DoubleEcho: Mitigating Context-Manipulation Attacks in Copresence Verification

Hien Thi Thu Truong*, Juhani Toivonen†, Thien Duc Nguyen‡, Claudio Soriente*, Sasu Tarkoma* and N.Asokan§
*NEC Laboratories Europe, Heidelberg, Germany
†University of Helsinki, Helsinki, Finland
‡Technische Universität Darmstadt, Darmstadt, Germany
§Aalto University, Espoo, Finland
hien.truong@neclab.eu; juhani.toivonen@cs.helsinki.fi; ducthien.nguyen@trust.tu-darmstadt.de;
claudio.soriente@emea.nec.com; starkoma@cs.helsinki.fi; asokan@acm.org

***Abstract*—Copresence verification based on context can improve usability and strengthen security of many authentication and access control systems. By sensing and comparing their surroundings, two or more devices can tell whether they are copresent and use this information to make access control decisions. To the best of our knowledge, all context-based copresence verification mechanisms to date are susceptible to *context-manipulation attacks*. In such attacks, a distributed adversary replicates the same context at the (different) locations of the victim devices, and induces them to believe that they are copresent. In this paper we propose `DoubleEcho`, a context-based copresence verification technique that leverages acoustic Room Impulse Response (RIR) to mitigate context-manipulation attacks. In `DoubleEcho`, one device emits a wide-band audible chirp and all participating devices record reflections of the chirp from the surrounding environment. Since RIR is, by its very nature, dependent on the physical surroundings, it constitutes a unique *location signature* that is hard for an adversary to replicate. We evaluate `DoubleEcho` by collecting RIR data with various mobile devices and in a range of different locations. We show that `DoubleEcho` mitigates context-manipulation attacks whereas all other approaches to date are entirely vulnerable to such attacks. `DoubleEcho` detects copresence (or lack thereof) in roughly 2 seconds and works on commodity devices.**

***Index Terms*—room impulse response, copresence verification, context-manipulation attack, acoustic, reverberation time**

I. Introduction

A number of authentication mechanisms leverage copresence of the (alleged) prover and the verifier to either strengthen authenticity or improve usability [6], [9], [10], [13], [33]. For example, in the context of IoT, pairing of two or more devices may only be allowed among those that are copresent. Similarly, payment systems may mitigate fraudulent in-store transactions by checking that the payment card being used and the mobile device of the legitimate card-holder are close to each other [9], [13].

Copresence may be verified, for example, by exchanging an unpredictable value over a short-range communication channel. The verifier device transmits a random value over Bluetooth or NFC, and challenges the prover device to echo that same value out of band.[1] However, such a naïve solution is vulnerable to *relay attacks*. The latter involves a pair of victim devices *far away* from each other and a distributed attacker, copresent with each of the victims. The attacker simply relays messages between the victim devices so that they conclude to be copresent. Relay attacks have been demonstrated in research papers [6], [7] and reported in the news.[2]

Distance bounding techniques, e.g., [1], [2], [6], [20] may be used against relay attacks. However, distance bounding relies on accurate measurements of round-trip times between the two devices; it therefore needs to be deployed at the lower levels of the communication stack, and often requires special hardware [20].

An alternative approach to copresence verification, that is easier to deploy than distance bounding, is based on *context*. The rationale behind context-based copresence verification is that two copresent devices should perceive similar context. Previous work has demonstrated the feasibility of using different context modalities, such as audio [9], [10], [27], radio signals [8], [33], or other features of the physical environment like humidity or pressure [9], [23]. For example, in [9], [10], [27] two devices record environmental noise and compare their recordings to tell whether they are copresent or not.

Yet, depending on the context modality used to verify copresence, a *context-manipulation attack* may be feasible. Similar to a relay attack, a context-manipulation attack involves two far away victim devices and an adversary that is copresent with each of them. Here, the adversary manipulates the context at the locations of the two devices so that they conclude to be copresent. For example, if copresence verification is based on audio [9], [10], [27], the adversary sitting next to each of the two (far away) victim devices, may simply play an arbitrary audio clip at both locations [25]; the two devices will record the same audio and conclude that they are copresent by comparing their recordings.

To the best of our knowledge, no context-based copresence verification mechanism tolerates context-manipulation attacks. For example, many copresence verification mechanisms [11],

[1] The out of band channel is authenticated, e.g., by means of a shared key.

[2] http://uk.businessinsider.com/thieves-unlock-a-mercedes-using-device-relays-keys-signal-west-midlands-police-2017-11

[17], [30], [33] leverage context based on GPS, Wi-Fi and Bluetooth signals. Previous work has shown that manipulation of such context modalities is feasible [24], [28], [30], [36]. Audio-based copresence schemes [9], [10], [27], [30] are also susceptible to adversarial manipulation of the context [24], [25]. Copresence verification based on physical context [23], [35] is not secure to context-manipulation attacks either [24]. Real-world applications relying on context-based copresence verification (e.g., GPS[3] or radio beacons[4]) are similarly vulnerable to these adversarial manipulations.

A few recent proposals [24], [30] address context-manipulation adversaries, by combining multiple sensor modalities and by assuming that the adversary cannot manipulate all modalities at the same time (e.g., the adversary can manipulate Wi-Fi signals but it cannot manipulate GPS signals). Yet, if the adversary manipulates all modalities, the schemes of [24], [30] become completely vulnerable to context-manipulation attacks (see Table 2 in [24]).

