, which will be estimated using a pilot-based SNR [23], which can be calculated from the spectrum result:

$$\mathrm{PSNR} = \frac{\mathbb{E}_{k \in \mathbb{P}}\left[X(k) \cdot X^*(k)\right] - \mathbb{E}_{k \in \mathbb{N}}\left[X(k) \cdot X^*(k)\right]}{\mathbb{E}_{k \in \mathbb{N}}\left[X(k) \cdot X^*(k)\right]} \tag{3}$$

where $\mathbb{N}$ is the null sub-channel set.

Deciding transmission mode: We have measured how BER of different modulations change in terms of different $\frac{E_b}{N_0}$ in a quiet room (15-20db SPL) and LOS. We control the ambient noise by an external speaker playing white noise audio we collected. The result is shown in Figure 5, in which the scatter plots are fitted by logarithmic tread-lines. The ranking order of our measures closely matches the theoretic result [24]. Due to hardware limitations, 16QAM is not usable in real experiments or at least may need heavy error correction techniques. Also due to the uneven responses of amplitude modulation and phase modulation of the audio hardware, amplitude-shift keying needs less SNR per bit than phase-shift keying. Therefore, we setup three transmission modes in total: QASK, QPSK, and 8PSK.

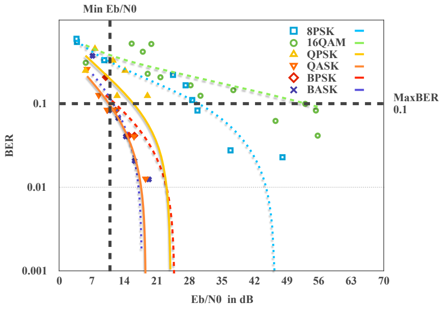

Fig. 5: The BER of different modulations changes with Eb/N0

Ambient noise measurement: The ambient noise is measured in the first processing phase at both sides. The smartphone also conducts a self-recording while the smartwatch is actively recording the incoming signals. By detecting the preamble existing in those recordings, we can coarsely align the two time series. The time series before the preamble are used to calculate the ambient noise. The ambient noise similarity is used to filter the cases that those devices are apparently not co-located. The noise level is also used to set proper speaker volume to control the transmission range.

NLOS filtering: To detection NLOS, we use a low cost method which analyzes the received preamble: a LFM modulated signal sent in the RTS/CTS phase. We first check the maximal normalized cross correlation score. If the max score is below a certain threshold (0.05 in our experiment), we will abort the transmission, since it indicates a mismatch on the preamble with high possibility. Otherwise, we can coarsely synchronize the signal. Next, we approximate a delay profile of the preamble using cross correlation. The root mean square (RMS) delay is calculated as $\tau_{\text{rms}} = \sqrt{\frac{\sum_n (t_n - \hat{\tau})^2 A(t_n)}{\sum_n A(t_n)}}$ where $A(t_n)$ is the approximate delay profile, $t_n = \frac{n}{F_s}$ and $\hat{\tau} = \frac{\sum_n t_n A(t_n)}{\sum_n A(t_n)}$ When the τ_{rms} is beyond a certain threshold τ_*, we assume that there is a severe body blocking.

How adaptive modulation works: According to our preliminary measurements in Fig. 4, in the first phase, a probing packet is sent out using a SPL(volume) that surpasses the SPL of noise at least a minimal SNR around 1 meters: $\text{SPL}_{\text{tx}} - 20\log_{10}(\frac{1.0}{d_0}) - \text{SPL}_{\text{noise}} > \text{SNR}_{min}$ where SNR_{min} can be decided from a minimal E_b/N_0, such as marked in Fig. 5. This ensures that the receiver in the range receives this probing packet. WearLock has no explicit ranging and we use this as the bound on the transmission range, if a receiver falls within this range, it will be able to receive the signal which is beyond the minimal SNR. The actual received SNR is estimated by the pilot-based SNR and will be reported in the CTS signal. After the transmitter gets the SNR_{rx}, this one is used to select the modulation scheme that can reach a BER at least smaller than a decided bound, the MaxBER as we have also marked in Fig. 5. For example, if the rx's SNR converts to $E_b/N_0 = 35dB$ and $\text{MaxBER} = 0.1$, we can send the signal using 8PSK, since we can get a guaranteed BER. If $\text{MaxBER} = 0.01$, then we can choose modulation like QPSK and QASK.

