Privacy-Enhancing Context Authentication from Location-Sensitive Data

Pradip Mainali
pradip.mainali@onespan.com
OneSpan
Brussels, Belgium

Carlton Shepherd
carlton.shepherd@onespan.com
OneSpan
Cambridge, United Kingdom

Fabien A. P. Petitcolas
fabien.petitcolas@onespan.com
OneSpan
Brussels, Belgium

ABSTRACT

This paper proposes a new privacy-enhancing, context-aware user authentication system, ConSec, which uses a transformation of general location-sensitive data, such as GPS location, barometric altitude and noise levels, collected from the user's device, into a representation based on locality-sensitive hashing (LSH). The resulting hashes provide a dimensionality reduction of the underlying data, which we leverage to model users' behaviour for authentication using machine learning. We present how ConSec supports learning from categorical *and* numerical data, while addressing a number of on-device and network-based threats. ConSec is implemented subsequently for the Android platform and evaluated using data collected from 35 users, which is followed by a security and privacy analysis. We demonstrate that LSH presents a useful approach for context authentication from location-sensitive data without directly utilising plain measurements.

CCS CONCEPTS

• **Security and privacy → Authentication**; *Privacy-preserving protocols*; Mobile and wireless security.

KEYWORDS

Location Privacy, Context Authentication, Mobile Security

ACM Reference Format:
Pradip Mainali, Carlton Shepherd, and Fabien A. P. Petitcolas. 2019. Privacy-Enhancing Context Authentication from Location-Sensitive Data. In *Proceedings of the 14th International Conference on Availability, Reliability and Security (ARES '19), August 26–29, 2019, Canterbury, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3339252.3340334

1 INTRODUCTION

Secure and usable mobile authentication mechanisms are critical to numerous applications, including internet banking, email, and social media. Traditional knowledge-based authentication mechanisms, such as PINs, patterns and passwords, remain fraught with concerns surrounding memorability, credential reuse, sufficient entropy, and shoulder-surfing attacks [20, 27]. Meanwhile, token-based methods, e.g. one-time password (OTP) generators and key fobs, can be lost or stolen and are generally costly to produce, maintain and replace [18]. Biometrics, such as fingerprint and facial authentication, have become widely-deployed on modern mobile handsets; however, environmental factors, like moisture, injury, skin complexion, and lighting can significantly increase error rates [6]. Traditional approaches also feature an overarching drawback: the user is authenticated only once, after which all access to assets and services and granted thereafter. Such all-or-nothing authentication has prompted both security and usability concerns in related literature [11, 13, 14, 22].

Context authentication systems have arisen from these concerns, which transparently compute an authentication score from device data, such as GPS location, nearby Wi-Fi access points (APs), and cellular network information, with respect to previously observed behaviour [11, 14, 17, 25]. Such systems may infer the user's authentication status directly, i.e. accept/reject [25], or determine access control policies and explicit authentication strength [14, 17]. Unfortunately, many context authentication systems necessitate collecting swathes of privacy-sensitive data. This is especially problematic if the scheme is deployed by a remote authentication service and the data is later disclosed in a security breach or misused without consent. Several surveys have already demonstrated users' reticence in disclosing location data used by many context authentication proposals [3, 5, 9]. Fisher-Short et al. [9] show, for instance, that users tend to deny mobile applications the permission to access location data because of concerns about it being transmitted remotely and shared surreptitiously with third-parties.

In light of this discussion, we present a novel context authentication system, ConSec, which enhances the confidentiality of device data using the Super-Bit Locality-Sensitive Hashing (SB-LSH) proposal by Ji et al. [15]. SB-LSH is a dimensionality reduction algorithm wherein the locality-sensitive hashes reveal only the relative distance between user location measurements without disclosing their precise location on Earth; from this, machine learning can be applied to model the user's behaviour. In this work, we demonstrate how ConSec enables computation over numerical and categorical data, e.g. Wi-Fi ESSIDs, to flexibly support various modalities for privacy-enhancing contextual authentication. We evaluate ConSec using data collected from 35 participants, after which we analyse its effectiveness against certain attacks, such as triangulation, and their effect on error rates.

The main contributions of this paper is a new approach to contextual authentication by protecting the confidentiality of numerical location-sensitive data, such as GPS coordinates, using SB-LSH [15]. ConSec can learn user authentication models from protected numerical and categorical data using standard machine learning algorithms with low error rates. This paper begins with a review of

ARES '19, August 26–29, 2019, Canterbury, United Kingdom

ACM ISBN 978-1-4503-7164-3/19/08…$15.00
https://doi.org/10.1145/3339252.3340334

existing contextual and privacy-enhancing continuous authentication schemes in Section 2. Section 3 then describes the architecture of the ConSec, including the threat model and the modalities used. Sections 4 and 5 describe the use of LSH and keyed HMACs for privacy-preserving behavioural modelling from numerical and categorical data respectively for contextual authentication. Next, an evaluation of the proposed scheme is presented from a user base of 35 users in Section 6, which is followed by a security and privacy analysis in Section 7. Lastly, Section 8 concludes this paper, including a discussion of future research directions.

2 RELATED WORK

This section explores the state-of-the-art of *contextual authentication*, focused on in this work, and *privacy-enhancing continuous authentication* generally. We summarise notable proposals and their contributions.

2.1 Contextual Authentication

The CASA system by Hayashi et al. [14] focuses on reducing the strength of user-authentication challenges based on the current location of the user and previously observed behaviour. The authentication strength is determined by the presence of the user in location of varying perceived risk, such as at home or at work. CASA applies a Naïve Bayes classifier to model user behaviour from GPS location, which produces a probability value based on their past behaviour to determine the device's explicit authentication strength, e.g. password, PIN or even no challenge. The system is trialled using 32 users, with a location classification accuracy of 92%, and 68% of explicit authentication attempts being reduced.

SenGuard by Shi et al. [25] uses five modalities—location, cell tower ID, voice, touch, and motion-based activity recognition (cycling, walking, stationary, etc.)—from which a multitude of features are extracted, including touch gestures, GPS correlation, and Levenshtein distance of cell IDs in a sliding window. From this, the system computes an aggregated authentication decision from individual, modality-specific classifiers, which is used to authenticate the user in a binary fashion, i.e. accept or reject. SenGuard is evaluated with a user base of four participants, yielding a classification accuracy of 95.8–97.1% depending on the user.