Our contributions. *We aim at designing a context-based copresence verification mechanism that mitigates context-manipulation attacks.* We focus on audio-based copresence verification because of the wide availability of microphones and speakers on commodity devices. In this settings, we study how to leverage the physical characteristics of the location where the protocol is executed to mitigate context-manipulation attacks. We observe that in nature many animals use echolocation to "make sense" of the physical environments around them. Bats emit high frequency sounds and listen to how they are reflected by the environment; dolphins also use a similar echolocation technique. Inspired by these, we make use of Room Impulse Response (RIR), which depends on both sound waves within a physical enclosure as well as *the shape and the materials of the enclosure*. Sound travels via multiple paths from the source to the receiver. These paths are significantly influenced by the boundaries and the obstacles in the enclosing space. Thus, we conjecture that RIR constitutes a unique *location signature* that is hard for an adversary to replicate.

We instantiate the idea above in `DoubleEcho`, a mechanism for copresence verification based on RIR. `DoubleEcho` mitigates context-manipulation attacks by minimizing the chances that an adversary may reproduce the same context at two different locations. In `DoubleEcho`, one device emits a short (2 seconds), wide-band audible chirp and all participating devices (the playing device and one or more listening devices) record reflections of the chirp from the surrounding environment. `DoubleEcho` extracts features on different frequency bands from each recording and compares those features to determine whether the devices are copresent or not. Since RIR is, by its very nature, dependent on the physical surroundings, `DoubleEcho` can effectively mitigate relay and context-manipulation attacks. The same chirp signal played at two different locations yields different RIRs, unless the two locations are very similar (e.g., same shape, building materials, furniture, etc.).

We evaluate `DoubleEcho` and compare it with previous work by using various mobile devices and in a range of different environments. `DoubleEcho` is *sound* with a false negative rate (as low as 0.021) similar to the one shown by similar proposals [10], [27]. In face of a context-manipulation attack, `DoubleEcho` is *secure* with a false positive rate that ranges between 0.089 and 0.189, whereas all other approaches to date are completely vulnerable. Finally, `DoubleEcho` can be easily deployed on commodity devices — it only requires microphones on all participating devices and a speaker on one of them — and takes roughly 2 seconds to detect copresence.

The dataset we used for the evaluation of `DoubleEcho` is publicly available for research use [29]. Source code for data collection and RIR extraction is also available upon request.

II. Secure Contextual Copresence Verification

The notion of copresence may change depending on the application scenario. In some applications, copresence means that two devices are a few centimeters away (e.g., NFC payments). Others may label as copresent a pair of devices that are a few meters apart (e..g, bluetooth applications). Similar to previous work we consider that two devices are copresent if they are at most half a meter away and within the same room. This design choice suits applications such as unlocking a desktop using a mobile phone when it is in close proximity [31] and managing meeting memberships [27].

We consider an application scenario where two devices— prover P and verifier V— leverage context-based copresence verification. The protocol starts by the prover sending a copresence verification request to the verifier. At this time both devices measure the environment via one or more available sensors and the prover sends its measurement to the verifier. Transmission happens out-of-band on a channel authenticated, e.g., by means of a shared key. Finally, the verifier compares its measurement with the one received by the prover to decide whether the two devices are copresent. Replay protection can be achieved if the verifier sends a random nonce at the start of the protocol, and the prover piggybacks that same nonce in the measurement sent to the verifier.

In this setting, a true (resp. false) positive happens when P and V are copresent (resp. not copresent) and V's output is "copresent". Similarly, a true (resp. false) negative happens when P and V are not copresent (resp. copresent) and V's output is "not copresent".

Clearly, the copresence verification protocol should be *sound*, i.e., exhibit a low rate of false negatives.

Threat model. Let prover P and verifier V be the two victim devices and assume they are at different locations. The goal of the adversary is to run the copresence verification protocol between P and V and make V conclude that the two devices are copresent.

All previous proposals for context-based copresence verification [9], [10], [23], [30], [33] assume a standard Dolev-Yao

[3]http://www.solidpass.com/authentication-methods/mobile-location-authentication.html

[4]https://securechannels.com/products/authentication/

attacker [5], where the adversary controls the communication network but cannot break cryptographic primitives, nor it can compromise the victim devices. Further, the adversary is not allowed to manipulate the context that the devices sense to detect copresence.

We consider a stronger adversary that is copresent with both devices and we allow for *context-manipulation* attacks. That is we allow the adversary to manipulate the context at P and at V. Since we use audio to verify copresence, we let the adversary control the noise in the environment of the prover and in the one of the verifier.

Therefore, the adversary succeeds if, while under attack, the copresence verification protocol outputs a *false positive* (i.e., V concludes that P is copresent while the two devices are actually far away).

Naturally, the copresence verification mechanism should be *secure*, i.e., exhibit a low rate of false positives.

III. Room Impulse Response

An impulse response is the response of a system to an impulsive stimulus. In this work, we consider sound systems and their acoustic environments. By "system" we refer to the collective of involved parts such as the space, walls and obstacles within a room, but also the speakers and microphones on the devices.

When emitted from a speaker, a sound is distributed, albeit unevenly, in every direction, traveling multiple paths of different lengths. This results in multiple arrivals at the receiver. Sound that travels the shortest path (direct sound) arrives first and is the loudest. Sound that has traveled through other paths starts arriving soon after, with delay directly and loudness inversely proportional to the path length. Along all paths, air and surfaces absorb some of the sound energy.

An acoustic impulse response is created by how the sound is reflected along this multitude of paths; how much sound is delayed, how much sound is weakened, how much sound energy is received etc. The paths, and hence the impulse response, is heavily affected by the enclosing space such as walls, ceiling, obstacles, the power of the sound source, and the position and orientation of both the sound source and the receiver. Theoretically, two non-identical rooms should not have identical impulse responses.

