Thumprint: Socially-Inclusive Local Group Authentication Through Shared Secret Knocks

Sauvik Das Gierad Laput Chris Harrison Jason I. Hong
Carnegie Mellon University, Human-Computer Interaction Institute
5000 Forbes Ave., Pittsburgh, PA, 15213
sauvik@cmu.edu, {gierad.laput, chris.harrison, jasonh}@cs.cmu.edu

ABSTRACT
Small, local groups who share protected resources (e.g., families, work teams, student organizations) have unmet authentication needs. For these groups, existing authentication strategies either create unnecessary social divisions (e.g., biometrics), do not identify individuals (e.g., shared passwords), do not equitably distribute security responsibility (e.g., individual passwords), or make it difficult to share or revoke access (e.g., physical keys). To explore an alternative, we designed Thumprint: inclusive group authentication with a shared secret knock. All group members share one secret knock, but individual expressions of the secret are discernible. We evaluated the usability and security of our concept through two user studies with 30 participants. Our results suggest that (1) individuals who enter the same shared thumprint are distinguishable from one another, (2) that people can enter thumprints consistently over time, and (3) that thumprints are resilient to casual adversaries.

Author Keywords
Authentication; usable security; socially-inclusive authentication; HCI; social cybersecurity; sensors

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous;

INTRODUCTION
Authentication is important for any secure system, but is typically designed for individuals with privately owned resources and a strong desire to protect them (e.g., bank statements, emails) [6,31]. This focus, while important, has resulted in authentication tools (e.g., PINs, biometrics) that are often inappropriate for a large spectrum of small, local groups who have relaxed security needs and *collectively* share accounts, devices and/or spaces. For example, families who share tablets with children. Shared passwords and PINs do not allow for parental controls, whereas requiring individual passwords for each family member is unwieldy and often subverted [23,26].

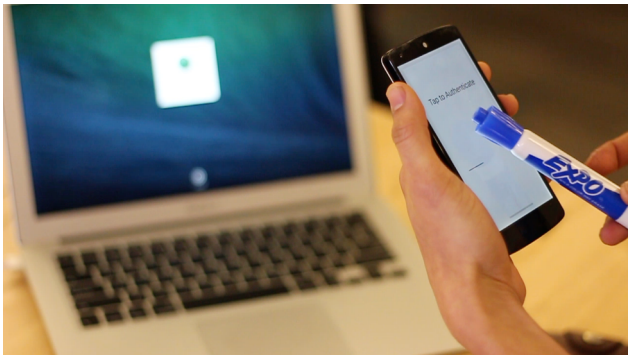

Figure 1. With Thumprint, groups of users learn a single, shared secret knock that they enter on a surface instrumented with (or containing) an accelerometer and microphone (here, a smartphone) in order to authenticate.

Another example is interest-based organizations that share equipment (e.g., a tennis club). Each group member should have access to this shared equipment, but group members often change so using a shared password or key can make it difficult to revoke access from old members [1]. Conversely, use of individual secrets to access group resources can be socially inappropriate [8,13] or rude [25]. Similar situations arise with, for example, employees who share kitchenettes, waitstaff who share access to employee-only areas, and roommates who share a Netflix account.

While these group-owned resources should only be accessible by members, individuals in the group trust each other and only need enough security to discourage casual outsiders [17,26]. Thus, these diverse groups could benefit from a new form of socially-inclusive authentication that provides reasonable outsider rejection and can identify group members without individual secrets.

To that end, we introduce Thumprint: group authentication through shared secret knocks. Secret knocks were famously used by Prohibition-era speakeasies to authenticate prospective bar patrons when sale of alcohol was prohibited in the U.S. [22]. As they are secrets shared through trusted social channels, they not only authenticate, but also promote group cohesion [34]. Our idea with Thumprint was to leverage advancements in sensing techniques to realize a digitized secret knock authenticator.

CHI 2017, May 06 - 11, 2017, Denver, CO, USA

ACM 978-1-4503-4655-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025991

In brief, Thumprint authenticates groups based on group members' expression of a shared, 3-second knock on a surface instrumented with (or containing) an accelerometer and microphone (see Figure 1). As the secret knock is shared, group members need not maintain their own individual secrets. However, because individual expressions of the knock are variable, Thumprint can still identify individuals. Current members can safely share the secret with new members, but as individuals are identifiable, previous members can have their access revoked or limited. Notably, Thumprint is not designed to provide perfect security—it is designed to be lightweight and inclusive.

To evaluate the usability and security of Thumprint, we ran two user studies. Through these studies, we found that (1) different people who enter the same thumprint can be reliably recognized, (2) people can consistently enter their thumprints over time-separated sessions, and (3) thumprints are reasonably secure against casual, motivated adversaries.

Concretely, we offer the following broad contributions to HCI and usable security: (1) We present an early exploration into the design of a "social" cybersecurity system motivated by growing empirical evidence that social factors strongly affect the usability and adoption of security systems [8–10]; (2) We introduce the concept of "socially-inclusive" authentication for groups who share collectively-owned resources, and, in so doing, synthesize design considerations for these groups that are not covered by existing taxonomies of authentication research [3,18].

BACKGROUND AND RELATED WORK

Apart from having relaxed outsider rejection needs, we identified three other design dimensions for local group authentication from a survey of background literature.

The first is *inclusivity with identifiability*. Inclusivity is broad, but we define it as reducing the need for individual secrets when authenticating to common group resources. Indeed, requiring individual secrets (e.g., private passwords) to access shared resources is cumbersome and can lead to non-compliance (e.g., sharing passwords) [26]. Individual secrets can also have social consequences: not sharing these secrets can be rude [33] or otherwise create social friction between people [5,8,21,32].

Consider the use case of a spouse needing her partner to check a shared calendar on her smartphone: What should she do if the phone is password protected? If the choice is between losing social capital with a loved one or sharing a password, people often opt for the latter [13,34], and, in so doing, break the security assumptions of the system.

Still, social group structures vary widely [35], and some group structures may require access control at the individual level. Thus, inclusivity should ideally come with identifiability to allow for audit logs, personalized functionalities, and tiered access to resources. For example, having one shared family PIN prevents individual family members from creating personalized profiles and precludes the ability for parents to have privileged access [5,11].