IV. Secure Unlocking

Existing work uses SIC to secure information transmitted in the acoustic channel. However, in our scenario, it is not feasible since most android wearable devices are not shipped with speakers. Therefore, we employed one time password (OTP) scheme to make use the acoustic channel with no secret disclosed.

Threat Model. We assume that the wireless link is securely established, and can safely be used as a control channel for OFDM communication. The sound channel is assumed to be insecure and an attacker can eavesdrop. We also assume that the attacker cannot take possession of the smart watch since it is hard to steal the watch from user's wrist without being caught. An attacker may take control of the phone and try to peak into it for the purpose of online payment, private photos and emails, etc. In order to fool the WearLock system, we assume that an attacker may try to perform various attacks. One is the co-located attack, in which the attacker holds the user's phone to get as close to the target as possible without being discovered. Another one is a record-and-replay attack, in which the attacker makes use of recording and replay devices to capture the acoustic signal and replay it to the smartphone. Jamming or Denial-of-Service attacks are not considered, since we can simply turn back to traditional locking scheme on smartphones. Currently, our design cannot protect acoustic channel against sophisticated relay attack which relies on some sort of relay to extend the range of between those two devices. However, we will argue the difficulty of launching this attack in acoustic channel, then discuss potential counter-measures.

One Time Password. To defend against replay attacks, we employ a counter-based one time password scheme(i.e., IETF RFC 4226 [25]). Assume that the phone and watch have negotiated a secret key k and a counter c through Bluetooth link, which can also be updated at anytime. The one time password is generate by keyed-hash message authentication code (HMAC) using `HMAC-SHA-1`, as $\text{HMAC}(k, c)$. Then a dynamic truncation (DT) technique is used to extract a 32 bit binary from the 160-bit result, which ensures that the outputs on different counter inputs are uniformly distributed. The final digits are generated by the DT result taking modulo 10^{Digit}, where Digit is the number of digits.

Security Discussion. As we have mentioned, an attacker possessing the victim's phone, will try various attacks. We have identified the following attacks and explained why our system can defend against or mitigate those attacks.

1) Brutal Force Attack: An attacker takes possession of victim's phone, will try to mount brutal force attack when the victim wearing smartwatch is in another room or quite far away while the Bluetooth is still linked. The attacker need to properly guess the acoustic modem parameters and guess the OTP. A 32 bits OTP has a large keyspace as 2^{32} and we can easily increase the keyspace by adding more data channels or using higher order modulations. The smartphone will be locked up after three consecutive failures, which makes the brutal force attack unrealistic.

2) Co-located Attack: Being similar to brutal force attack, the attack just tries to get close enough to the victim to perform a successful unlock. The defense against this attack lies in the design of the modem that there is high bit error rate when the transmission distance is beyond around 1 meter. Getting closer to the user and covering the smartphone stealthily may not work, since it will obstruct the direct path and result in significant loss when acoustic channel becomes NLOS.

3) Record and Replay Attack: Since attack can monitor the acoustic channels, disclosing the OTP token may suffer from a replay attack, in which an attacker can record the token signal and replay it to the watch like the man-in-the-middle (MITM) attack. This attack is defeated by examining the timing window, since in the protocol, we can measure the software stack delay and wireless round-trip-time. A MITM attacker with recorder and player in the loop definitely adds more delay in the acoustic path. Every time the power button is pressed, a Bluetooth message is sent to the watch indicating the start of the protocol, and the watch replies a Bluetooth message and starts recording. Then the smartphone starts to send acoustic token, after which smartphone also sends a Bluetooth message of stopping recording. And the watch will stop recording as well. This procedure has two phases, and it is interactive, which means we can examine the result of the first phase, and abort the second phase if there is anything specious. Since the OTP token is sent in the second phase, we avoid the disclosure of OTP token in such attack.

4) Relay Attack: Sophisticated relay attack will try to use record-and-replay in a live manner, to circumvent the time window based protection. If this attack can be performed in ideal case, our current design cannot protect acoustic channel against this attack. However, this attack is very hard to mount since it needs very flat frequency/phase-response speaker/microphone to avoid acoustic distortions in ADC and DAC. Otherwise, we can use fingerprinting method to unique identify those acoustic hardware to check if there are relays. Additionally, high quality speaker/microphone usually cannot be made in small sizes, which enlarges the chance being spotted by victim. Another potential counter-measure is to employ distance bounding protocol [26].