Gupta et al. [11] present the first proposal for setting mobile device access control policies based on users' contexts. The authors use features based on Context of Interest (CoI), corresponding to locations a user visits frequently or spends significant amounts of time in. CoIs are generated from clusters of GPS coordinates, Wi-Fi access points (APs) and nearby Bluetooth devices, which are compared with past data using similarity metrics, such as the set intersection, using manually set threshold values. The CoIs are then divided into 'safe' and 'unsafe' locations, such as 'at home' and 'out shopping' respectively, which are used for dynamically setting access control policy profiles. An evaluation with 37 users resulted in average precisions of 0.854 (safe locations) and 0.311 (unsafe), and recalls of 0.917 (safe) and 0.341 (unsafe).

Miettenen et al. [17] present ConXsense, which uses contextual data for setting risk-based device locks, akin to [14], as well as access control policy profiles. The authors also generate CoI features, but, unlike [11], these are inputted to a classifier with ground truth labels drawn from user input. The authors evaluate the system using data from 15 users and Random Forest, k-Nearest Neighbour (kNN), and Naïve Bayes classifiers, with an approximate 0.70 true positive rate (TPR) and 0.10 false positive rate (FPR).

Witte et al. [30] propose a system for scoring authentication behaviour based on the device's GPS location, accelerometer, magnetic field, light, battery and sound measurements. Statistical features are extracted from the raw measurements, such as the arithmetic mean, median, maximum and minimum values, which are aggregated with system-level features, such as screen status and boot and shutdown times. Feature vectors are inputted to an SVM classifier trained on user data collected over a three-day period; an evaluation is performed using data from 15 participants, with an average F1-score of 0.85.

2.2 Privacy-Enhancing Continuous Authentication

Shahandashti et al. [23] propose a scheme for computing authentication decisions remotely on an honest-but-curious server from encrypted feature vectors using Paillier homomorphic encryption [19]. Encrypted behavioural models are stored on a remote server and a decision is computed homomorphically from encrypted feature vectors transmitted from the user's device without exposing the plaintext measurements. The decision is computed from the dissimilarity between incoming features and the stored user profile using the average absolute distance (AAD); the user is authenticated if the AAD similarity falls within a threshold determined by the service provider. However, while a security proof is provided, the scheme is neither implemented nor evaluated in practice.

Domingo-Ferrer et al. [8] develop the work from [23] by presenting an additional approach based on the set intersection of homomorphically encrypted feature vectors to authenticate users. The set intersection is used to compute the dissimilarity function between the encrypted user profile and incoming feature vectors to support categorical features, beyond only numerical features supported in [23]. The authors provide a performance evaluation in which authentication takes 0.08–31.2 seconds depending on the number of input features (1–50 features respectively).

Sedenka et al. [28] construct protocols for the outsourcing of the scaled Manhattan (L1) and Euclidean (L2) distances and principal component analysis (PCA) using garbled circuits and homomorphic encryption for use in continuous authentication. The work tackles the case of computing the similarity of incoming feature vectors with stored user models in the presence of an honest-but-curious server. The authors provide a security analysis and performance evaluation using a consumer laptop and smartphone, the results of which yield a communication and time penalty of 4–174MB and 0.85s–45.9s respectively based on the submitted feature vector size.

Halunen and Vallivaara [12] present a privacy-enhancing continuous authentication system based on keystroke dynamics with order-preserving symmetric encryption (OPSE) and, like [23], Paillier homomorphic encryption. The proposal extracts four features regarding the down-down, up-down and down-up times of each keystroke, along with the entered string. Next, the AAD is computed homomorphically between the sample and the stored template values, while OPSE is used for comparing samples with various

Table 1: Related contextual authentication schemes.

Proposals	Data Modalities	Error Rates
CASA [14]	GPS Location	92% Acc.
SenGuard [25]	GPS Location, Voice, Touch, AR	95.8–97.1% Acc.
Gupta et al. [11]	GPS Location, Wi-Fi APs, Bluetooth Devices	0.311–0.854 Pr., 0.341–0.917 Re.
ConXsense [17]	GPS Location, Wi-Fi APs, Bluetooth Devices	~70% TPR, 10% FPR
Witte et al. [30]	GPS Location, Sys., AC, MF, Sound, Light	~0.85 F1-score

AC: Accelerometer, MF: Magnetic Field, Sys.: System Data, Acc.: Classification Accuracy, FPR: False Positive Rate, TPR: True Positive Rate, AR: Activity Recognition, Pr.: Precision, Re.: Recall.

thresholds to fine-tune security and usability. An evaluation is conducted with 20 users yielding a best-case accuracy of 91.5%.

2.3 Discussion

Many existing contextual authentication proposals, summarised in Table 1, use device measurements in unprotected form—giving rise to privacy concerns especially when authentication decisions are outsourced to a remote server [8, 23, 28]. As noted previously, studies have already shown that users are generally reluctant towards disclosing, in particular, their GPS location to device applications [3–5, 9]. Some users also disable GPS depending on the sensitivity of the location they visit, irrespective of the application [3]. Work in [9] also shows that, when permission is denied to an application that accesses GPS location, it is principally from concerns relating to whether location data is stored securely (on-device and remotely) and whether it is shared with other parties without consent.

Current approaches to privacy-enhancing continuous authentication have employed homomorphic encryption, e.g. Paillier [19], and garbled circuits for two-party computation (2PC) [8, 12, 23, 28]. However, challenges still remain with respect computational complexity and storage overhead, which is exemplified by worst cases of 31.2s and 45.9s to compute authentication decisions in [8] and [28] respectively. Storage complexity is also an issue, with 4–175MB required for computing a single authentication decision in [28]. Homomorphic encryption-based schemes also face inherent challenges regarding the supported arithmetic operations for learning from data, i.e. *only* additions for Paillier-based proposals. Fully homomorphic encryption (FHE) [10], meanwhile, has been deemed too cumbersome in the literature for high-frequency, high-dimensional classification tasks like continuous authentication [23, 28].

Lastly, we note that the work in [12] focuses on keystroke-based authentication, which necessitates user interaction. Rather, the focus of this work is context authentication from device sensors *without* interaction from the user. Ultimately, we aim to demonstrate a simpler approach to privacy-enhancing, zero-interaction context authentication based on keyed HMACs and the Super-Bit Locality-Sensitive Hashing (SB-LASH) algorithm by Ji et al. [15], without resorting to the complexities of homomorphic encryption and multi-party computation used in existing solutions.

3 SYSTEM ARCHITECTURE

In this section, we present the system architecture of ConSec for performing context authentication, beginning with a high-level overview before describing the threat model and data modalities considered in this work.