Another need is *proportionate distribution of security responsibility*. Responsibility for the well-being of the group should be appropriately distributed across members [20,24], so individuals may be resistant to weighty security solutions that require a large personal investment of time or effort. Accordingly, authentication should be sensibly simple for individual group members. Otherwise, individuals who are less knowledgeable or motivated about security could compromise the whole group's security (e.g., by creating a weak password to access group resources).

One example of where proportionate security responsibility is employed is when nursing staff must use authentication to access hospital computing systems. Nurses often have urgent needs and cannot each be expected to remember long, complex passwords, even if hospital IT has a different perspective. Doing so results in nursing staff writing down the passwords for sensitive hospital equipment right on the apparatus [23].

Finally, there are a number of small, local groups that are built off of a common-identity (e.g., a common interest in tennis). Typically, groups like these have a lot of churn: i.e., they often gain new members and lose old members [28]. For these groups, it should be *easy to share access with new members and revoke access from old members*. Student organizations that have members rotating every semester, for example, need a simple and reliable way to revoke and grant access to shared equipment closets.

Few existing security solutions support these needs as core functionality. Indeed, in a longitudinal field study of the access control habits of a local group who shared a work space, Bauer et al. found that existing strategies for authentication and access control (i.e., sharing physical keys) could not support the group's ideal policies [1].

However, while there has been little work on creating better local group authenticators, there has been some promising work that explores the problem space. Toomim et al. introduced a photo access control mechanism where the correct audience should be able to answer a question based on shared knowledge [33]. Gilbert created a social encryption tool, OpenBook, that obfuscates messages in a way that can only be reconstructed by the shared social context between sender and receiver [14]. And, Egelman et al. and Brush introduced the "Family Account" [5,11]—a shared account for all family members. Still, Family Accounts are for access control, not authentication.

Our key idea is to use sensable, behavioral interactions as shared group secrets that have varying individual expressions. For instance, accelerometers in mobile devices have been used for detecting a wide range of gestures, activities [7] and hand postures [15]. There has been increasing interest in using these forms of sensable user behaviors for authentication. One notable example is the use of keystroke

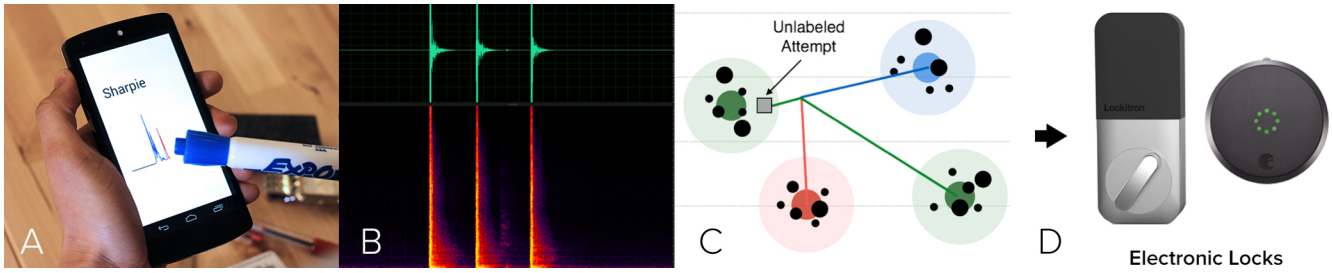

Figure 2. With Thumprint, users enter secret knocks on an instrumented sensor surface (A) from which a variety of time and frequency domain features are extracted (B). These readings are projected onto a reduced feature space, where each authentication attempt is compared against previously learned thumprint expressions from group members (C). If a match, Thumprint will provide access by regulating an end-point such as an electronic lock (D).

dynamics, or the rhythm with which people type, for authentication [19,27]. With TapSongs, Wobbrock extended this approach to intentional behaviors in the form of rhythmic up-down taps on a binary sensor to match a known jingle timing model [36]. Lin, Ashbrook and White used a similar approach to pair I/O constrained devices through entry of a secret "tapword" on both devices [25]. In all of these cases, outsider rejection was not perfect (~20% failure rates), but insider acceptance was promising.

These approaches, while inspirational, were not designed to be inclusive nor were they meant for groups. With Thumprint, we extend these advances in using sensable, intentional behaviors for group authentication.

THUMPRINT

Design Inspiration

There are many analogues in the offline world that illustrate the use of shared secrets for group authentication [2]. There is the famous biblical example of correctly pronouncing the word "shibboleth" that the Gileadites used to identify the invading Ephraimites who could not pronounce the "sh" sound [37]. Other examples include secret handshakes (e.g., the use of selective pressure in handshakes) and code phrases (e.g., saying the words "open sesame" to gain access to a secret lair) [2]. In all of these cases, the shared secret not only authenticates, but is inclusive and reinforces group cohesion [34]. As previously mentioned, Thumprint is inspired by the secret knocks used at speakeasies [22].

Accordingly, Thumprint authenticates local groups with a secret knock consisting of a shared secret token and pattern. The token is typically just a finger or a knuckle. However, any small, solid object can be used (e.g., a pen or coin). The pattern can be any sequence of knocks within a three-second period. Authentication occurs by entering the knock on a sensed surface. Furthermore, as each person has a different mechanical expression of the knock, Thumprint can also identify individuals.

System Overview

Figure 2 shows a high-level description of how Thumprint works. To operate, Thumprint requires two components: a surface instrumented with an accelerometer and microphone (or a device already containing these sensors, such as a smartphone), and an endpoint to regulate access.

The *sensed surface* can take on many forms—e.g., a smartphone screen, a door, or tabletop. As a proof-of-concept implementation, we used an Android smartphone as our sensed surface. Meanwhile, the authentication *end-point* can be anything that regulates access control, such as a tablet or an electronic smart lock.

To use Thumprint, a group of at least two members must register themselves by entering the shared secret knock. To register, each member enters the secret knock on the sensed surface five to ten times. Thumprint records 3-seconds of accelerometer and microphone data from each of the registration attempts, extracts a set of time and frequency-domain features from those sensor streams, and stores each feature vector labeled with the individual's ID as training data. We selected a three-second duration to allow for sufficient variation in knock expression. Thumprint then processes these training data to "learn" both the shared secret knock and each individual's expression of the knock.