Fig. 1 describes how an RIR looks like in the time domain. The propagation delay is the time it takes for direct sound to travel from the source to the receiver. The first part of the RIR is the arrival of the direct sound. After that, the earliest reflected sounds (lowest order reflections) start arriving, followed by sound traveling along a multitude of longer paths (higher order reflections, reverberant space). The noise floor is the point where the receiver can no longer distinguish the reverberant sound from background noise. Energy absorption by air and surface materials causes reverberant sound to decay as a function of time (and distance). In practice, the difference between direct sounds and early reflections might not be as clear as in Fig. 1. Reverberant decay is usually measured in the range from 5 dB below the level of direct sound to a point of 60 dB below that on the backward energy curve.

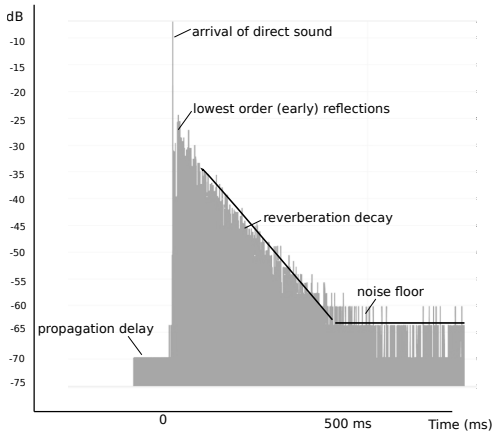

Fig. 1. RIR representation in the time domain.

We now briefly describe standard features of RIR signals; these features are later selected by a classification model and used in `DoubleEcho`.

Reverberation time (RT), the most common RIR feature, represents the time it takes for a reverberating signal to decay until it can no longer be distinguished from background noise. RT can be a single value even when measured for a wide band signal, for example a sweep from 20 Hz to 20 kHz. It can also be computed for different frequency bands separately.

A band covers a specific range of frequencies, with a lower bound f_l and an upper bound f_h. Octave bands are identified by their middle frequency f_0. A band is one octave in width when the upper band frequency is twice that of the lower band frequency, $f_l = f_0/2^{1/2}$, $f_h = f_0 \times 2^{1/2}$. An one-third octave band has $f_l = f_0/(2^{1/2})^{1/3} = f_0/2^{1/6}$ and $f_h = f_0 \times (2^{1/2})^{1/3} = f_0 \times 2^{1/6}$. Reverberation time varies for different frequency bands, and hence it should be indicated if the RT applies for a specific band.

One of the most accurate method to compute RT is based on studying the decay curve. The energy curve is a curve obtained by backwards integration of the squared impulse response, which ideally starts from a point where the response falls into the noise floor.

The slope of Schroeder curve is used to measure how fast the impulse response decays. For instance, RT60 is the time it takes for a sound to decay by 60 dB. This is the standard base for measuring RT according to Sabine's reverberation equation empirically developed in the late 1890s.[5] In practice, the level of direct sound might be less than 60 dB above the noise floor. In such cases, RT30 and RT20 are used instead. RT30 is the time a sound would decay 60 dB when extrapolated from a 30 dB decay range in the Schroeder curve (often from -5 dB to -35 dB). RT20 accordingly is extrapolated from a 20 dB decay range (from -5 dB to -25 dB).

[5]https://www.acoustics-engineering.com/html/sabin.html

Beside RT, there are other RIR features used for particular purposes of acoustical analysis. Early Decay Time (EDT) is derived from reverberation decay curve, conventionally in the section between 0 dB and 10 dB below the level of direct sound. EDT therefore is the reverberation time measured over the first 10 dB of the decay. It gives information of overall signal clarity and intelligibility in a room. Early-to-late energy ratio is a measure of the sound energy arriving within some specified interval after direct sound (first n milliseconds) over the energy in the remaining part of the impulse response. For example, clarity ratios C10, C35, C50, C80 are the early-to-late ratios (in dB) at 10, 35, 50, and 80 milliseconds as split times. Direct-to-Reverberant energy Ratio (DRR) is another useful measure for assessing the acoustic configuration.

IV. DoubleEcho

DoubleEcho has been designed for contextual copresence verification using RIR measurements. As RIR remains stable across a *single* environment, it is more resilient to context-manipulation attacks than previous contextual copresence systems. In the following we describe our overall system design, after which we detail how the RIR is measured.

A. System design

DoubleEcho uses audio as the context to verify copresence of two devices and leverages RIR to mitigate context-manipulation attacks. We focus on audio for two main reasons. First, sensors to capture audio signals (i.e., microphones) are widely available on commodity devices. Second, an audio signal is heavily affected by the surrounding space; thus, acoustic RIR may allow to tell whether two devices are within the same room.

Further, DoubleEcho accommodates scenarios where the entropy of context data is insufficient (e.g., a silent room), by *injecting* a stimulus (i.e., a chirp signal) in the environment. Therefore, we assume prover and verifier to be equipped with microphones and at least one of them to be equipped with a speaker.

The following steps show how DoubleEcho works when run between a prover device P and a verifier device V. We assume that P and V have access to an authenticated channel (i.e., share a secret key).

1) **Start.** The protocol may be triggered by either P or V. This is done by sending a well-known message over the authenticated channel. During this stage, V also sends a random nonce to P.[6]
2) **Playing and Recording.** The device with a speaker (without loss of generality we assume V has a speaker) plays a known audio clip $\mathcal{S}$; at the same time both devices record audio in the environment via their microphones.
3) **RIR.** V (resp. P) computes the RIR h_{V} (resp. h_{P}) based on the original audio clip $\mathcal{S}$ (the system's input) and the signal recorded via its microphone (the system's output). P sends its RIR h_{P} to V via the authenticated channel, along with the nonce received previously from V.
4) **Comparison.** V compares the nonce sent at the beginning of the protocol with the one received from P. If the two nonces are different, V assumes a replay attack and aborts. Otherwise, V compares h_{V} computed locally with h_{P} received by P to decide whether the two devices are copresent or not.