V. Performance Optimizations

WearLock Controllers are the running instances of our system on the smartphone and smartwatch. One task of WearLock Controller is to gather information from various sources and make the final decisions on questions such as where to run the computation and when to abort a transmission, which gives us plenty of opportunities for performance optimizations. The rationale is that the change of the way of unlocking smartphone using a paired smartwatch does not actually reduce the frequency of unlocking. Every audio transmission is followed by a series of intensive computations, which would be a burden on wearable devices. Even though the microphone and speaker power consumption are relatively low, digit signal processing computations such as cross correlation, FFT based Modulation and Demodulation, FFT based interpolation are all relatively computationally intensive, consuming more power.

We believe that by well addressing those questions, we can not only save energy for wearable devices but also reduce the delay of processing. We conduct computation load balance and computation reduction as two main solutions.

Computation Offloading. To mitigate the power drain on wearables, we leverage the natural computation pattern of the smartphone and its paired wearable, offloading heavy computation tasks from the smartwatch to the smartphone. Since all the acoustic modem and digital signal processing libraries are implemented as a common module shared by both phone-side and watch-side apps, we can easily partition the computations among those two devices.

In order to understand the trade-off here, we have measured the time cost of processing after the recording and the corresponding rough power consumption, in Figure 6. The processing mainly consists of a sliding window based cross correlator and an OFDM demodulator. Since it is not possible to tear apart the Android smartwatch and connect it to a power meter, we run our system for 50 rounds of acoustic unlocking and rely on the Android OS battery status to roughly measure the power consumption by the API provided by Android framework. To be noted that, this energy consumption measure is pretty rough, as the measurement procedure keeps the device awake, violating the life-cycle design pattern of an Android wear app. We anticipate more energy saving in daily usage. From the result, we can see that by offloading to the smartphone, it not only saves energy but also reduces the computation time.

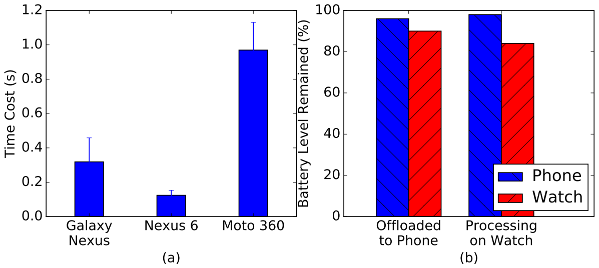

Fig. 6: Time Cost (a) and Power Consumption (b) Comparison on Offloading and Local Processing on Wearable.

Computation Reduction. The basic idea of the computation reduction is to leverage a series of filters using information such as wireless network, ambient noise and motion sensors, to avoid unnecessary follow-up heavy computation. For example, the WearLock only works when the Bluetooth link exists. Therefore, if there is no Bluetooth link, all the protocols and algorithms will not run. Alternatively, the technique used in Sound-Proof [20] is complementary to WearLock by leveraging the similarity of ambient noise, to eliminate unnecessary acoustic transmission, which is scheduled in the RTS/CTS phase of adaptive modulation. If the ambient noise similarity is below a threshold, we believe those two deices are not co-located with a high confidence and then the transmission is aborted. Additionally, we can also leverage the activity context information or hand movement derived from sensor units to reduce the number of acoustic transmissions.

Leveraging Motion Sensor-based Filtering: When the user is engaged in activities, or the smartphone is hold by the same hand that wears the watch, we can use the raw inertial sensor data to detect the device movement similarity. This will serve as a filter that can eliminate unnecessary acoustic transmission if the similarity distance is lower or higher than predefined thresholds. In order to use sensor traces, we need to convert the 3-axis sensors to its magnitude representation by $s \leftarrow \sqrt{s_x^2 + s_y^2 + s_z^2}$, since it is challenge to obtain accurate relative orientation between those two devices. The alignment of the sensor time series is not necessary since we use Dynamic Time Warping (DTW) to find the best alignment in the time domain [27]. The procedure is presented in Alg. 1.

Even though the time complexity of DTW is $O(n^2)$ assuming two inputs are length of n, it is very cheap since n is usually small ranging from 50 to 150 samples. We will verify the feasibility and measure the time cost in the evaluation.