To later authenticate, an individual should reproduce the secret knock roughly in the same manner in which she registered. The system extracts an unlabeled feature vector from the authentication attempt and compares it against training data. If the unlabeled feature vector is similar enough to the group thumprint, it is authenticated as the member whose training data is most similar. Moreover, Thumprint computes a similarity score for each group member—so, depending on the security needs of the group, it is possible to provide tiered access control so that a knock is only authenticated if its similarity score is sufficiently high. If the score is too low, it is possible to provide lower tier access, or prompt the user to repeat the knock.

Training Pipeline

Once participants have provided a set of training data during the registration process, the key question is how can we use this training data to later authenticate group members? More formally, if we have an unlabeled authentication attempt, $\vec{u}$, we must determine determine whether or not to authenticate $\vec{u}$ and, if so, which group member is most likely to have produced $\vec{u}$.

One approach is to use a one-class classifier, but these typically require a large amount of training data—dozens, if not hundreds of training points per group member. Instead,

Signal Trans-formation	Applicable sensor streams	Signal parti-tioning	Extracted features
Time-domain	Acceleration & Acoustic	Whole & 1-second windows	Mean, mean absolute value, std. dev., max, min, RMS, zero-crossings, total energy, second order average, third order average, average amplitude change
Wavelets (D4)	Acceleration & Acoustic	Whole	Total power, max power, power bands, mean absolute coefficient value per band, coefficient standard deviation per band
Fourier	Acceleration & Acoustic	1-second windows	Dominant frequency, spectral centroid, spectral rolloff, spectral crest factor, spectral flatness, lower 1Khz bins
MFCCs	Acoustic	25ms windows	For each of the 12 coefficients, over all 25 ms windows: mean value, std. dev., mean first order-change, mean second-order change

Table 1. Features extracted for every thumprint following a variety of recommendations from prior work in sensing techniques. In total, 1020 features are extracted though the feature space is later dramatically reduced to avoid overfitting.

to make accurate decisions with far fewer training data, we use a form of template matching: i.e., we compare $\vec{u}$ to the set of templates, T, that are constructed during training to represent individual expressions of the shared secret knocks. If the distance between $\vec{u}$ and any $\vec{t} \in T$ is sufficiently low, then we authenticate $\vec{u}$ as coming from the user who produced $\vec{t}$. Otherwise, we reject $\vec{u}$ as coming from an outsider. In brief, this process requires three implementation steps: *feature extraction*, *feature processing*, and *template construction*.

Feature Extraction

We extracted a set of features from each of the input acceleration and acoustic signals that users entered during registration. Features were extracted from the raw time-domain PCM values, as well as a Daubechies D4 wavelet and Fourier transformation (FFT) of the signals. For the raw-time domain and FFTs, we extracted features for each one-second segment of the signal to better preserve the temporal variance of the thumprints across the three-second window (i.e., to characterize thumprints that may be intentionally non-rhythmic and irregular). This was unnecessary for the wavelet transformation, as wavelet coefficients capture temporal variation by design [4]. Finally, for the acoustic signal, we also extracted features from the mel-frequency cepstral coefficients (MFCCs) computed for each 25 millisecond time-window of the signal. A complete description of all considered features is in Table 1.

At the end of the feature extraction process, we have a matrix, $F \in \mathbb{R}^{mxn}$, where m is the number of training attempts in the system and n is the number of features that have been extracted. Each row of this matrix represents the features extracted for a particular training attempt. We also have a class vector, $\vec{y} \in \mathbb{Z}^m$, that represents which participant produced which row of F.

Feature Processing

Next, we employ a number of supervised pre-processing techniques on F. First, we use correlation-based feature subset selection (CFS) [16] to reduce the feature space to a parsimonious subset that distinguishes group members. The reduced feature space is reduced to *at most* one feature per row of training data to mitigate overfitting. We then discretize the feature space using Fayyad-Irani discretization [12]—a technique to bin continuous variables into discretized intervals that minimize the entropy of known class values in each bin. Supervised discretization can enhance predictive performance in many cases [12]. Intuitively, we discretize the feature space so that our template matching algorithm is less sensitive to micro-fluctuations in raw feature values. At the end of the feature selection and discretization process, we have a reduced matrix, $A \in \mathbb{Z}^{mxk} = Discretize(CFS(F, \vec{y}))$, where $k \leq m$ is the number of features in the reduced feature space.

Template Construction

Following feature selection and discretization, we need to deconstruct the training matrix, A, into a set of known templates, T. The two most straightforward approaches are: (1) create a single template for each user by averaging all of their training attempts; and, (2) create a distinct template for each training attempt. Both are suboptimal. The first approach fails to acknowledge that individuals might have multiple expressions of the shared secret knock—for example, one might sometimes enter the knock with more force, or other times at a slower pace. If all of these different expressions are averaged, then the average will look different than any of the individual expressions. The second approach fails to learn common patterns across training attempts and reduces security by expanding the surface area in the feature space that represents the group shared secret. Thus, a single stray training attempt can compromise the security of the group by expanding the acceptable definition of the group's shared secret.

Instead, we take a middle-ground approach by clustering together related training attempts into distinct templates. With this compromise approach, we can detect multiple distinct expressions of the secret knock within users but still minimize the surface area that represents the group shared secret in feature space. To do so, we run a k-means clustering algorithm on the training data for each individual group member and automatically determine the number of clusters that are appropriate using the average silhouette width method [30]. At the end of this process, we have a set of templates, T, that each contain a subset of the training attempts derived from one registered group member.

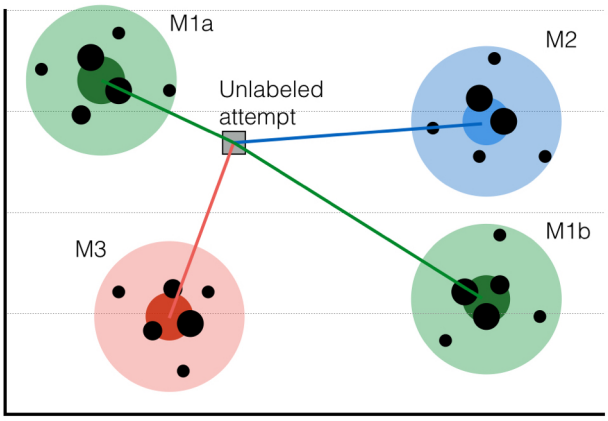

Figure 3. To authenticate, an unlabeled feature vector is transformed into the reduced feature space and then its distance to nearby training clusters is calculated. In this case, the unlabeled attempt would not be authenticated because it is too far from candidate training clusters.