3.1 System Overview

Figure 1 illustrates the block diagram of ConSec, comprising the ConSec-App on the smartphone for data collection and transmission, and the authentication algorithm (ConSec-Auth) that executes remotely on the authentication server. ConSec-App transforms *categorical* contextual data, such as nearby Wi-Fi ESSIDs and MAC addresses, using HMAC-SHA256 under a randomly generated, per-user key within the application, which we denote HK. *Numerical* location-sensitive data, such as the device's GPS location and barometric altitude, is transformed using the Super-Bit LSH (SB-LSH) algorithm proposed in [15]. The transformed HMAC and SB-LSH values are transmitted to ConSec-Auth, which computes the authentication score based on the user's previously observed behaviour. We return to the complete set of contextual data collected by ConSec-App in Section 3.3.

Figure 1(b) shows the block diagram of the ConSec-Auth algorithm, comprising enrollment and authentication phases. We build on the assumption that contextual data generally form clusters around locations frequently visited by the user, such as home, work, and so on, according to their day-to-day mobility patterns. The enrollment phase constructs the initial model that represents these mobility patterns, which is employed in the authentication phase to compute a real-valued authentication score of the user's newly observed behaviour, $\mathcal{S} \in [0, 1]$, to be used by the service provider.

3.2 Assumptions and Threat Model

ConSec aims to provide lightweight, server-side context authentication for informing secure access to remote assets from mobile sensor data. The scheme is intended to allow a service provider to tailor access to sensitive assets based on the output probability without incurring the complexities of existing techniques, such as homomorphic encryption [8, 12, 23, 28].

Server-side, we assume a remote service operating under the honest-but-curious model, which executes the scheme as intended but attempts to infer users' behaviour based on the device data samples that it observes. The service may use any insights for unintended purposes, such as profiled advertising or selling the information to unauthorised third-parties without consent. That is, the goal is to provide context authentication for informing server-side access control while precluding the service's ability to easily recover the actual measurements it receives. However, we note that, even under model whereby the server is wholly honest, the use of privacy-preserving context authentication also limits the impact of security breaches that lead to the unauthorised disclosure of plaintext measurements, and the publicity or legal repercussions that may follow. We also assume that the ConSec mobile application is available in a trusted application store.

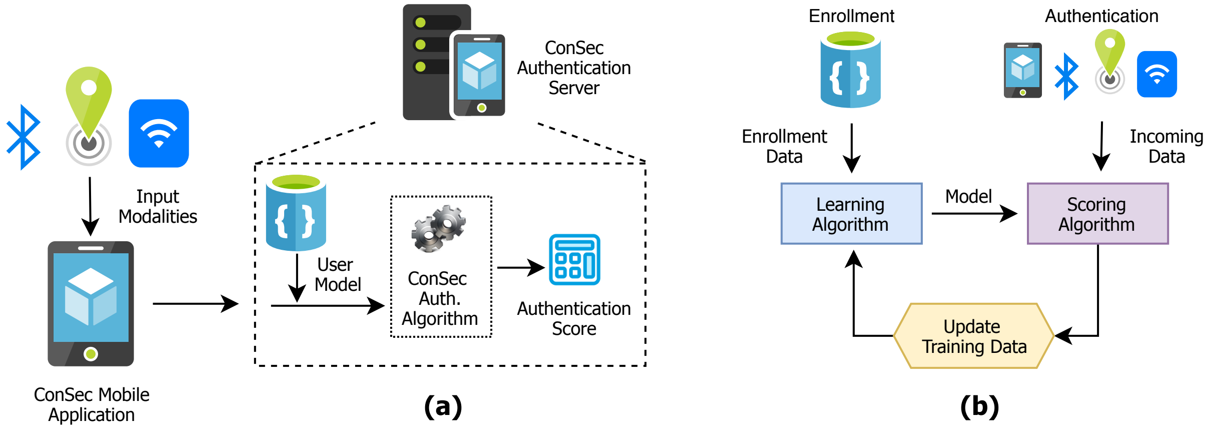

Figure 1: High-level architecture: (a) ConSec context-aware authentication flow; (b) ConSec authentication algorithm.

On the device, we assume a trusted keystore and the ability to collect, transform and transmit sensor values securely. The keystore is used to hold certificates used to mutually authenticate the end-points for transmitting feature data across a standard network channel, e.g. Wi-Fi, securely using TLS. Both components may be instantiated using conventional security controls provided by modern mobile operating systems, such as application sandboxing, which we concentrate on this work. However, for further security assurances against kernel-mode (ring 0) adversaries, this may be realised using a trusted execution environment (TEE), as suggested in [16] and [24]. We also consider a context-manipulating adversary, e.g. [26], with the ability to influence the measurements collected by the device; for example, by instantiating rogue Wi-Fi APs with spoofed ESSIDs and MAC addresses to maliciously influence the authentication scoring algorithm. This forms the basis of the evaluation described later in Section 6.

3.3 Contextual Data

Modern mobile devices contain a varieties of sensors, such as GPS chips, magnetometers, accelerometers and pressure sensors, and related modules, e.g. Wi-Fi and Bluetooth, that can be used for collecting contextual data. The data types used by ConSec are described in the following subsections.

3.3.1 Geographic Location. The geographic location of the user is collected from the device's GPS module or, if it is unavailable, using Android's network location, which returns recently observed GPS coordinates, or coordinates based on the Cell ID or Wi-Fi network location in that order [1]. The geodetic latitude (ϕ) and longitude (λ) values are first transformed to the Earth-Centre-Earth-Fixed (ECEF) Cartesian coordinate system accounting for the Earth's ellipsoidal shape using the WGS84 model [7]. Equation 1 is used to transform the geodetic latitude ϕ, longitude λ and altitude h cordinates to the ECEF coordinate system. The coordinates of a point $\mathbf{p}$ in Cartesian coordinates on Earth's surface are given by:

$$\begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} = \begin{pmatrix} (N(\phi) + h)\cos(\phi)\cos(\lambda) \\ (N(\phi) + h)\cos(\phi)\sin(\lambda) \\ ((b^2/a^2)N(\phi) + h)\sin(\phi) \end{pmatrix} \tag{1}$$

where

$$N(\phi) = \frac{a}{\sqrt{(1 - e^2\sin^2(\phi))}} \tag{2}$$

and the squared first-eccentricity (e^2), the semi-major axis (a) and the semi-minor axis (b) are taken from WGS84: $e^2 = 6.69437999014 \times 10^{-3}$, $a = 6378137$m and $b = 6356752.3142$m.

3.3.2 Barometric Altitude. Barometric pressure provides information regarding the height of the device from the sea level as atmospheric pressure is proportional to altitude; it is also measurable within enclosed spaces, e.g. buildings, where GPS coordinates may not be ascertained. The barometric altitude is calculated from the measured pressure, p, reference pressure, p_0, and temperature, T_0, using the following formula from [29]:

$$H = \frac{273.15 + T_0}{0.0065}\left(1 - \left(\frac{p}{p_0}\right)^{\frac{1}{5.255}}\right) \tag{3}$$

ConSec-App uses METAR[1] (Meteorological Aerodrome Report) for acquiring p_0 and T_0.