Algorithm 1 Sensor-based Filter

```
procedure SENSOR-BASED FILTERING
    for each first phase do
        while recording do
            $sp_{x,y,z} \leftarrow$ phone accelerometer
            $sw_{x,y,z} \leftarrow$ watch accelerometer
        $sp \leftarrow \mathrm{Normalized}(\mathrm{Magnitude}(sp_{x,y,z}))$
        $sw \leftarrow \mathrm{Normalized}(\mathrm{Magnitude}(sw_{x,y,z}))$
        if $\mathrm{DTW}(sp, sw) > d_h$ then
            abort protocol                    ▷ save the computation
        else if $\mathrm{DTW}(sp, sw) < d_l$ then
            skip second phase                 ▷ save the computation
        else
            continue to the second phase
```

VI. Evaluation

In this section, we will first briefly discuss the implementation details. Then, we will evaluate our system in terms of communication range, adaptive modulation, sensor-based filtering, system delay, a filed test and a case study.

Implementation Details. We have implemented our system on Android OS, consisting of Android phone app and Android wear app. We have wrapped the MessageAPI and ChannelAPI of Android Wear SDK for implicit message/file transferring so that we do not need to handle the underlying networking using either Bluetooth or WiFi. We have also ported the wear app to a smartphone in order to test near-ultrasound frequency in WearLock. The OFDM modem is written in pure JAVA libraries, which can be running on both sides. The digital signal processing library is also written in JAVA and we plan to move on native DSP library in the future. The default FFT size is 256 and the sampling rate is 44.1 kHz, which gives about 172Hz sub-channel bandwidth. We index our channels from 1-256. and in default we pick channel $\{16, 17, 18, 20, 21, 22, 24, 25, 26, 28, 29, 30\}$ as data channels, and $\{7, 11, 15, 19, 23, 27, 31, 35\}$ as pilot channels for working at 1-6kHz frequency band. The rests are null channels. We shift this channel assignment with higher index when we want 15-20Khz frequency band. This channel assignments will be adjusted during sub-channel selection. The preamble size is 256 samples, the post-preamble guard size is 1024 samples and the CP duration is 128 samples. All those parameters can be easily tuned in the setting activity of our app.

Communication Range. The communication range is a very important performance metric. Ideally, we want to the communication range to be strictly constrained within one meter. However, the performance varies due to different modulations and ambient noise. In Figure 7, we show the communication range of the acoustic modem in terms of BER in three different transmission modes. They are measured at an office room with a LOS setup. We can see that by constraining the max BER we can adaptively change the transmission mode to guarantee that the signal fades significantly when the current communication range is increased.

Adaptive Modulation. To understand the performance of adaptive modulation, we have conducted two experiments. First, we enable adaptive modulation selection in the previous measurements to show the effectiveness of adaptive modulation. In Figure 8, by constraining the BER, we can adaptively change the modulation schemes, which can allow us to have shorter packets or more redundant bits. It also guarantees that an eavesdropper located nearby will have a larger BER since a higher order modulation is more vulnerable to noise and interference. Next, we demonstrate WearLock adaptation to ambient noise in sub-channel selections. We use audible frequency range for this experiment and employ an external tone generator as an acoustic jammer, the Audacity, which supports at most 6 mono-tracks simultaneously. We use QPSK modulation with the smartwatch and smartphone placed at a fixed distance about 15cm. The jammed sub-channel index is randomly selected every time. The result, depicted in Figure 9, shows that when the sub-channel selection is enabled, the modem is able to avoid the noisy or interfered sub-channels and maintain a stable BER.

Sensor-based Filtering. We have also evaluated the sensor-based filtering to see how much similarity in sensor data we can leverage to reduce the number of acoustic transmissions. We tested WearLock in activities such as sitting, walking and jogging, and also in different activities. The normalized DTW scores and the running time are reported in Table II. The activity context can be queried through Google Play Service APIs. By setting a threshold on the DTW scores (0.1 in our case), we can reduce the Max BER or skip the second phase when the DTW score is under the threshold and abort the transmission when the DTW score is above the threshold.