Authentication

Once training is complete, making an authentication decision on an unlabeled attempt, $\vec{u}$, is a matter of finding the cluster(s) closest to $\vec{u}$ and then thresholding on the distance between $\vec{u}$ and the closest cluster centroid:

$$\min_i d(\vec{u}, i) = \left(\sum_j \frac{|\vec{u} - \overrightarrow{t_{ij}}|}{k}\right) \Big/ |T_i|$$

where T_i represents the ith cluster and $\overrightarrow{t_{ij}}$ represents the jth training vector in T_i, $|T_i|$ represents the size of T_i, and k represents the size of the feature space after feature reduction. In practice, the value of $d(\vec{u}, i)$ should typically fall within a range of 0 to 1 for any reasonably close attempt. Lower $d(\vec{u}, i)$ suggests a closer match between $\vec{u}$ and T_i so in the simplest case of identification without authentication, we can identify $\vec{u}$ as coming from the member who produced the cluster that minimizes $\min_i d(\vec{u}, i)$. To add authentication, we can introduce a threshold h. If $d(\vec{u}, i) \leq h$, then we authenticate; otherwise, we reject. Figure 3 visually illustrates the process.

One potential concern is drift—or the idea that individuals might gradually change their expression of the secret knock over time. We handle drift by incrementally updating our training model as new training data is available (e.g., as a group member *successfully* authenticates over time) and by increasing the weight of more recent training attempts.

STUDY 1: FEASIBILITY EVALUATION

To evaluate the feasibility of our Thumprint concept, we ran an initial lab study with 15 participants ranging in age from 18-55 years old (mean age 26, eight females). Our goal with this feasibility evaluation was to answer the questions: Given a group of pre-registered users who all share a thumprint and a set of un-registered adversaries who know the group's shared thumprint, (i) how easily can outsiders impersonate group members? (ii) how often are group members confused as outsiders? and, (iii) how often are group members are confused for one another?

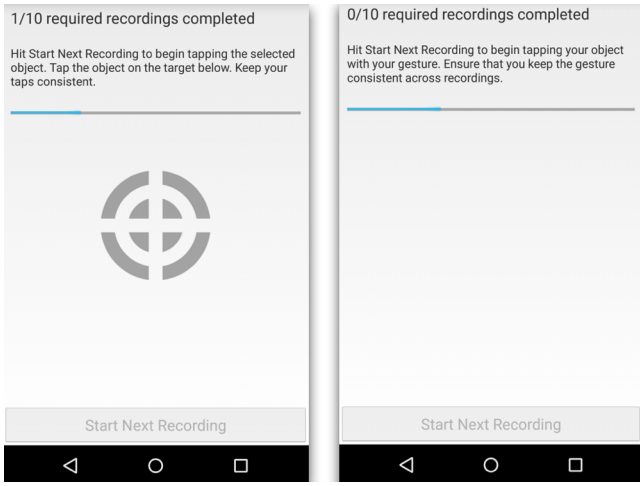

Figure 4. Screenshots of the app in which participants entered preset (left) and custom (right) thumprints.

Procedure

To answer these questions, we ran an hour-long lab study. Consenting participants proceeded through two flows: a flow in which they entered pre-selected thumprints, and a flow in which we had them create their own unique thumprints.

Participants entered their thumprints on a Nexus 5 Android phone. For each thumprint, our application recorded 3-seconds of accelerometer data sampled at 2kHz and 3-seconds of microphone data sampled at 44.1 kHz.

In the first flow, we selected 10 example objects that spanned a variety of materials: a *wooden* letter opener, a *rubber* eraser and fridge magnet, a *plastic* eye drop bottle, pen and chapstick, a *metal* Swiss army knife and watch, a *leather* wallet, and the participant's *knuckle*. Participants were instructed to hold the phone comfortably in their non-dominant hand. Then, for each of the 10 thumprints, participants held the object in their dominant hand (or used the knuckle of their dominant hand) and knocked repeatedly on the center of the screen for 3 seconds. They repeated the entry of each thumprint 10 times in total.

After completing this flow, participants were allowed to create their own custom thumprints. Participants selected four tokens from the 10 objects provided and then had to develop their own unique knock for each of these tokens. Thus, participants could knock using any part of an object, anywhere on the screen and in any pattern. A researcher demonstrated these options to participants prior to start of this flow. Participants again had to repeat each of their four unique thumprints ten times each. We video-recorded participants entering their unique thumprints so that we could later use these recordings to simulate shoulder surfing adversaries. Data from this flow was primarily used as raw material for our second study.

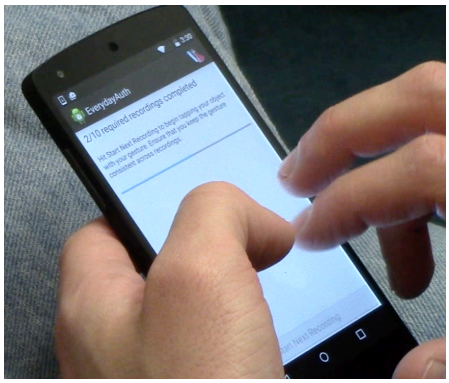

Figure 5. Photo of a participant entering a unique thumprint on our data collection app. Participants had to create four unique thumprints and repeat them 10 times each. The interface provided a three-second countdown timer to assist.

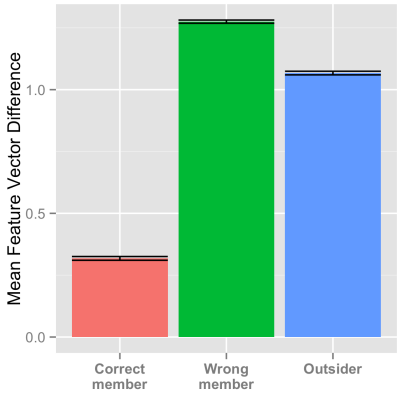

Figure 6. Mean feature vector difference (along with 95% confidence intervals) for user testing attempts (relative to their own training data and other group member training data), as well as outsider attempts.