During the last step of the protocol, V compares its RIR with the one received from P. The comparison mechanism clearly has a dramatic impact on the performance of the system. That is, the comparison function should enable V to reliably detect whether P is copresent or not. DoubleEcho extract features from RIR and feeds them to a binary classifier that outputs a copresence verdict. In the remaining of this section we illustrate how we measure RIR; the next section provides details about the classifier and the features used in DoubleEcho.

B. Measuring RIR

A common method for measuring RIR is to apply a known input signal and measure the system's output. Assuming the system to be linear time-invariant,[7] the output signal $x(t)$ is the result of convoluting the input signal $s(t)$ and the RIR $h(t)$ (i.e., $x(t) = s(t) * h(t)$, where t is time and $*$ denotes convolution). Hence RIR can be extracted by deconvoluting input and output.

The measurement method, therefore, depends on the excitation signal and the deconvolution technique. Both the excitation signal and the deconvolution technique should maximize the Signal-to-Noise Ratio (SNR) and allow to eliminate non linear artifacts in the deconvolved impulse response.

Among various methods to measure RIR, we opt for the sine sweep technique because it overcomes the limitation of having distortion artifact when the condition of linear time variance is not fulfilled. This technique uses a linear (or an exponential) time-growing frequency sweep as the excitation signal. The output of the system in response to the sine sweep input consists of both the linear response to the excitation and the harmonic distortion at various orders due to non-linearity (Fig. 2). A deconvolution of the recorded output can be used to compute the RIR. The linear impulse response is the first order harmonic in the deconvolved result.

The steps to measure RIR in DoubleEcho are as follows:

1) Convert the recorded signal (system output) $x(t)$ and the generated signal (system input) $s(t)$ from time domain to frequency domain by computing discrete Fourier transform (FFT). The FFT of a signal represents amplitudes and phases of individual frequencies in the signal.
2) Compute their deconvolution to get the transfer function. This is done by computing the time reversal of the input signal and convolve it with the recorded signal using an FFT based method (Matlab fftfilt). The transfer function

[6]The nonce is used as a challenge to prevent replay attacks.

[7]That is, the system exhibits a linear relationship between input and output that does not change over time.

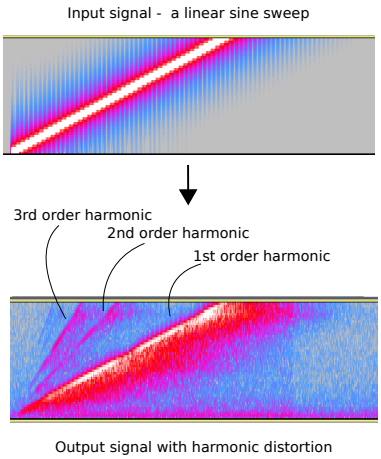

Fig. 2. Distortion artifact.

shows how each frequency has been affected by the system.

3) Convert the transfer function from the frequency domain to the time domain by inverse FFT to obtain the room impulse response.

From the RIR, we extract a series of acoustic features (RT60, EDT, D/R, C10, C35, C50, C80) across 32 frequency bands from 0 to 22050 Hz, including 1 wide-band, 10 octave bands, and 21 one-third octave bands. Features from these bands yield a vector of 224 features for each audio sample.

V. Evaluation

We started evaluation of `DoubleEcho` by tuning system parameters and designing experiments. Next we ran binary classification to assess the performance of copresence verification resulted by `DoubleEcho`. Given results, we conducted extra experiments to compare `DoubleEcho` with previous work and elaborated its advantages.

A. Parameters tuning

Copresence. The physical distance that defines copresence may vary from application to application. In our experiments, we consider that two devices are copresent if (1) they are *in the same room,* and (2) the distance between them is at most half a meter.

Excitation level. According to Section III, `DoubleEcho` must ensure the power of the audio signal to be at least 40-50 dB above that of the noise floor level, of which 20-30 dB to measure reverberation time and 10 dB to separate reverberation and noise floor. In detail, we evaluate reverberant decay over a range, starting from 5 dB below the direct sound down to 20 or 30 dB below the starting point. One option is to emit the excitation signal at the highest level that the source device can achieve. However, if the signal is too loud for the receiver, recording will be clipped. A clipped signal is a measurement error that prevents extraction of RIR. The challenge is thus to ensure enough excitation of the acoustic space, while at the same time avoiding the clipping effect.

Measurement duration. A sweep of 500 ms or 1s is sufficient for measuring small spaces such as kitchens or offices. Larger rooms would need longer reverberation times (e.g., a couple of seconds for medium-size rooms like classrooms, up to a few seconds for large indoor spaces). In our experiments we use a sweep of two seconds to cover a wider range of rooms. In practice, if a measurement does not return good results, the parties may increase the duration of the audio clip and re-run the measurement.

Audio sample and RIR features. The signal we use is a linear sine sweep from 0 to 22050 Hz during two seconds. Recording is done at 44100 Hz 16 bit PCM. To ensure that initializing the audio system does not cut away from our signal, we pad the signal with one second of silence in the beginning and two seconds in the end, making the full sample five seconds long.

Post-processing for recorded signal. We align the recorded signal with the original generated signal (using signal cross-correlation) and remove the parts before the beginning of the sine sweep and after 500 milliseconds past the end of the sine sweep, leaving us with just the sweep and the reverberation. We then normalize the recorded signal to account for decreased volume caused by the recording process.

Linear RIR extraction. The first order harmonic is detected based on the highest peak in the energy decay curve. We find the highest peak and cut from 100 ms before that until 750 ms after. Background noise level is determined by energy level of 10 ms before direct sound of the first order harmonic.