Activities	Sitting	Walking	Running	Different	Cost(ms)
DTW Scores	0.05	0.02	0.06	0.20	45.9

TABLE II: Sensor-based Filtering

System Delay. The system delay is important since users will lose their patience with the WearLock technique if it is much slower than entering a password. There are two types of delay: computation delay and communication delay. We have broken down the computation delay into phase 1 channel probing processing, phase 2 pre-processing and phase 2 demodulation in Figure 10 when the computation is carried out on different devices. We have also measured the communication delay in WiFi/Bluetooth message and file transfer in Figure 11. Every experiment is repeated at least 20 times. We did not measure the modulation since the generation is very fast. Part of them can be generated ahead-of-time and therefore the cost can be amortized. For purpose of comparison, we also measured the time cost for a user entering 4/6-digit PIN codes on an Android device using similar method as [2]. The results are also aligned to the medians of measurements in [2]. We compare the results with three different configurations: *Config1*: the fastest case where the smartwatch offloads computation via WiFi to a high end smartphone (Nexus 6), *Config2*: the slowest case where the smartwatch offloads computation via Bluetooth to a low end smartphone (Galaxy Nexus), and *Config3*: local processing case where the processing is on the smartwatch (Moto 360) as

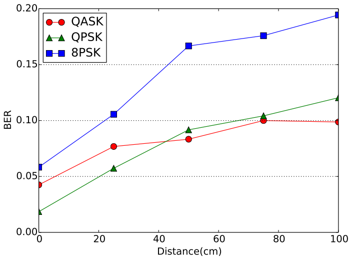

Fig. 7: The BER in distances and transmission modes (near-ultrasound).

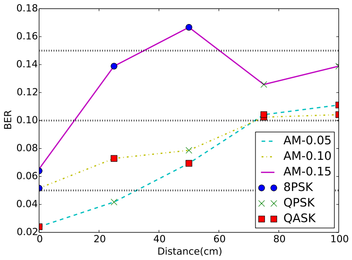

Fig. 8: The BER in adaptive modulation under different BER constrains(near-ultrasound).

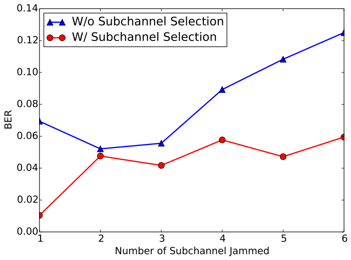

Fig. 9: The BER under jamming and subchannel selection. (QPSK, audible sound)

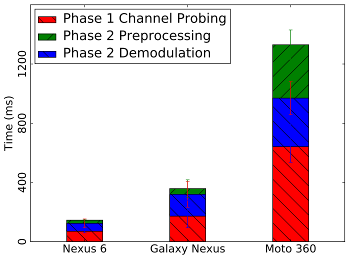

Fig. 10: The computation delay of each phase on different devices.

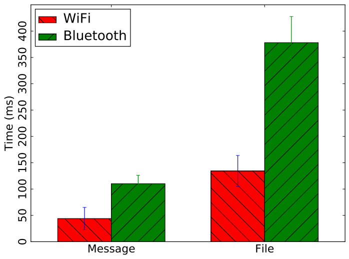

Fig. 11: The communication delay between smartphone and smartwatch.

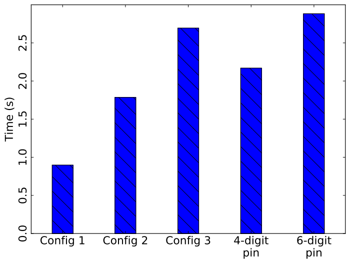

Fig. 12: Compare the total delay in different configurations with manually entering pin codes.

BER v.s. Locations	Office	Class Room	Cafe	Grocery Store
Diff. Hand (Audible)	0.0486(8PSK)	0.0333(8PSK)	0.0263(QPSK)	0.0119(QPSK)
Same Hand (Audible)	0.0889(8PSK)	0.0512(8PSK)	0.0655(QPSK)	0.0648(QPSK)
Diff. Hand (Near-ultrasound)	0.0556(8PSK)	0.0417(QPSK)	0.0233(QPSK)	0.0139(QPSK)
Same Hand (Near-ultrasound)	0.1054(QPSK)	0.1875(QPSK)	0.1971(QPSK)	0.2060(QPSK)

TABLE I: Field Test Result. The average BER is around 0.08.

shown in Figure 12. The results indicate WearLock has a delay advantage over manually unlocking even on a low end device and slow Bluetooth link with a speedup of at least 17.7%. For the fastest case, the WearLock speedup is at least 58.6%. Notably, WearLock experiences less delay and only needs the user to click the power button.