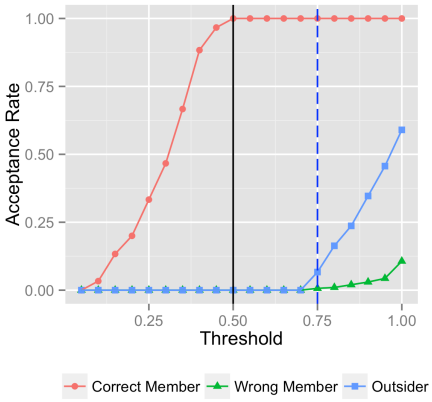

Figure 7. Acceptance rate as a function of feature vector difference. The black vertical line is where 100% of user attempts are accepted, and the blue dashed line is where >0% of outsider attempts are first accepted.

To improve data collection, the application interface showed a sliding progress bar to inform participants of their three-second time limit. For the first flow, the interface also contained a target at the center of the screen to assist participants with their aim. Figures 4 and 5 show screenshots of the process.

With 15 participants, 14 thumprints, and 10 repetitions per thumprint, we had 2100 thumprint data points consisting of 3-second accelerometer and acoustic streams. We computed the aforementioned time and frequency domain features for each of these data points.

Results

To answer questions (i), (ii), and (iii), we needed to simulate data from small groups with a shared thumprint, as well as outsiders attempting to break those thumprints. For the 10 pre-defined thumprints (first flow), simulating small groups and competent outsiders was straightforward. As each participant produced the same set of 10 thumprints, every participant could effectively be partnered with some number of other participants to simulate a small group, and every other participant could be a casual adversary.

Thus, we randomly aggregated different subsets of $n \in [3,5,10]$ participants to represent small groups of varying sizes. For training, we used a random sample of 80% of each group member's data, and kept a holdout set of 20% for testing. Then, for each simple thumprint, we used data from the remaining $15 - n$ participants to simulate a strong adversary who knew the group thumprint (as all users in the first flow entered the same thumprints).

It is worth noting that we did not design Thumprint to be extremely strong against adversaries who exactly knew the group thumprint. Yet, our results exceeded expectations.

Figure 6 shows the mean minimum feature vector difference, $\min_i d(\vec{u}, i)$, for authentication attempts by actual group members versus those of adversaries. From Figure 5, we can see a large and clear separation between the feature vector differences of authentic attempts (d=0.32) from adversarial attempts (d=1.06). In Figure 7, we plot the acceptance rate of these attempts as a function of a configurable authentication threshold. We can see that Thumprint worked well: at a threshold between [0.5, 0.75), we achieved 100% true positives and no false positives.

This result is promising—suggesting that thumprints might provide reasonable outsider rejection while maintaining high insider acceptance. However, it is worth keeping in mind that our adversaries in this evaluation were not specifically trying to replicate a thumprint in a way that they observed someone else. Furthermore, we collected all data within a single session, so it is not surprising that people's testing attempts were quite similar to their training attempts. We address these weaknesses in our second study.

STUDY 2: CONSISTENCY AND SECURITY EVALUATION

We ran a second lab study, with 15 new participants, ranging in age from 18-57 years old (mean age 28, five females). Our goal with this study was to answer the following two questions: (iv) can people consistently enter complex thumprints after time-separated sessions? And (v) how well can thumprint reject motivated adversaries?

Procedure

This study consisted of two 30-minute sessions that took place 24 hours apart. Broadly, we had participants register a thumprint in the first session and re-enter the same thumprint a day later. In addition, we had participants play the role of an adversary trying to break into others' thumprints, given a set of capabilities and constraints.

Session 1: Participants initially had to enter four simple, pre-defined thumprints to familiarize themselves with the application interface. This flow was the same as it was in the first study, where participants selected from a set of

provided objects and tapped them repeatedly on the center of the screen. Once they had completed the pre-defined thumprint flow, they were shown a video of a custom thumprint created by a participant from the second flow in first study. Participants were allowed to watch the video as often as they liked. Once satisfied, they were instructed to replicate what they saw to the best of their ability. Participants were also told that they would have to re-enter this thumprint the next day.

Of note, participants were shown one of three custom thumprints corresponding to the study group to which they were assigned. We selected three groups because we wanted several participants to learn the same thumprint so that we could later group them, and to ensure that our results were not tied to any single thumprint.

Session 2: Participants came back for a follow-up session a day later. Their first task in this follow-up session was to re-enter the custom thumprint they had seen at the end of the previous day's session. They had to do so from memory—no assistance was provided. Once completed, each participant had to enter four more custom thumprints. This time, however, we had participants play the role of adversary. Their task was to replicate other thumprints given a set of constraints to simulate different adversary models.

The four adversary models and their corresponding affordances were: (1) *video+correct token:* the full video recording of thumprint entry and use of the correct token; (2) *video+wrong token:* the full video recording of thumprint entry, but the correct token could not be used; (3) *sound only*: the audio recording of thumprint being entred (stripped from the video recording) and a best-guess attempt at picking the correct token; and, (4) *token only*: only knowledge of the correct token provided. Table 2 shows all of the thumprints each participant had to enter, along with the relevant constraints. Note that, as before, participants entered 10 repetitions for *each* thumprint.

At the end of the study, we had data for three thumprints (T1, T2, and T3) across two sessions from five participants each. For each of these thumprints, we also had 10 *video+wrong, token only* and *sound only* adversarial replications. For another set of three thumprints (T4, T5, T6), we had 10 *video+correct* adversarial replications. Notably, as *video+correct* adversaries can be considered authentic group members (if their data is included in the process of training Thumprint), we can divide the 10 *video+correct* replications into subsets of group members and adversaries as necessary.

Consistency and Security Evaluation

To answer the question (iv), can people remember and enter complex thumprints over time, we trained a model on data collected for T1, T2, and T3 from the first day's session and tested it on data collected for those same thumprints collected in the second day's session. Specifically, we calculated the minimum feature vector difference of the authentication attempts from the second session relative to data from the same user in the first session. As a point of reference, we also calculated the minimum feature vector difference of the 10 *video+wrong*, *sound only*, and *token only* adversarial attempts relative to group member training data from the first session. To see if group members could be misidentified with each other, we also calculated the minimum feature vector difference between user authentication attempts and the training data for other group members. Figure 8 shows the results.