B. Experiment setting

Android application. We developed an Android app and a backend server to facilitate collection of audio samples. The protocol is triggered by the server that instructs one device to emit a chirp signal, and all devices, including the emitter, to record. After recording, the devices send the recorded sample to the backend server.

Triggering the protocol over the network allows us to record on many devices at the same time without needing to tap buttons on their screens, and makes synchronizing the emitting and recording devices easier. To account for the different amounts of delay that the devices may experience and to make sure that the entire signal gets played through the speaker, we prepared our audio files with a few seconds of silence padding both before and after the signal. We also implemented a loudness calibration feature to avoid unwanted effects such as signal clipping.

Locations and devices. We collected data with a set of 16 smartphones and tablets spanning different brands and models (Google Nexus 5X LG, Sony Xperia Z5 compact, Sony Xperia XA, Samsung Galaxy S7, HTC One A9s, Huawei P8, ZTE, and Lenovo Phab2). Data collection took place at two geographic locations (Helsinki and Darmstadt) and spanned 20 different rooms including offices, kitchen, corridor, classroom, meeting room, living room, bed room, stairway, computer room.

TABLE I
BENIGN COPRESENT AND ATTACK NON-COPRESENT PAIRS.

Experiments	No. benign pairs	No. attack pairs
Dataset1	203	8120
Dataset2.1	228	5725
Dataset2.2	189	3780
Dataset2.3	208	7350
Dataset2.4	220	6952
Dataset3	392	26605

Procedure. For each room, we took a subset of the devices available, randomly chose the emitting device, and placed the remaining devices in a circle around it at a distance of roughly half a meter. We then played and recorded 5 audio samples, thereby emulating 5 executions of the copresence verification protocol.

Datasets. We split collected data in two datasets, based on whether data collection was performed in Helsinki (Dataset1) or in Darmstadt (Dataset2). For each room we used in Darmstadt, we repeated the data collection in four different locations within that room. Resulting datasets are labelled Dataset2.1, Dataset2.2, Dataset2.3, and Dataset2.4 (where Dataset2 is the union of those four). We used four locations per room in order to understand whether the location within a room affects the performance of `DoubleEcho`.

Our data collection totalled 32975 pairs of audio samples. We discarded pairs where recording had failed for any reason (about 2% of all data samples). Details of datasets are shown in Table. I; Dataset3 is the result of merging Dataset1 and Dataset2.

Similar to previous work [23], [30], we use the collected audio samples to emulate benign as well as adversarial executions of the protocol. A benign execution of the protocol is one where the two devices are truly copresent. An adversarial execution happens when the two devices are not copresent. Any pair of samples recorded by two distinct devices (where one plays an audio clip) at the same time and in the same room is considered a benign pair. Any pair of samples recorded by two distinct devices (where one also plays an audio clip) in two different rooms is considered an attack pair.

The way we build attack pairs allows us to emulate a context-manipulation attack where prover and verifier are not copresent, and the adversary plays in the prover's environment the (well known) audio clip played by the verifier. We do not emulate the random nonce sent by the verifier as a challenge. Nevertheless, an attacker copresent with both the prover and the verifier could easily relay the verifier's random nonce to the prover.

C. Copresence classification

We consider copresence verification as a classification task for labeling two classes: *copresent* (positive class) and *non-copresent* (negative class). We tested different supervised learning algorithms and settled with RandomForest which performs best among all.

Our dataset is heavily imbalanced towards attack pairs and this comes from the method we use to generate our benign/attack pairs. We address the imbalance by undersampling the set of attack pairs during training; we use the RandomUnderSampler algorithm in the Scikit-learn library. We do not alter the imbalance during testing.

For each sample pair (of both benign and attack pairs), using their RIR acoustical feature vectors, we compute (component-wise) the vector of squared differences, yielding a 244-elements feature vector that are fed to the classifier. Though the feature vector has 244 elements, many of them are not informative for copresence and non-copresence classification. We applied the RandomForestRegressor algorithm implemented in Scikit-learn to select the 50 (empirically chosen among trials of 10, 20, 30, 50, 100 top features) most relevant features out of 244 available. Each training dataset we select 50-top features accordingly. Due to space limit, we do not list those selected features for each dataset.

We used 5-fold cross-validation: for each dataset, we randomly divided the data into 5 subsets (folds); 4 subsets were used for training the model and the remaining subset for testing. Table II shows the confusion matrix. Notably the false negative rate (FNR) ranges between 0.021 to 0.106. That is, when the two devices are copresent, `DoubleEcho` will fail to detect copresence no more than 10% of the times. Also, the false positive rate (FPR) ranges between 0.089 and 0.189 (5 over 6 datasets have FPR less than 12%). That is, in presence of an adversary, `DoubleEcho` will fail to detect attack roughly 12% of the times.

Classification results on datasets 2.1-2.4 show that locations in rooms for copresent pairs do not affect performance of the verification mechanism. That is, when two devices are within proximity of less than half a meter, at different locations in a room, `DoubleEcho` predicts copresence with similar performance.

D. Comparison with previous work

The effectiveness of `DoubleEcho` is better appreciated when our proposal is compared to previous work. In particular, we pick the audio-based copresence verification protocol of [30] as an alternative to `DoubleEcho`, and compare the two proposals in terms of usability, performance, and deployability. We show that `DoubleEcho` provides better security without giving up usability; furthermore, requirements to deploy `DoubleEcho` are the same as the ones to deploy the mechanism of [30].

Table III compares false positive rates (FPR) of `DoubleEcho` and [30] evaluated on our datasets. Recall that a false positive happens when the two devices are not copresent but the mechanism outputs a copresence verdict. From the table, it is clear that `DoubleEcho` mitigates context-manipulation attacks while the mechanism of [30] is completely vulnerable to such attacks. We note that the FPR reported in [30] is 0.093 but the adversary in [30] is not allowed context-manipulation attacks.