Field Test. We tested WearLock with the smartphone and smartwatch hold or worn in different configurations: same hand and different hands. We also tested them in different locations as offices, classrooms, cafes and grocery stores where the typical sounds in those scenarios are human voice and noises from sources such as keyboard typing, cafe machines, air conditioners, etc. We report the BER results in Table I. From the results, we find that near-ultrasound may have less interference but significant signal fade due to direct path blocking in the same hand case. The audible sound is less convenient but more usable in most noise cases. It would be better to use inaudible sound in quiet spaces and audible sound in noisy spaces as long as the volume is controlled. We can easily integrate this choice to current mobile OS since it is in line with how smartphone users set their the sound preferences.

A Case Study. We asked five graduate students to try our system in a class room environment one by one and made detailed observation during the procedure. One of the students held the bottom of the phone tightly covering the speaker at the beginning. In this case, it gives a success rate of 3/10 when required BER=0.1. We asked the student to try the second time without holding the phone so tight. In this case, the success rate is 8/10 when BER=0.1 and 10/10 when BER=0.15. One student held the phone in one hand and wore the watch on another hand, which yielded a success rate of 8/10 at BER=0.1. One of the students preferred to use the phone with one hand and wore the watch on the same hand, which gave a success rate of 4/10 if using BER threshold as 0.1. However, the NLOS detection can identify 3/10 as NLOS cases. If relaxing the corresponding required BER of NLOS cases to 0.25, the corrected success rate is 7/10. The average success rate among five participants is 90%. From the perspective of convenience, although the participants have perceived the delay due to retry after failures in certain scenarios, they still felt that our scheme was convenient and rated higher in convenience level comparing to entering 6-digit PINs manually. The overall error rate is acceptable and they felt no harassment to repeat the unlocking via acoustics in case of failures. We leave as future work a comprehensive user study involving more participants and environments.

VII. Discussion and Limitations

Acoustic Frequency Range: Due to the frequency range limitation of the mobile acoustic hardware, the implemented system can work on audio range (1-6Khz) on a phone-watch pair, and near-ultra sound range (15-20Khz) in a phone-phone pair. This brings the limitation that the acoustic is either audible or can be possibly heard by babies or animals. This is one limitation of our work and we leave this to the smartphone manufacture when devices with higher sampling rate will be made. For example, several latest models of Samsung Galaxy Note supports 96kHz and higher audio recording/playback. Device with higher sampling rate can utilize higher and more frequency bands with less noise and more bandwidth.

Bluetooth Proximity: According to the document of Bluetooth proximity profile that even the link between devices has been securely enabled, the device can be spoofed into assuming that the other device is close due to the internal design of Bluetooth protocol, which means that naively using Bluetooth proximity profile for secure distance measurement is not encouraged [28]. Currently secure distance measurement using Bluetooth requires additional development upon existing stacks. Comparatively, our system on mobile and wearable devices can be easily implemented in the application level and ported to other devices. However, we do admit that a solution via Bluetooth is promising, and we leave this in our future work to explore secure ranged and easy-to-implement token-based authentication in wireless channel.

VIII. Related Work

There are two main areas related to our work. First, we will briefly outline the acoustic communication on mobile device and justify the difference of our work. Then, we will discuss the existing work about authentications with reduced efforts.

Acoustic Communication on Mobile Devices. WearLock is an extension of acoustic communications work on smart devices. Dhwani [22] aims to replace NFC with an acoustic orthogonal frequency division multiplexing (OFDM) modem secured by a self-interference cancellation (SIC) technique. Dolphin [29] and PriWhisper [30] also leverage similar idea for secure acoustic channel. However, their schemes are not suitable for practical and efficient implementation on phone-watch pairs, since most smartwatches have no speakers and generating a cancellation signal imposes both energy and processing burdens on wearable devices. We use a different secure scheme tailed for smartwatch which acts as a listener in acoustic channel and conduct offloading to shift computation and energy burdens on smartwatch to more capable smartphone. Work [31] used On-off keying on chirp signals to overcome one of the main limitations of acoustic communication on mobile devices: the short communication range. However, our work make a good use of the relatively short communication range, and we use OFDM which yields much higher data rate. Google NearBy [32] is a recently published API to provide near filed communication and interaction using Bluetooth, WiFi and acoustics. The acoustic signal is modulated in Dual-tone multi-frequency signaling (DTMF), which is slower and less spectrum-efficient compared to OFDM. However, it requires the devices to support near-ultrasound in 18.5kHz-20khz and therefore is not supported on Android Wear devices yet. Other work requires the provision of special acoustic communication hardware [15], [33], [34]. WearLock requires no additional special hardware.