Part.	T1	T2	T3	T4	T5	T6
1	Main	*S*	*V*	*V+T*		*V+T*
2	Main	*V*	*T*	*V+T*		*V+T*
3	Main	*T*	*S*	*V+T*		*V+T*
4	Main	*S*	*V*	*V+T*		*V+T*
5	Main	*V*	*T*	*V+T*		*V+T*
6	*S*	Main	*T*	*V+T*	*V+T*	
7	*V*	Main	*S*	*V+T*	*V+T*	
8	*T*	Main	*V*	*V+T*	*V+T*	
9	*S*	Main	*T*	*V+T*	*V+T*	
10	*V*	Main	*S*	*V+T*	*V+T*	
11	*V*	*T*	Main		*V+T*	*V+T*
12	*T*	*S*	Main		*V+T*	*V+T*
13	*S*	*V*	Main		*V+T*	*V+T*
14	*V*	*T*	Main		*V+T*	*V+T*
15	*T*	*S*	Main		*V+T*	*V+T*

"Main"*: Group thumprint;* ***"V"****: video+wrong token;*
"S"*: "sound only";* ***"T"****: token only;* ***"V+T"****: video+correct token*

Table 2. Study 2 flow for each participant. The columns represent the six thumprints selected from Study 1. Cell values with "main" refer to thumprints participants learned in session 1 and replicated in session 2. Other cell values refer to thumprints replicated as adversaries.

We can see that mean feature vector difference for all authentication attempts by participants as compared to their *own* training data (d=0.38) from a previous session is much lower than the three adversary models (ds=0.70, 0.74, 0.70), as well as those of the wrong group members (d=0.76). In fact, participants are not much more inconsistent across time-separated sessions than they are within the same session (d=0.32 in Study 1). This marked difference between authentic user and adversarial attempts lends support to the conjecture that users can effectively replicate thumprints over time and cannot easily be impersonated by casual but motivated adversaries.

To definitively answer question (v), we next sought to translate our findings into individual authentication decisions. In addition to the models for T1-T3 that we used in the previous analysis, we also included models for T4-T6. Specifically, for each of T4, T5 and T6, we selected five participants to be "group members" and five participants to be *video+correct token* adversaries. We trained a model on 80% of the available data for the group members, holding out the additional 20% for testing.

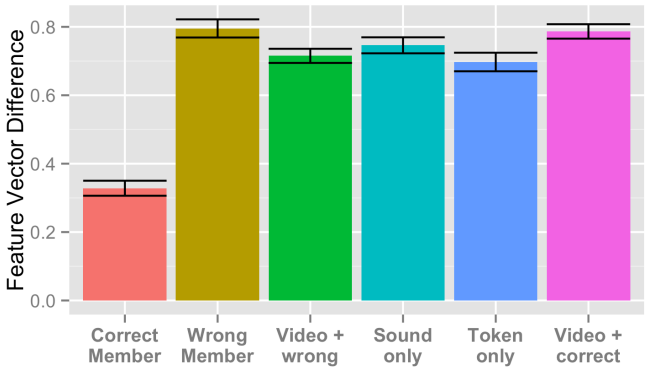

Figure 8. Mean feature vector difference (along with 95% confidence intervals) for T1-T3 across authentic and adversarial attempts. User testing data was collected one day after the training data.

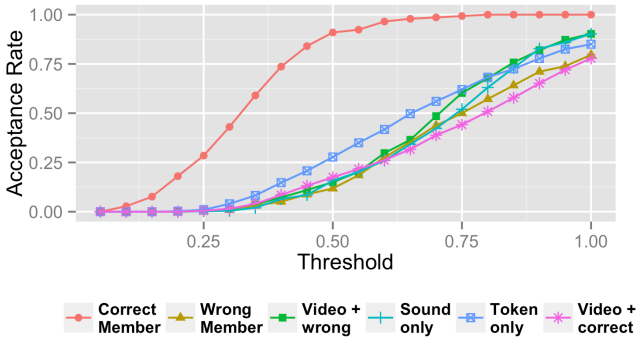

Figure 9. Acceptance rate as a function of minimum acceptable threshold across all thumprints. There is no threshold value to perfectly distinguish authentic attempts from adversarial attempts, but threshold values between 0.4 and 0.5 yield high true positives and low false positives.

Figure 9 shows a plot of acceptance rates for correctly identified group members ("correct member"), all four adversary types (*video+wrong, sound only, token only* and *video+correct*), as well as how often a user would be authenticated but misidentified as another member of the group (the "wrong member" trend line).

Expectedly, these results are not as optimistic as the analysis from our first study, when all data was collected from a single session and when the adversaries were not explicitly trying to exactly replicate the thumprint expression of a specific group member. One immediately notable result is that group members are rarely misidentified—this makes sense, as our preprocessing pipeline during training uses differences between group members to learn individual expressions of the thumprint.

Adversaries can have some success at cracking thumprints, particularly at higher thresholds. A good compromise between false positives and false negatives appears to occur in between the threshold values of 0.45 and 0.5. In between those thresholds, authentic user attempts are correctly let in between 85 and 91% of the time, while adversaries are granted acceptance between an average of 13% and 19% of the time. While these adversarial success rates seem high, they are comparable to other intentional behavioral approaches, such as TapSongs (83.2% user recognition, 19.4% adversarial acceptance) [36] and keystroke dynamics for user identification (83-92% recognition) [27].

Interestingly, what we believed was our "weakest" adversary model, the token only model, was most successful at cracking thumprints. This appears to be because adversaries with more information quickly honed in on how they would try to replicate the thumprint and simply repeated this process for all ten attempts. Token only adversaries, however, explored a wider space of possibilities with their 10 replications (i.e., they tried many different knocks as opposed to just one knock).

Finally, it is important to remember that Thumprint is not designed to provide perfect security against strong, motivated adversaries (who have advantages such as a video of the secret knock and ten unfettered attempts). We designed Thumprint to provide reasonable security, but emphasized inclusivity with identifiability, equitable distribution of responsibility and ease of sharing and revoking access. Indeed, for local group resources that are already largely physically secure (e.g., in homes), we believe our results suggest sufficient security.