3.3.3 Noise Level. The average amplitude of the background noise is estimated using a three-second recording from the device's microphone. This provides additional contextual information regarding the user's location, which is also used in [30] as an authentication feature; for intuition, the device will likely measure a lower amplitude in a quiet office versus a loud city-centre environment.

3.3.4 Magnetic Fingerprint. This comprises the geomagnetic field strength and the geomagnetic inclination angle computed from the device's magnetometer and accelerometer sensors. Magnetic field data varies on the Earth's surface—it is strongest at the poles and weakest at the equator—and thus provides some information about the user's location. The magnetic inclination angle is the angle at which the magnetic field lines intersects with the surface of the Earth; it ranges from zero degrees at the equator to 90 degrees at the poles. The accelerometer data is used to find the orientation of the device with respect to the world coordinate system, and the field data read by the device's magnetometer is transformed to

[1] METAR: https://www.aviationweather.gov/dataserver

align with respect to the world coordinate system from which the magnetic inclination angle is computed.

3.3.5 *Wi-Fi.*

The data collected from Wi-Fi networks includes the name and the MAC address of the router to which the phone is connected, the primary and secondary DNS settings of the router, the received signal strength indication (RSSI), and the list of Wi-Fi names and MAC addresses of nearby access points (APs).

3.3.6 *Mobile Network.*

The data collected from a mobile network comprises the mobile network type—2G, 3G, and so on—RSSI, operator name, mobile network code (MNC), mobile network country code (MCC), location area code (LAC), and mobile network cell ID.

3.3.7 *Bluetooth.*

The Bluetooth module of ConSec acquires a list of the (visible) Bluetooth device names and MAC addresses detected within the device's vicinity.

4 ENHANCING THE PRIVACY OF CONTEXTUAL DATA

Evidently, location-sensitive data carries private information regarding the user's whereabouts that could be exploited, particularly when authentication decisions are computed remotely. This is a major shortcoming of many existing contextual authentication schemes identified in Section 2. As such, to enhance the privacy of the location-sensitive information listed previously, *numerical* contextual data is subject to the SB-LSH algorithm by Ji et al. [15], while keyed HMACs are used for *categorical* data. The following sections describe these procedures in further detail.

4.1 Numerical Data

The SB-LSH algorithm [15] is computed from seven features: the geographic location $\mathbf{p} = (p_x, p_y, p_z)$ from Section 3.3.1; the barometric altitude; the noise level; the magnetic field strength; and the magnetic inclination angle. This information is salted with a long-lived, randomly generated, per-user value, s, which is used to create an eight-dimensional vector, $\mathbf{x}$.

SB-LSH is initialised by generating K eight-dimensional vectors $[\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_K}]$ sampled from the normal distribution $\mathcal{N}(0,1)$ and orthogonalised in blocks of eight vectors using the Gram-Schmidt process. The vector, $\mathbf{x}$, is then projected to produce $h_{\mathbf{v_i}}(\mathbf{x}) = \text{sign}(\mathbf{v_i}^T\mathbf{x})$, where sign(.) is defined as:

$$\begin{aligned} \text{sign}(z) =& 1, z \geq 0 \\ & 0, z < 0 \end{aligned} \tag{4}$$

This results in a K-bit hash code $H(\mathbf{x}) = \left\{h_{\mathbf{v_1}}(\mathbf{x}), h_{\mathbf{v_2}}(\mathbf{x}), \ldots, h_{\mathbf{v_K}}(\mathbf{x})\right\}$. It is also shown in [15] that the Hamming distance between two hash codes, a and b, is related to the angular distance, $\theta_{a,b}$:

$$E\left[d_{\text{Hamming}}\left(h(a), h(b)\right)\right] = \frac{K\theta_{a,b}}{\pi} = C\theta_{a,b} \tag{5}$$

Where $C = K/\pi$ is constant. As such, the similarity (cosine distance) between the hash values can be computed. The K randomly initialized LSH vectors are used to compute the location-hash. An issue then arises pertaining to setting the value of K, which depends on the security strength and the accuracy that is needed in computing the distance from the location-hash. In both cases, a large value of K is desired; however, on the other hand, a large K increases the amount of data needed to be transmitted and stored on the server. A value of K that gives a reasonable error in computing the distance without compromising security is hence needed.

Table 2: Error variation for K bits in LSH.

K bits	MAE	RMSE
128	41.908	65.709
256	35.132	44.051
512	26.654	32.236
1024	17.655	23.181
2048	13.355	17.049
4096	9.785	12.648
8192	7.568	9.722
16384	6.160	7.874

We evaluated this for different values of K as follows. One thousand different location pairs were generated randomly, where the first location value in the pair was generated randomly and the second was generated to be 25km away from the first. 25km was chosen as a preliminary value as we assume that general user mobility patterns are limited geographically to small regions, say, for work and home. The true and approximate central angles between the locations in the pair were computed from the actual location values and the location hash codes respectively. The experiment was repeated for ten thousand different SB-LSH hash codes. Table 2 shows the mean absolute error (MAE) and root-mean-square error (RMSE) for different K values, averaged over all hash code instances. The error in computing the distance between the locations can be computed as the central angle between the locations is known, which is computed using the great circle distance formula as follows:

$$d = 2 \times R \times \sin\left(\frac{\theta}{2}\right) \tag{6}$$

Where θ is the central angle between the actual locations in radians and $R = 6371$ km is the mean radius of Earth. The error in computing the distance from location-hashes is given by:

$$d_e = 2 \times R \times \left(\sin\left(\frac{\theta_t}{2}\right) - \sin\left(\frac{\theta_a}{2}\right)\right) \tag{7}$$

Where θ_a is approximate central angle between the locations in radians computed from the location-hashes and θ_t is true central angle computed from actual location values. Table 2 shows that both the MAE and RMSE reduces with larger K values. The error is larger than the actual distance of 25km for K smaller than 2048. In this paper, we fix $K = 4096$ for our experiments in Section 6.

4.2 Categorical Data

For categorical data types, HMAC-SHA256 is computed upon each data sample, which is keyed under a 128-bit long-lived key, HK, that is generated randomly upon initialisation by the ConSec mobile application. HK is unique to each application and should, ideally, be generated and stored securely in the device's keystore. The following section describes how HMAC and SB-LSH protected data is used by the ConSec authentication algorithm.

Table 3: ConSec contextual data modalities.