Reduced-Effort Authentication. The reduced-effort authentication are those techniques that seek to reduce or eliminate the human effort involved in authentications. The simplest schemes utilize short-range radio communication using Bluetooth or NFC. ZIA [35] is one of the earliest work with zero-interaction authentication, leveraging an authentication token. WearLock can be taken as a natural extension from PC and electronic tokens to the nowadays common smartphone-smartwatch pairs. Work [4] has proposed the combination of multiple signals to define a security confidence level and subsequent the authentication only at certain levels. Their scheme can reduce the authentication frequency but requires large effort in data collection and training. Similarly, work [36] has proposed a method to lock the device when the users physical separation is detected. Their method is complementary to ours and can be combined. Another way of reducing effort in authentication is to leverage device co-location or localization [20], [37]–[39]. Sound-Proof [20] has proposed to leverage similarities in ambient noise signals for user authentication. Sound-of-silent [37] has proposed to utilize the silence patterns in recordings to provide co-location context. However, these techniques cannot defend against co-located attackers due to their reliance on relatively widely pervading and unvalidated signals. WearLock relies on the presence of a validated acoustic signal that is designed not to be detectable more than one meter away from the generating device.

IX. Conclusion

In this paper, we show that a convenient and secure smartphone unlocking can be achieved by leveraging a paired smartwatch. We argue that the smartwatch is an ideal wearable token device that is theft-proof and has constant connections to the phone. Smartphone users can save much effort from unlocking. WearLock, the implemented system, secures the acoustic channel by adapting the transmission power and modulation configurations, and sends an OTP tokens for validation via acoustics to unlock the smartphone. To optimize the system performance, we offload the heavy computation to the phone, and leverage multi-source information including sensor data to reduce unnecessary audio transmissions. WearLock can achieve an average bit error rate of 8% in our experiments. WearLock achieves at least 18% speedup even on a low-end device, compared to entering PINs.

X. Acknowledgments

The authors would like to thank all the reviewers for their helpful comments. This project was supported in part by US National Science Foundation grant CNS-1320453. We also acknowledge the support of the Arts & Sciences Graduate Research Grant, College of William & Mary, for the smartwatch used in this research.

References

[1] D. Van Bruggen, S. Liu, M. Kajzer, A. Striegel, C. R. Crowell, and J. D'Arcy, "Modifying smartphone user locking behavior," in *Proceedings of the Ninth Symposium on Usable Privacy and Security*. ACM, 2013, p. 10.

[2] M. Harbach, E. von Zezschwitz, A. Fichtner, A. De Luca, and M. Smith, "It'sa hard lock life: A field study of smartphone (un) locking behavior and risk perception," in *Symposium On Usable Privacy and Security (SOUPS 2014)*, 2014, pp. 213–230.

[3] A. Adams and M. A. Sasse, "Users are not the enemy," *Communications of the ACM*, vol. 42, no. 12, pp. 40–46, 1999.

[4] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos, "Progressive authentication: deciding when to authenticate on mobile phones," in *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*, 2012, pp. 301–316.

[5] L. O'Gorman, "Comparing passwords, tokens, and biometrics for user authentication," *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2021–2040, 2003.

[6] J. Sun, R. Zhang, J. Zhang, and Y. Zhang, "Touchin: Sightless two-factor authentication on multi-touch mobile devices," in *CNS'16*, San Francisco, CA, Oct. 2014, pp. 436–444.

[7] Y. Chen, J. Sun, R. Zhang, and Y. Zhang, "Your song your way: Rhythm-based two-factor authentication for multi-touch mobile devices," in *INFOCOM'15*, Hong Kong, China, Apr. 2015, pp. 2686–2694.

[8] S. Yi, Z. Qin, E. Novak, Y. Yin, and Q. Li, "Glassgesture: Exploring head gesture interface of smart glasses," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE, 2016, pp. 1–9.

[9] Z. Hao and Q. Li, "Towards user re-authentication on mobile devices via on-screen keyboard," in *Hot Topics in Web Systems and Technologies (HotWeb), 2016 Fourth IEEE Workshop on*. IEEE, 2016, pp. 78–83.

[10] Z. Kleinman, "BBC News: Politician's fingerprint 'cloned from photos' by hacker," http://www.bbc.com/news/technology-30623611.

[11] Y. Zhang, Z. Chen, H. Xue, and T. Wei, "Fingerprints on mobile devices: Abusing and leaking," in *Black Hat Conference*, 2015.