It should also be noted that any probabilistic authenticator carries some risk of accidentally authenticating outsiders (e.g., even stronger, more sophisticated ones like Apple's TouchID [29]). Indeed, given the similarity in outsider rejection performance between our approach, TapSongs [36] and RhythmLink [25], this detection rate could be a natural limitation of using sensable behavioral interactions for authentication—at least using existing sensors and modeling techniques. Still, we argue that this level of outsider rejection is reasonable for the small, local-group setting, especially given our focus on inclusiveness.

DISCUSSION

Our evaluations suggest that groups of users who enter the same thumprint can reliably be distinguished from one another; that users can enter their thumprints fairly consistently over time; and, that casual but motivated adversaries are often detectable and can thus be protected against. Taken together, these results suggest that Thumprint is a promising step towards the vision of socially-inclusive authentication for small, local groups. This evidence does not, however, suggest that Thumprint is immediately ready for mainstream use.

Though immediate viability is often an objective of traditional authentication research, we believe that this objective can be short-sighted. Traditional authentication works well for the purpose of identifying individuals who access private accounts, but Thumprint, and any other form of

socially-inclusive authentication, is a significant departure from these models. Indeed, if the goal of traditional authentication is to create hard, impermeable boundaries that differentiate any two individuals, the goal of socially-inclusive authentication is to construct tweakable, semi-permeable boundaries between an in and out-group. While identifiability within the in-group is important, the process of identifying the individual should not raise hard barriers between those in the group.

Accordingly, while we have evaluated Thumprint to the standards expected of traditional authentication tools (e.g., with formally modeled adversaries), we believe our work opens up more interesting lines of inquiry. We reflect on some of these open questions and limitations, as well as discuss strategies for tackling them in future work.

Uncovering hidden group authentication needs

In designing Thumprint, we synthesized a number of unmet group authentication needs through a survey of the existing literature. However, these needs have only been explored in the socio-technical context of traditional authentication. As passwords and other typical forms of authentication have been long ingrained into everyday technology use, it may be difficult for users to conceptualize forms of authentication that are more group-friendly.

Accordingly, in future work, it would be pertinent to deploy Thumprint and other forms of socially-inclusive authentication as design probes in a field study with real groups. Through this field study, we may uncover additional insights into how local groups use socially-inclusive authenticators and how they can be improved.

Designing for group variety

While Thumprint was designed to better cater to the authentication needs of local groups, these groups can have tremendous variety in their structure, composition and broader social context [35]. Families, for example, typically have little to no churn and often have clear power structures. Groups of friends, on the other hand, may be more egalitarian and prefer equal access to collectively shared resources. Work teams may have a lot of churn, be short lived, or require compatibility with broader security infrastructures. Student organizations may have expensive equipment that should be sharable, but require audit logs to keep track of who had access to what.

Many other factors no doubt affect how appropriate solutions like Thumprint are for groups. For example, some groups may have greater risk perception than others (e.g., a group of journalists). Other groups may be aversive towards probabilistic authentication as opposed to deterministic authentication. Still other groups may value anonymity and want to do away with identifiability, while preserving an equitable distribution of security responsibility.

Thumprint, thus, is likely to better suited to the needs of some groups than others – it is not a panacea. Still, we believe it is a promising a step forward and could be a starting point for further explorations into the design space of socially-inclusive authentication for different groups.

Strength of security

Thumprint is not and was not designed to be perfectly secure. Though it is about as secure as comparable approaches for individuals (e.g., TapSongs [36], keystroke dynamics [27] and RhythmLink [25]), it is likely that a motivated adversary who observes individual group members entering the secret knock would be able to fool the model. Still, Thumprint's security may improve as more data from multiple time-separated sessions become available. As group members continue to use Thumprint for extended periods of time, there may be enough training data to employ these more sophisticated models (e.g., one-class classifiers) for stronger outsider rejection. In future work, we would like to explore this possibility.

CONCLUSION

In this paper, we designed and evaluated Thumprint, a socially-inclusive group authentication mechanism that authenticates and identifies group members through their expression of a shared secret knock. Specifically, we designed Thumprint to quickly and easily authenticate and identify individual members of a small group with a single shared secret. Through two user studies, we found that individuals who enter the same thumprint can be reliably distinguished from one another, that people can enter thumprints consistently over time, and that Thumprint provides reasonable security against a variety of casual, but motivated and capable adversaries. In light of these results, we believe that Thumprint is a promising step towards the vision of socially-inclusive authenticators that better meet the needs of small, local groups.

ACKNOWLEDGMENTS

This research was supported by the Qualcomm Innovation Fellowship. We thank Gini Keating, Olivier Benoit and Laura Dabbish for their helpful feedback.

REFERENCES

1. Lujo Bauer, Lorrie LF Cranor, RW Robert W Reeder, Michael K MK Reiter, and Kami Vaniea. 2007. *Comparing access-control technologies: A study of keys and smartphones*. Carnegie Mellon University CyLab Tech Report 07-005. Retrieved from http://repository.cmu.edu/cylab/46/
2. Mike Bond. 2005. The Dining Freemasons (Security Protocols for Secret Societies). In *Security Protocols*. Springer Berlin Heidelberg, 258–265.
3. Joseph Bonneau, Cormac Herley, Paul C Van Oorschot, and Frank Stajano. 2012. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. *Symposium on Security and Privacy (S&P'12)*, IEEE, 553–567. http://doi.org/10.1109/SP.2012.44
4. Anders Brandt. 2011. *Noise and Vibration Analysis: Signal Analysis and Experimental Procedures*. John

Wiley & Sons.