Contextual Modality	Feature Type	Data Type
Location-hash	†	N
Wi-Fi state	OHE	C
Wi-Fi router MAC	OHE	C
Wi-Fi SSID	OHE	C
Wi-Fi IP	OHE	C
Wi-Fi network ID	OHE	C
Wi-Fi RSSI	V	N
Wi-Fi frequency	OHE	C
Wi-Fi router IP	OHE	C
Wi-Fi router DNS-1	OHE	C
Wi-Fi router DNS-2	OHE	C
List of Wi-Fi names	OHE	C
List of Wi-Fi MACs	OHE	C
List of Wi-Fi frequencies	OHE	C
SIM state	OHE	C
Network data state	OHE	C
Network data type	OHE	C
Network RSSI	V	N
Network operator name	OHE	C
Network MCC MNC	OHE	C
Network LAC	OHE	C
Network Cell ID	OHE	C
Bluetooth device names	OHE	C
Bluetooth device MACs	OHE	C
Day index	OHE	C
Time	V	N

†: Cosine similarity with six references, V: real value, OHE: one-hot encoded, N: numerical, C: categorical.

5 AUTHENTICATION ALGORITHM

This section details how users' authentication models are learned from SB-LSH and HMAC-applied contextual data.

5.1 Feature Extraction

Table 3 lists the 32 contextual data types employed by ConSec. The LSH data is pre-processed to compute the feature wherein the cosine similarity values are computed with respect to a reference value collected during the enrollment phase. The modal (most frequent) LSH value from the enrollment samples is used as this reference. For categorical data, one-hot encoded (OHE) features are created to map categorical values to numerical representations; OHE results in a sparse matrix as an output, where each column corresponds to one possible value of the data. Real-valued data types, indicated in Table 3, are used directly. We note that feature vectors are standardised to zero mean and unit variance once computed.

5.2 Learning User Authentication Models

Computed features are subsequently inputted to a k-means clustering algorithm, which is initialised using the k-means++ method by Arthur and Vassilvitskii [2]. Clustering was chosen from the assumption, based from related work [11, 14, 17], that users' contextual data has a tendency to form clusters according to their mobility patterns, such as regularly frequenting places of work and home.

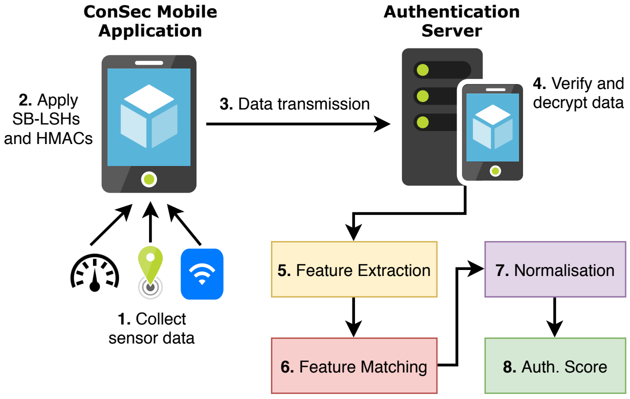

Figure 2: Information flow for user authentication.

We are interested in detecting the centroid of these clusters, thus k-means was used, but this necessitates some number of clusters, k, to be specified in advance. As such, the algorithm is executed for different values of $k = [5, 10, 15, 20, 25]$, stopping for a value of k if the rate of the clustering error falls below a convergence threshold, ϵ_1. In the experiments, a threshold of $\epsilon_1 = 0.05$ (5%) was used. Additionally, we prune clusters with population density lower than a separate threshold, ϵ_2. We fix $\epsilon_2 = 0.025$ (2.5%) in our experiments so that clusters with densities below this, i.e. very rarely or transiently frequented, are removed from the user model.

5.3 Authentication Procedure

As shown in Figure 1b, ConSec-Auth is ready to enter the authentication phase once the user models are created. This comprises an eight step process illustrated in Figure 2 and described as follows:

1. **Data Collection**. Data is sampled by ConSec-App at 10 minute intervals using the modalities listed in Table 3.
2. **Device Pre-Processing**. Next, ConSec-App applies SB-LSHs and HMACs to numerical and contextual modalities respectively, as described in Sections 4.1 and 4.2.
3. **Data Transmission**. The SB-LSH and HMAC values are transmitted over a secure channel between ConSec-App and ConSec-Auth using TLS.
4. **Verification and Decryption**. The server verifies and recovers the device feature vectors from the secure channel.
5. **Feature Extraction**. This follows the procedure described in Section 5.1 to produce a feature vector, $\mathbf{f}$.
6. **Feature Matching**. ConSec-Auth computes the Euclidean (L2) distance between $\mathbf{f}$ and the cluster centroids of the user's model. The smallest distance (d) is selected.
7. **Normalisation**. The smallest distance d is normalized as $\frac{d}{q}$, where q is a normalizing constant. The score is computed as $\mathcal{S} = 1 - \frac{d}{q}$.
8. **Score Output**. The authentication score, $\mathcal{S} \in [0, 1]$, is returned as output.

5.4 Model Refreshing

Some contextual data may change over time due to behavioural shifts in the users' mobility patterns; for example, after joining a new workplace or moving house in a new town/city where the old locations have little relevance. This applies not only to changes in the users' GPS locations, but also the lists of detected Wi-Fi APs, nearby Bluetooth devices, background noise, and other modalities; users' authentication data models should be updated regularly to reflect these changes. In this work, a weekly interval was chosen as a default to re-evaluate behavioural shifts; however, the model of the previous week is retained if 33.3% of the contextual data is predicted with higher accuracy (assuming the user spends at least 8 out of 24 hours at a regular location, e.g. home or office).

6 EXPERIMENTAL RESULTS

In this section, experiments are presented for evaluating the robustness of ConSec, which were conducted using data collected from 35 participants.

6.1 Data Collection

Participation emails were sent to 250 employees within a European-based US technology company, resulting in 35 users who enrolled in the trial; no incentives were offered for participation. Participants were requested to install and launch the ConSec application on their primary smartphone, after which no further interaction was required; users were able to start/stop data collection at will and withdraw at any time by uninstalling the application. Prior to this, participants were informed about types of data collected; that it would be stored alongside an anonymous, randomly-generated user ID; and an explanation of how the collected data itself was protected on the server. Supplementary material was also distributed via email. The data collection period lasted five weeks.