TABLE II
EXPERIMENT RESULTS. NOTATION: COPRESENCE **1**, NON-COPRESENCE **0**.

		Dataset1		Dataset2.1		Dataset2.2		Dataset2.3		Dataset2.4		Dataset3	
		Prediction											
		1	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**
Ground truth	**1**	198	5	215	13	185	4	186	22	202	18	379	13
	0	751	7369	728	4997	336	3444	1390	5960	992	6708	2450	24155
FNR		0.025		0.057		0.021		0.106		0.082		0.033	
FPR		0.092		0.127		0.089		0.189		0.129		0.092	

TABLE III
FPRS COMPARISON OF DOUBLEECHO AND [30].

Datasets	[30]	DoubleEcho
1	0.995	0.092
2.1	0.996	0.127
2.2	0.999	0.089
2.3	0.995	0.189
2.4	0.998	0.129
3	0.997	0.092

TABLE IV
FNRS COMPARISON OF DOUBLEECHO AND [30].

Datasets	[30]	DoubleEcho
1	0.000	0.025
2.1	0.004	0.057
2.2	0.005	0.021
2.3	0.005	0.106
2.4	0.000	0.082
3	0.003	0.033

Table IV compares false negative rates (FNR) of DoubleEcho and [30] evaluated on our datasets. Recall that a false negative happens when the two devices are copresent but the mechanism outputs a "non-copresent" verdict. A false negative hinders usability and it is desirable to have a low FNR. Both mechanisms show similar FNR across all datasets. That is, additional security of DoubleEcho is not traded for usability.

Finally, we note that both methods have the same requirements in terms of hardware, duration, energy consumption and binary classifier.

We did not compare DoubleEcho with other context-based copresence verification mechanisms that leverage audio [9], [10], [27]. We could not carry out the comparison because either the code was not available, lack of detailed description prevented us from replicating the proposed mechanism, or simply the considered system model was not comparable with the one of DoubleEcho.

VI. DISCUSSION

In this section we discuss limitations of our evaluation and possible application domains of DoubleEcho.

We have not controlled background noise levels or tested with different distances between the devices. All our experiments were made in the presence of an audible excitation signal emitted by one device. Also, devices were placed at a distance of roughly half a meter. In future work, we plan to assess how DoubleEcho performs in presence of (loud) background noise and when devices are more than half a meter apart.

DoubleEcho uses a wide-band audible chirp for measuring RIR. It may therefore be unsuitable for application scenarios where users my be uncomfortable with the noise DoubleEcho generates. A possible solution may be to use ultra-sounds. Given the limited capacity of commodity devices to emit and record reflections of high frequency sounds, using ultra-sound may face several system challenges and we leave it for future work.

Our evaluation of DoubleEcho focuses on indoor environments. We leave a feasibility study of DoubleEcho in outdoor environments as future work. Measuring RIR in large spaces may, however, be challenging with commodity devices.

We argue that DoubleEcho can be used in a number of application domains where it can help to mitigate context-manipulation attacks. Similar to Sound-Proof [10], DoubleEcho could be used in two-factor authentication system to enhance security while retain ease-of-use. DoubleEcho could be also used for multiple device association [27]. During our experiments, we have empirically tested that DoubleEcho can be effectively used to pair three or more devices at once. Finally, DoubleEcho can be used in zero-interaction authentication systems like BlueProximity++ [31] to mitigate context-manipulation attacks shown in [24].

Last but not least, we argue that DoubleEcho can detect DoS attacks. With this attack, an adversary can inject audio in the environment to scramble or confuse recordings. DoubleEcho can detect the attack by comparing the recorded signal with the priorly known input.

VII. RELATED WORK

The two main directions for copresence verification in the literature are *distance bounding* (radio distance bounding [1], [2], [6], [20]; audio distance bounding [18]) and *copresence verification* (RF based [11], [14], [17], [33]; audio [9], [10], [22], [26], [27]; multi-context [16], [23], [30], [31]; and other context [3], [4], [12], [15]). In this section we discuss approaches of the latter which are related to our work. For each approach we will argue why it does not meet our design goals.

Context-based copresence verification. Radio context such as GPS, Wi-Fi and Bluetooth is commonly used for proximity verification. Narayanan et al. [17] studied the use of various modalities including Wi-Fi broadcast packets and access points, Bluetooth, GPS, GSM and audio atmospheric gas for private proximity detection. They concluded that Wi-Fi performs the most prominently. Krumm et al. [11] proposed "NearMe" which also uses Wi-Fi features for proximity detection. Varshavsky et al. [33] presented Amigo to authenticate copresent devices using various features extracted from Wi-Fi environment. Although these solutions use the radio environment in different ways, they all depend on the availability of deployed base stations.

Acoustic copresence verification has been studied intensively. In existing solutions [9], [10], [22], [27], [30], devices passively perceive ambient sound, extract its features and apply them for pairing, copresence verification and authentication. In these techniques, ambient sound is first recorded and acoustic features are then extracted. Similarities or distances of the features, used by a classifier, are computed based on pre-defined metric. Comparing to other context such as Wi-Fi, GPS and physical environment context, audio offers several benefits. Speaker and microphone, required for producing and capturing audio, are available in most devices from computers to smartphones and wearable devices. Audio is efficient even with short recording times, only a few seconds compared to sometimes minutes for other context types such as Wi-Fi or GPS. Halevi et al. [9] proposed to use ambient audio for copresence verification of NFC devices. In their approach, ambient sounds recorded by a pair of devices in one second are compared to each other via their maximum cross correlation. In [22], Schürmann et al. proposed an approach where the devices extract an audio fingerprint as an energy matrix and compute the Hamming distance between their matrixes. Their method need at least 6 seconds of ambient audio to obtain efficient fingerprint. For automatic group membership maintenance, Tan et al. [27] assumed copresent devices form a group, and by checking similarity of silence signatures extracted from ambient sound, membership of a device can be continuously verified. Sound-Proof [10] also adopted ambient sound for two factor authentication.