[12] R. Brandom, "The Verge: Your phone's biggest vulnerability is your fingerprint - can we still use fingerprint logins in the age of mass biometric databases?" https://goo.gl/VmNKsP, May 2016.

[13] H. Bojinov and D. Boneh, "Mobile token-based authentication on a budget," in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*. ACM, 2011, pp. 14–19.

[14] M. Koschuch, M. Hudler, H. Eigner, and Z. Saffer, "Token-based authentication for smartphones," in *Data Communication Networking (DCNET), 2013 International Conference on*. IEEE, 2013, pp. 1–6.

[15] W. Wang, L. Yang, and Q. Zhang, "Touch-and-guard: secure pairing through hand resonance," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 670–681.

[16] Kantar, "Wearable technology report," https://goo.gl/N6DEV4.

[17] Morgan Stanley, "Wearable devices the 'internet of things' becomes personal," https://goo.gl/6MCO2M.

[18] Google, "Trusted bluetooth devices," https://goo.gl/xEZSCw.

[19] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, "Magpairing: Pairing smartphones in close proximity using magnetometers," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1306–1320, 2016.

[20] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun, "Sound-proof: usable two-factor authentication based on ambient sound," in *USENIX Security 15*, pp. 483–498.

[21] M. J. Crocker, *Handbook of acoustics*. John Wiley & Sons, 1998.

[22] R. Nandakumar, K. K. Chintalapudi, V. Padmanabhan, and R. Venkatesan, "Dhwani: secure peer-to-peer acoustic nfc," in *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4. ACM, 2013, pp. 63–74.

[23] H. Yan, L. Wan, S. Zhou, Z. Shi, J.-H. Cui, J. Huang, and H. Zhou, "Dsp based receiver implementation for ofdm acoustic modems," *Physical Communication*, vol. 5, no. 1, pp. 22–32, 2012.

[24] B. Sklar, *Digital communications*. Prentice Hall NJ, 2001, vol. 2.

[25] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, "Hotp: An HMAC-based one-time password algorithm," *The Internet Society, Network Working Group. RFC4226*, 2005.

[26] S. Brands and D. Chaum, "Distance-bounding protocols," in *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1993, pp. 344–359.

[27] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, 2009.

[28] P. WG, "Bluetooth proximity profile spec doc v1.0.1," July 2015.

[29] L. Li, G. Xue, and X. Zhao, "The power of whispering: Near field assertions via acoustic communications," in *ASIA CCS'15*. ACM, pp. 627–632.

[30] B. Zhang, Q. Zhan, S. Chen, M. Li, K. Ren, C. Wang, and D. Ma, "PriWhisper : Enabling keyless secure acoustic communication for smartphones," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 33–45, Feb 2014.

[31] H. Lee, T. H. Kim, J. W. Choi, and S. Choi, "Chirp signal-based aerial acoustic communication for smart devices," in *Proc. of IEEE Conf. on Computer Communications (INFOCOM), Hong Kong SAR, PRC*, 2015.

[32] Google Inc., "Nearby API," https://developers.google.com/nearby/.

[33] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling fine-grained indoor localization via smartphone," in *Mobisys*. ACM, 2013, pp. 235–248.

[34] G. E. Santagati and T. Melodia, "U-wear: Software-defined ultrasonic networking for wearable devices," in *Mobisys*. ACM, 2015, pp. 241–256.

[35] M. D. Corner and B. D. Noble, "Zero-interaction authentication," in *Proceedings of the 8th annual international conference on Mobile computing and networking*. ACM, 2002, pp. 1–11.

[36] T. Li, Y. Chen, J. Sun, X. Jin, and Y. Zhang, "ilock: Immediate and automatic locking of mobile devices against data theft," in *CCS'16*, Vienna Austria, Oct. 2016, pp. 933–944.

[37] W.-T. Tan, M. Baker, B. Lee, and R. Samadani, "The sound of silence," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2013, p. 19.

[38] H. Zhang, W. Du, P. Zhou, M. Li, and P. Mohapatra, "Dopenc: acoustic-based encounter profiling using smartphones," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 294–307.

[39] H. Han, S. Yi, Q. Li, S. Guobin, Y. Liu, and E. Novak, "AMIL: localizing neighboring mobile devices through a simple gesture," in *The 35th Annual IEEE International Conference on Computer Communications (INFOCOM 2016)*. IEEE.