Upon installation, the application randomly generated unique SB-LSH parameters and *HK* for the keying HMACs, which were held only on the device. The application collected contextual data at 10 minute intervals, which is encrypted under AES in GCM mode using a key, *AK*, generated randomly for each message. *AK* is then encrypted using asymmetric encryption (RSA-2048) using the ConSec certified public-key, *PK*, contained within the application, and transmitted to the server which possesses the private portion. Both the data and the encrypted *AK* are transmitted as a single message, *m*, over TLS to the authentication server. This is listed in Equations 8 and 9. This process was conducted to protect participants' data at rest on our authentication server.

$$Data = \text{AES-GCM}_{AK}\Big\{ \begin{aligned} &\text{LSH}(p_x, p_y, p_z, \text{Alt.}, \text{Noise}, \dots), \\ &\text{HMAC}_{HK}(\text{Wi-Fi}_{\text{MAC}}), \\ &\text{HMAC}_{HK}(\text{Wi-Fi}_{\text{ESSID}}), \dots \Big\} \end{aligned} \tag{8}$$

$$m = \big\{Data,\ \text{RSA-Enc}_{PK}(AK)\big\} \tag{9}$$

To minimise battery consumption, the ConSec-App placed sensors into low-power mode due to the relatively low sampling frequency (once every 10 minutes). Preliminary experiments on a single device showed that ConSec-App consumes a negligible (~1%) amount of power, indicated by the power management metrics provided by Android OS on our test device (Samsung Galaxy S8). The encrypted contextual data was uploaded to a server on a weekly basis over TLS with an anonymous user ID generated by ConSec-App. The encrypted data was stored on a private corporate machine, with access granted only to the authors of this work.

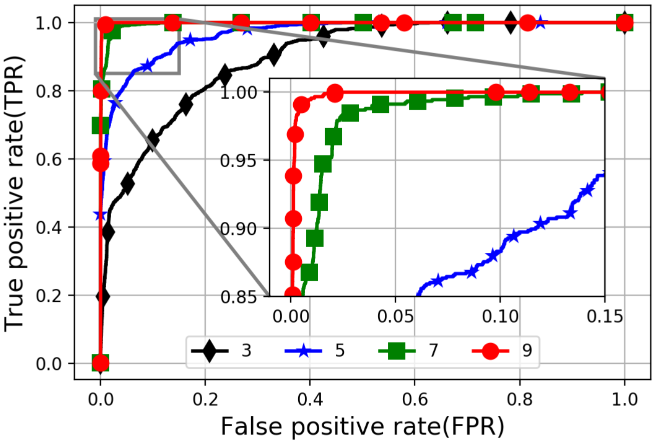

Figure 3: ROC curves for (a) 3, (b) 5, (c) 7 and (d) 9 numbers of the modified contextual data modalities (averaged across all participants).

6.2 Robustness to Inaccurate Inputs

Unlike previous approaches that rely upon *supervised* learning, e.g. CASA [14], ConXsense [17] and SenGuard [25], whereby the algorithm is trained using labels collected from user input, ConSec is underpinned by clustering, i.e. unlabelled *unsupervised* learning. Evaluating the performance of clustering algorithms is non-trivial compared with computing the precision, recall, or similar metric drawn from counting the number of true/false positives/negatives used in previous work [21].

In this paper, we began our evaluation by measuring the robustness of ConSec to inaccurate values sourced from varying numbers of data modalities. This is to evaluate its robustness to values modified by a contextual adversary listed in Section 3.2. To this end, Receiver Operating Characteristic (ROC) curves were plotted for the True Positive Rate (TPR) and False Positive Rate (FPR) to evaluate the detection of compromised samples versus unmodified ones.

For this, we selected the 100 closest feature vectors to the cluster centroids using the L2-distance from the training sets of each user. These samples are labelled as positives that should be accepted by the system. Next, *n* feature fields of these samples are modified—using the procedure described below—and labelled as negatives that ought to be rejected by the system. This process was repeated for 1000 distinct trails for various combinations of $n = [3, 5, 7, 9]$ modalities and across all users.

To modify the categorical data fields, such as Wi-Fi ESSIDs, the HMAC values (base 64-encoded) were modified to a random base 64 string of the same length. For numerical data, the SB-LSH values were modified to a random binary string, also of the same length. Figure 3 shows the ROC curves using vectors with three, five, seven and nine contextual modalities. Our results show that the system approaches an ideal system when seven and nine feature

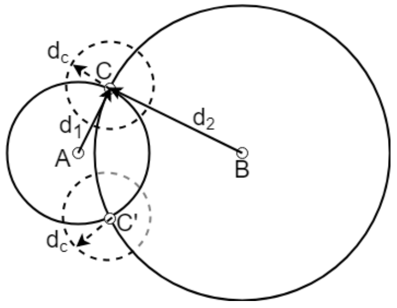

Figure 4: Triangulation to derive unknown location, C, from two known locations, A and B. Only distances from SB-LSH values d_1 and d_2 are known for C with respect to A and B.

fields are modified; that is, the experiments show that erroneous measurements can be detected with TPR=0.975 and FPR=0.025 for seven affected modalities. This error rates increase significantly as the number of affected fields decreases; the system is less robust when $n = 3$, with approximately TPR=0.8 and FPR=0.2.

7 PRIVACY AND SECURITY ANALYSIS

In this section, we present informal and experimental analyses of the privacy and security properties offered by ConSec.

7.1 Privacy Analysis

For categorical modalities, ConSec uses HMACs directly to model user behaviour, which are keyed under a randomly generated, application-specific key, HK, assumed to be stored securely on the device. Assuming the use of a secure cryptographic hash function, this avoids disclosing the actual underlying values to an observant server. Moreover, the server does not possess HK, thus precluding the use of dictionary attacks to which non-keyed functions would be vulnerable.

Regarding SB-LSH for numerical data, we note that some information could be leaked based on SB-LSH hashes leaked previously in conjunction with knowledge of their distances in reality. While the actual locations on Earth's surface are masked, SB-LSH reveals the *angular distance* between them. This raises the following question: assuming that some SB-LSH values taken at points during the day can be mapped to known locations—for example, the user being at work during the day and at home at midnight—can one deduce other locations using triangulation?

Suppose locations A and B correspond to the user's home and office locations respectively, and the actual GPS coordinates of these locations are known. The issue pertains to the precision and accuracy with which the latitude and longitude values of unknown location C can be triangulated. From the SB-LSH values of A, B and C, one can compute distances d_1 (between C and A) and d_2 (between C and B) and solve for the latitude and longitude values for unknown location C. This can result in locations either C or C'. We use the chord distance given in Equation 6 to compute the distance between the locations, where the approximate angle θ_a can be computed from the location-hashes and, R, the mean value of the radius of Earth. The precision and accuracy to which C can be localised depends on how accurately and precise d_1 and d_2 can be computed from SB-LSH values. Ultimately, C may be computed with some precision and accuracy that is determined by a confusion region with radius d_c. The larger error in computing d_1 and d_2, the larger d_c, thus decreasing the precision to which C can be localised and, hence, offering a greater degree of privacy to the user. We illustrate this in Figure 4 for clarity.