Despite benefits of using audio in copresence verification, integrity of acoustic ambience is susceptible to adversarial manipulation [25]. Audio context attackers simply record audio, transmit it over a high speed channel and replay it in the victims vicinity. Furthermore, solutions in previous works largely depend on the quality of ambient sound since it is passively sensed by participating devices. In many cases where both devices are in quiet or in noisy environments, the verification result is not reliable. In addition, some approaches require to use raw audio data [9], [10], which may raise privacy concerns. A proposed method that uses only sound signatures [27] requires long recording duration to obtain enough features.

Physical environmental context such as temperature, humidity, gas, altitude etc. also have been studied for copresence verification. In their work [23] Shrestha et al. showed the feasibility of using such context.

RIR for localization. Room level localization techniques [19], [21], [32], [34] exploit the intrinsic acoustic properties of rooms extracted from room impulse response. These approaches exploit different types of acoustic features for room localization. In RoomSense [21] Rossi et al. showed that common audio features are superior to room acoustic features and their results were primarily based on them. SoundLoc [19] explored more acoustic features based on room impulse response including temporal features, spectral features and energetic features. EchoTag [32] and UbiK [34] use similar acoustic features based on characteristics of reflections of room impulse response in frequency domain to fingerprint locations.

Even though echolocation and RIR have been used for indoor localization and room fingerprint [19], [21], [32], [34], they have several limitations. First, whether RIR can be used for copresence verification has not been explored. Like other context-based copresence verification approaches, two devices in close proximity should perceive similar RIR. Second, previous works on room fingerprinting [19], [21] only show that the same device in different places can identify the place based on RIR, thus omitting hardware variance. For copresence verification we need to compare signals of two copresent devices. When two or more devices are involved, new challenges emerge such as hardware heterogeneity. Two different device models might behave differently even if they are in the same location. In our work, we elaborate such challenges and show that our system overcomes variance of hardware. In addition, room fingerprinting approaches require that the locations have been seen earlier. For copresence verification, this is an unreasonable assumption.

VIII. Conclusion

We have presented `DoubleEcho`, a copresence verification mechanism based on Room Impulse Response (RIR). To the best of our knowledge, `DoubleEcho` is the only context-based copresence verification mechanism that is robust against context-manipulation attacks. In `DoubleEcho`, one device emits a wide-band audible chirp and all devices record reflections of the chirp from the environment. Features extracted from the recordings are fed to a classifier to make a copresence decision. Since RIR is, by its very nature, dependent on the physical surroundings, an adversary may not replicate the same context for two devices at different locations. By means of an Android application and an experimental campaign we have shown that RIR-based copresence verification is feasible using commodity devices. Results show that `DoubleEcho` effectively mitigates context-manipulation attacks with a false positive rate as low as 0.089. While strengthening security, `DoubleEcho` shows a false negative rate in line with similar proposals. `DoubleEcho` operates in roughly 2 seconds and can be used with commodity devices. We expect that our proposal, experiments and evaluation results open opportunities for further applications relying on secure copresence verification.

ACKNOWLEDGMENT

This work was supported by the Intel Collaborative Research Institute for Secure Computing. This paper has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 779852. We thank Emmanuel Vincent (INRIA) for his initial discussion and for the Matlab source code to compute RIR; Jens Matthias Bohli (NEC) and Petteri Nurmi (University of Helsinki) for their insightful discussions and comments.

REFERENCES

[1] S. Brands and D. Chaum. Distance-bounding Protocols. In *Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, EUROCRYPT '93, pages 344–359, 1994.

[2] S. Capkun and J. P. Hubaux. Secure Positioning in Wireless Networks. *IEEE J.Sel. A. Commun.*, 24(2):221–232, 2006.

[3] C. Castelluccia and P. Mutaf. Shake them up!: a Movement-based Pairing Protocol for CPU-constrained Devices. In *Proc. 3rd international conference on Mobile systems, applications, and services*, MobiSys '05, pages 51–64, 2005.

[4] A. Czeskis, K. Koscher, J. R. Smith, and T. Kohno. RFIDs and Secret Handshakes: Defending Against Ghost-and-leech Attacks and Unauthorized Reads with Context-aware Communications. In *Proc. 15th ACM conference on Computer and communications security*, CCS '08, pages 479–490, 2008.

[5] D. Dolev and A. C.-C. Yao. On the Security of Public Key Protocols. *IEEE Transactions on Information Theory*, 29(2), 1983.

[6] S. Drimer and S. J. Murdoch. Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks. In *16th USENIX Security Symposium on USENIX Security Symposium*, SS'07, pages 7:1–7:16, 2007.

[7] A. Francillon, B. Danev, and S. Capkun. Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. In *NDSS*, 2011.

[8] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis. Practical NFC Peer-to-Peer Relay Attack using Mobile Phones. In *S6th international conference on Radio frequency identification: security and privacy issues*, RFIDSec'10, 2010.

[9] T. Halevi, D. Ma, N. Saxena, and T. Xiang. Secure Proximity Detection for NFC Devices Based on Ambient Sensor Data. In *European Symposium on Research in Computer Security (ESORICS)*, 2012.

[10] N. Karapanos, C. Marforio, C. Soriente, and S. Capkun. Sound-Proof: Usable Two-Factor Authentication Based on Ambient Sound. In *24th USENIX Security Symposium*, pages 483–498, 2015.