Table 4: Mean Absolute Error (MAE), Root-Mean-Square-Error (RMSE), Mean, and Standard Deviation of distances errors from SB-LSH values for varying actual distances in kilometers. (All values given w.r.t. each actual distance).

Actual distance (km)	MAE	RMSE	Mean	Std. Dev.
5	2.826	3.266	-1.109	3.804
10	3.908	4.776	-1.645	4.616
25	9.785	12.648	-2.643	12.344
50	14.779	18.819	-4.683	18.588
100	26.258	33.389	-9.417	29.975
500	100.980	124.377	-51.619	113.846
1000	181.192	230.074	-96.341	212.993

We show that the errors in computing distances, e.g. d_1, from the location-hashes are non-uniform and increase for larger actual distances. To show this, we perform simulations using randomly generated pairs of latitude and longitude values with a fixed distance between them. The SB-LSH hashes are computed for the pair values, and the distance between the locations is computed from SB-LSH values. The distances are computed using Equation 7. Table 4 shows the MAE, RMSE, mean, and standard deviation of the errors in computing distances for differing (actual) distances between the pairs of $d = [5, 10, 25, 100, 500, 1000]$ kilometers.

Our results show that, as expected, the error is larger when the location is at a greater actual distance. For example, when C is at 5km from A, the error in computing the distances from the SB-LSH values has MAE=2.826km; at an actual distance of 25km, this increases to MAE=9.785km. Thus, C is localised with greater error, and with poorer accuracy, when it is located further away in reality.

Understanding the actual distribution of the distance errors is also important; uniformly distributed error is desirable such that any unknown locations cannot be localised with high probability. Figures 5 and 6 show the histogram of distance errors for actual distance of 25 and 50 kilometers respectively using a bin size of 1000 meters. The histogram of the distance error is approximately Gaussian, with a flat peak and wide variance; the greatest distances fall at the tails of the distributions. That is, unknown locations that are far away cannot be localised with low error.

From the above, we can conclude that inferring locations from SB-LSH values incurs significant error, even if some information is leaked about the user with the assistance of supplementary knowledge about the users' actual locations in reality. This uncertainty is determined primarily by the largest distance of the two closest known locations. The greatest distance determines the error (d_c in Figure 4) in estimating the unknown locations from SB-LSH hashes. Moreover, the error increases dramatically and proportionally to larger actual distances in reality.

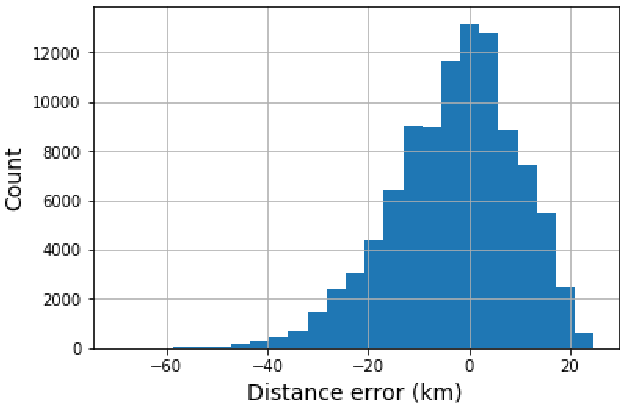

Figure 5: Histogram of errors in computing distances from SB-LSH values w.r.t. points at an actual distance of 25 km.

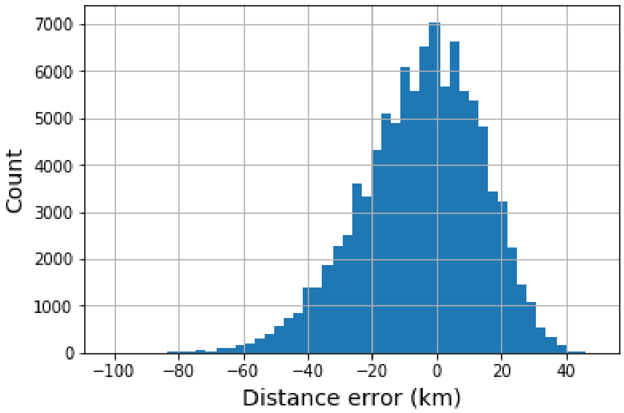

Figure 6: Histogram of errors in computing distances from SB-LSH values w.r.t. points at an actual distance of 50 km.

7.2 Security Analysis

We now describe how ConSec thwarts a variety of security threats drawn from the threat model in Section 3.2.

7.2.1 Modified Input Modalities. One attack pertains to modifying the SB-LSH and HMAC values produced by ConSec-App with the aim of poisoning the model to deny future legitimate accesses or facilitate illegitimate accesses. While data is secured within transit using TLS, as described in Section 5, this does not preclude an adversary with the ability to alter some number of SB-LSH and HMAC values within ConSec-App itself. Our experimental results in Section 6.2 show that seven or more inaccurate modalities (of 26 in total) can be detected with low error (approximately TPR=0.975 and FPR=0.025), which somewhat decreases when a smaller number of inputs are modified (TPR=0.8, FPR=0.2 for three inputs).

7.2.2 Server-side Privacy. Evidently, using raw contextual values gives rise to privacy risks against honest-but-curious adversaries who aim to exploit users' behavioural data for ulterior purposes. This extends to unauthorised disclosure if an attacker successfully exfiltrates data from the server and releases it publicly after exploiting some vulnerability. We reduce the impact of these cases as ConSec models and authenticates users' behaviour *directly* from keyed HMAC and SB-LSH values rather than raw, unprotected values. In Section 6, we also show experimentally that significant errors are involved when attempting to infer unknown locations from known SB-LSH values.

7.2.3 Network Attacks. Another threat to contextual authentication systems is where a network adversary re-sends previously observed messages containing the feature vectors (whether protected or not) to the server, with the aim of poisoning the behavioural model and influencing the authentication algorithm. This attack is protected through the use of a secure channel (TLS) between the application and authentication server with replay protection. Moreover, the contextual data vector includes a timestamp to ensure its recentness against replay attacks.

8 CONCLUSION

In this work, we presented ConSec—a privacy-enhancing, context-aware authentication system that utilises locality-sensitive hashing for masking plain sensor measurements from a user device. ConSec learns, models and authenticates users' behaviour directly from protected values in an outsourced fashion without disclosing raw measurements to an honest-but-curious authentication server. We began with a detailed review of existing contextual authentication schemes and methods for preserving the privacy of sensitive input modalities for continuous authentication generally. We then showed how ConSec supports learning from both numerical and categorical data in a flexible manner using the SB-LSH algorithm by Ji et al. [15] and keyed HMACs, without incurring the complexities of existing privacy-enhancing approaches, such as homomorphic encryption and multi-party computation. After describing the data types and authentication algorithms used by ConSec, experimental results were presented using data collected from 35 users in the field. This was followed by analyses showing its its robustness to modified input modalities and the errors involved in attempting to infer unknown locations from known values.

In future work, we aim to evaluate ConSec in a long-term usability study to determine users' attitudes towards using the system and, in particular, concretely measuring its handling of gradual behavioural shifts in users' behavioural patterns. We also aim to explore the scheme's robustness to *mimicry attacks* in which a physical attacker possesses the device and attempts to reconstruct the victim's behaviour, such as visiting their home or place of work.

ACKNOWLEDGMENTS

The authors would like to thank other members of the OneSpan Innovation Centre—Paul Dunphy, Andreas Gutmann, Sharon Lee, Steven J. Murdoch, and Tom De Wasch—and the reviewers for providing many useful comments and suggestions towards improving this work.

REFERENCES

[1] Android. 2019. Location strategies. https://developer.android.com/guide/topics/location/strategies.

[2] David Arthur and Sergei Vassilvitskii. 2007. K-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms.*

[3] Louise Barkhuus and Anind K. Dey. 2003. Location-based services for mobile telephony: a study of users' privacy concerns. In *Interact*, Vol. 3. 702–712.

[4] Dan Cvrcek, Marek Kumpost, Vashek Matyas, and George Danezis. 2006. A study on the value of location privacy. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society.* ACM.

[5] George Danezis, Stephen Lewis, and Ross J. Anderson. 2005. How much is location privacy worth?. In *Workshop on the Economics of Information Security*, Vol. 5.

[6] Alexander De Luca and Janne Lindqvist. 2015. Is secure and usable smartphone authentication asking too much? *Computer* 48, 5 (2015), 64–68.

[7] Defense Mapping Agency. 1984. Department of Defense World Geodetic System: its definition and relationships with local geodetic systems. TR8350.2 (1984).

[8] Josep Domingo-Ferrer, Qianhong Wu, and Alberto Blanco-Justicia. 2015. Flexible and robust privacy-preserving implicit authentication. In *IFIP International Information Security Conference.* Springer.

[9] Drew Fisher, Leah Dorner, and David Wagner. 2012. Location privacy: user behavior in the field. In *Proceedings of the 2nd ACM Workshop on Security and Privacy in Smartphones and Mobile Devices.* ACM.

[10] Craig Gentry. 2009. *A fully homomorphic encryption scheme.* Vol. 20. Stanford University.

[11] Aditi Gupta, Markus Miettinen, N Asokan, and Marcin Nagy. 2012. Intuitive security policy configuration in mobile devices using context profiling. In *International Conference on Privacy, Security, Risk and Trust.* IEEE.

[12] Kimmo Halunen and Visa Vallivaara. 2016. Secure, usable and privacy-friendly user authentication from keystroke dynamics. In *Nordic Conference on Secure IT Systems.* Springer, 256–268.

[13] Marian Harbach, Emanuel Von Zezschwitz, Andreas Fichtner, Alexander De Luca, and Matthew Smith. 2014. It's a hard lock life: A field study of smartphone (un) locking behavior and risk perception. In *Symposium on Usable Privacy and Security.* ACM.

[14] Eiji Hayashi, Sauvik Das, and Shahriyar Amini. 2013. CASA: Context-Aware Scalable Authentication. In *Proceedings of the 9th Symposium on Usable Privacy and Security.*

[15] Jianqiu Ji, Jianmin Li, Shuicheng Yan, Bo Zhang, and Qi Tian. 2012. Super-bit locality-sensitive hashing. In *Advances in Neural Information Processing Systems.*

[16] He Liu, Stefan Saroiu, Alec Wolman, and Himanshu Raj. 2012. Software abstractions for trusted sensors. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services.* ACM.

[17] Markus Miettinen, Stephan Heuser, Wiebke Kronz, Ahmad-Reza Sadeghi, and N. Asokan. 2014. ConXsense: Automated context classification for context-aware access control. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security.*

[18] Lawrence O'Gorman. 2003. Comparing passwords, tokens, and biometrics for user authentication. *Proc. IEEE* 91, 12 (2003), 2021–2040.

[19] Pascal Paillier. 1999. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques.*

[20] Sarah Pearman, Jeremy Thomas, Pardis Emami Naeini, Hana Habib, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Serge Egelman, and Alain Forget. 2017. Let's go in for a closer look: observing passwords in their natural habitat. In *Proceedings of the ACM Conference on Computer and Communications Security.* ACM.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[22] Oriana Riva, Chuan Qin, Karin Strauss, and Dimitrios Lymberopoulos. 2012. Progressive authentication: deciding when to authenticate on mobile phones. In *USENIX Security Symposium.*

[23] Siamak F. Shahandashti, Reihaneh Safavi-Naini, and Nashad Ahmed Safa. 2015. Reconciling user privacy and implicit authentication for mobile devices. *Computers & Security* 53 (2015), 215–233.

[24] Carlton Shepherd, Raja Naeem Akram, and Konstantinos Markantonakis. 2017. Towards trusted execution of multi-modal continuous authentication schemes. In *Proceedings of the 32nd ACM Symposium on Applied Computing.* ACM.

[25] Weidong Shi, Jun Yang, Yifei Jiang, Feng Yang, and Yingen Xiong. 2011. Senguard: Passive user identification on smartphones using multiple sensors. In *7th International Conference on Wireless and Mobile Computing, Networking and Communications.* IEEE.

[26] Babins Shrestha, Nitesh Saxena, Hien Thi Thu Truong, and N. Asokan. 2015. Contextual proximity detection in the face of context-manipulating adversaries. *arXiv preprint arXiv:1511.00905* (2015).

[27] Frank Stajano. 2011. Pico: No more passwords!. In *International Workshop on Security Protocols.* Springer.

[28] Jaroslav Šeděnka, Sathya Govindarajan, Paolo Gasti, and Kiran S. Balagani. 2015. Secure outsourced biometric authentication with performance evaluation on smartphones. *IEEE Transactions on Information Forensics and Security* 10, 2 (2015), 384–396.

[29] John M. Wallace and Peter V. Hobbs. 1977. Atmospheric science: an introductory survey. (1977), 103–104.

[30] H. Witte, C. Rathgeb, and C. Busch. 2013. Context-aware mobile biometric authentication based on support vector machines. In *4th International Conference on Emerging Security Technologies.* IEEE.