[11] J. Krumm and K. Hinckley. The NearMe Wireless Proximity Server. In *Ubiquitous Computing (UbiComp)*, 2004.

[12] J. Lester, B. Hannaford, and G. Borriello. Are You with Me? Using Accelerometers to determine if two Devices are carried by the same Person. In *2nd International Conference on Pervasive Computing*, pages 33–50, 2004.

[13] C. Marforio, N. Karapanos, C. Soriente, K. Kostiainen, and S. Capkun. Smartphones as Practical and Secure Location Verification Tokens for Payments. In *21st Annual Network and Distributed System Security Symposium (NDSS)*, 2014.

[14] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and Mandayam. ProxiMate: Proximity-based Secure Pairing using Ambient Wireless Signals. In *International Conference on Mobile Systems, Applications, and Services*, MobiSys '11, pages 211–224, 2011.

[15] R. Mayrhofer and H. Gellersen. RassShake Well Before Use: Intuitive and Secure Pairing of Mobile Devices. *Mobile Computing, IEEE Transactions on*, 8(6):792–806, 2009.

[16] M. Miettinen, N. Asokan, F. Koushanfar, T. D. Nguyen, J. Rios, A.-R. Sadeghi, M. Sobhani, and S. Yellapantula. I know Where You Are: Proofs of Presence Resilient to Malicious Provers. In *10th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2015)*, 2015.

[17] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh. Location Privacy via Private Proximity Testing. In *Network and Distributed System Security Symposium (NDSS)*, 2011.

[18] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. BeepBeep: A High Accuracy Acoustic Ranging System Using COTS Mobile Devices. In *5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 1–14, 2007.

[19] Z. C. R. Jia, M. Jin and C. J. Spanos. SoundLoc: Accurate Room-level Indoor Localization using Acoustic Signatures. *IEEE International Conference on Automation Science and Engineering*, abs/1407.4409, 2015.

[20] K. B. Rasmussen and S. Čapkun. Realization of RF Distance Bounding. In *Proceedings of the 19th USENIX Conference on Security*, pages 25–25, 2010.

[21] M. Rossi, J. Seiter, O. Amft, S. Buchmeier, and G. Tröster. RoomSense: An Indoor Positioning System for Smartphones Using Active Sound Probing. In *4th Augmented Human International Conference*, pages 89–95, 2013.

[22] D. Schurmann and S. Sigg. SSecure Communication Based on Ambient Audio. *IEEE Transactions on Mobile Computing*, 12(2):358–370, 2013.

[23] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan. Drone to the Rescue: Relay-Resilient Authentication using Ambient Multi-Sensing. In *18th International Conference on Financial Cryptography and Data Security*, 2014.

[24] B. Shrestha, N. Saxena, H. T. T. Truong, and N. Asokan. Sensor-based Proximity Detection in the Face of Active Adversaries. *IEEE Transactions on Mobile Computing*, PP:1–1, 05 2018.

[25] B. Shrestha, M. Shirvanian, P. Shrestha, and N. Saxena. The Sounds of the Phones: Dangers of Zero-Effort Second Factor Login Based on Ambient Audio. In *ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 908–919, 2016.

[26] P. Shrestha and N. Saxena. Listening Watch: Wearable Two-Factor Authentication using Speech Signals Resilient to Near-Far Attacks. In *ACM Conference on Security and Privacy in Wireless and Mobile Networks*. ACM, 2018.

[27] W.-T. Tan, M. Baker, B. Lee, and R. Samadani. The Sound of Silence. In *11th ACM Conference on Embedded Networked Sensor Systems*, SenSys '13, pages 19:1–19:14, 2013.

[28] N. O. Tippenhauer, K. B. Rasmussen, C. Pöpper, and S. Čapkun. Attacks on Public WLAN-based Positioning Systems. In *7th International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, 2009.

[29] J. Toivonen, H. Truong, and T. Nguyen. RIR samples Darmstadt and Helsinki, Summer-Autumn 2018, Sept. 2018.

[30] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, and P. Nurmi. Comparing and Fusing different Sensor Modalities for Relay Attack Resistance in Zero-Interaction Authentication. In *IEEE International Conference on Pervasive Computing and Communications, PerCom 2014*, 2014.

[31] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, and P. Nurmi. Using Contextual Co-presence to Strengthen Zero-Interaction Authentication: Design, Integration and Usability. *Pervasive and Mobile Computing*, 16:187 – 204, 2015.

[32] Y.-C. Tung and K. G. Shin. EchoTag: Accurate Infrastructure-Free Indoor Location Tagging with Smartphones. In *21st Annual International Conference on Mobile Computing and Networking*, MobiCom '15, pages 525–536, 2015.

[33] A. Varshavsky, A. Scannell, A. LaMarca, and E. De Lara. Amigo: Proximity-Based Authentication of Mobile Devices. In *International Conference on Ubiquitous Computing (UbiComp)*, 2007.

[34] J. Wang, K. Zhao, X. Zhang, and C. Peng. Ubiquitous Keyboard for Small Mobile Devices: Harnessing Multipath Fading for Fine-grained Keystroke Localization. In *12th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '14, pages 14–27, 2014.

[35] Wilson S Melo Jr, Raphael C S Machado, and Luiz F R C Carmo. Using Physical Context-Based Authentication against External Attacks: Models and Protocols. 1:1, 1 2018.

[36] K. C. Zeng, Y. Shu, S. Liu, Y. Dou, and Y. Yang. A Practical GPS Location Spoofing Attack in Road Navigation Scenario. In *18th International Workshop on Mobile Computing Systems and Applications*, HotMobile '17, pages 85–90, 2017.