5. A J Bernheim Brush. 2012. It' s Used by Us: Family Friendly Access Control. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Workshop on Technology for Today's Family*.
6. L.F. Cranor and S. Garinkel. 2005. *Security and Usability: Designing Secure Systems that People Can Use*. O'Reilly Media.
7. Sauvik Das, LaToya Green, Beatrice Perez, Michael Murphy, and Adrian Perrig. 2010. *Detecting User Activities Using the Accelerometer on Android Smartphones*. Carnegie Mellon University.
8. Sauvik Das, Hyun Jin Kim, Laura A. Dabbish, and Jason I. Hong. 2014. The Effect of Social Influence on Security Sensitivity. *Proceedings of the 10th Symposium on Usable Privacy and Security (SOUPS'14)*.
9. Sauvik Das, Adam D.I. Kramer, Laura A. Dabbish, and Jason I. Hong. 2014. Increasing Security Sensitivity With Social Proof: A Large-Scale Experimental Confirmation. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*, ACM Press, 739–749. http://doi.org/10.1145/2660267.2660271
10. Sauvik Das, Adam D.I. Kramer, Laura A. Dabbish, and Jason I. Hong. 2015. The Role of Social Influence in Security Feature Adoption. *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15)*, ACM Press, 1416–1426. http://doi.org/10.1145/2675133.2675225
11. Serge Egelman, A.J. Bernheim Brush, and Kori M. Inkpen. 2008. Family accounts. *Proceedings of the ACM 2008 conference on Computer supported cooperative work (CSCW '08)*, ACM Press, 669. http://doi.org/10.1145/1460563.1460666
12. Usama M. Fayyad and Keki B. Irani. 1993. Multi-Interval Discretization of Continuos-Valued Attributes for Classification Learning. *Proc. International Joint Conference on Uncertainty in AI*, 1022–1027. Retrieved from http://trs-new.jpl.nasa.gov/dspace/handle/2014/35171
13. Shirley Gaw, Edward W Felten, and Patricia Fernandez-Kelly. 2006. Secrecy, flagging, and paranoia. *Proceedings of the SIGCHI conference on Human Factors in computing systems (CHI '06)*, ACM Press, 591–600. http://doi.org/10.1145/1124772.1124862
14. Eric Gilbert. 2015. Open Book. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*, ACM Press, 477–486. http://doi.org/10.1145/2702123.2702295
15. Mayank Goel, Jacob Wobbrock, and Shwetak Patel. 2012. GripSense. *Proceedings of the 25th annual ACM symposium on User interface software and technology (UIST '12)*, ACM Press, 545–554. http://doi.org/10.1145/2380116.2380184
16. Mark A. Hall. 1999. Correlation-based Feature Selection for Machine Learning. *University of Waikato*. http://doi.org/10.1080/01422419908228843
17. Eiji Hayashi, Sauvik Das, Shahriyar Amini, Jason Hong, and Ian Oakley. 2013. CASA: A Framework for Context-Aware Scalable Authentication. *Proceedings of the 9th Symposium on Usable Privacy and Security (SOUPS'13)*.
18. Cormac Herley and P van Oorschot. 2009. Passwords: If We're So Smart, Why Are We Still Using Them? *Proceedings of the 13th International Conference on Financial Cryptography and Data Security (FC'09)*. http://doi.org/10.1007/978-3-642-03549-4_14
19. Seong Seob Hwang, Sungzoon Cho, and Sunghoon Park. 2009. Keystroke dynamics-based authentication for mobile devices. *Computers and Security* 28, 1–2: 85–93. http://doi.org/10.1016/j.cose.2008.10.002
20. Steven J. Karau and Kipling D. Williams. 1993. Social Loafing: A Meta-Analytic Review and Theoretical Integration. *Interpersonal Relations and Group Processes* 65, 4: 681–706. http://doi.org/10.1037/0022-3514.65.4.681
21. Amy K Karlson, A.J. Bernheim Brush, and Stuart Schechter. 2009. Can i borrow your phone? *Proceedings of the 27th international conference on Human factors in computing systems (CHI 09)*, ACM Press, 1647–1650. http://doi.org/10.1145/1518701.1518953
22. Brendan Kiley. 2005. Secret Knocks and Passwords. *The Stranger*. Retrieved January 5, 2017 from http://www.thestranger.com/seattle/secret-knocks-and-passwords/Content?oid=25434
23. Ross Koppel, Sean Smith, Jim Blythe, and Vijay Kothari. 2015. Workarounds to Computer Access in Healthcare Organizations: You Want My Password or a Dead Patient? *Studies in Health Technology and Informatics* 208: 215–220. http://doi.org/10.3233/978-1-61499-488-6-215
24. Bibb Latané, Kipling Williams, and Stephen Harkins. 1979. Many hands make light the work: The causes and consequences of social loafing. *Journal of Personality and Social Psychology* 37, 6: 822–832. http://doi.org/10.1037/0022-3514.37.6.822
25. Felix Xiaozhu Lin, Daniel Ashbrook, and Sean White. 2011. RhythmLink: Securely Pairing I/O-Constrained Devices by Tapping Felix. *Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11)*, ACM Press, 263–271. http://doi.org/10.1145/2047196.2047231
26. Michelle L Mazurek, Brandon Salmon, Richard Shay, et al. 2010. Access control for home data sharing: Attitudes, needs, and practices. *Proceedings of the 28th*

international conference on Human factors in computing systems (CHI '10), ACM Press, 645–654. http://doi.org/10.1145/1753326.1753421

27. Fabian Monrose and Aviel D. Rubin. 2000. Keystroke dynamics as a biometric for authentication. *Future Generation Computer Systems* 16, 4: 351–359. http://doi.org/10.1016/S0167-739X(99)00059-X
28. Deborah A. Prentice, Dale T. Miller, and Jenifer R. Lightdale. 1994. Asymmetries in attachments to groups and to their members: Distinguishing between common-identity and common-bond groups. *Personality and Social Psychology Bulletin (PSPB)* 20, 5: 484–493.
29. Frank Rieger. 2013. Chaos Computer Club breaks Apple TouchID. Retrieved January 5, 2017 from https://www.ccc.de/en/updates/2013/ccc-breaks-apple-touchid
30. Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20, C: 53–65. http://doi.org/10.1016/0377-0427(87)90125-7
31. Bruce Schneier. 2000. *Secret & Lies: Digital Security in a Networked World*. John Wiley & Sons.
32. Supriya Singh, Anuja Cabraal, Catherine Demosthenous, Gunela Astbrink, and Michele Furlong. 2007. Password sharing. *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI '07)*, ACM Press, 895–904. http://doi.org/10.1145/1240624.1240759
33. Michael Toomim, Xianhang Zhang, James Fogarty, and James A Landay. 2008. Access control by testing for shared knowledge. *Proceeding of the Twenty-sixth annual CHI conference on Human factors in computing systems (CHI '08)*, ACM Press, 193–196. http://doi.org/10.1145/1357054.1357086
34. Gérard Vincent. 1991. A history of secrets? In *A History of Private Life: Riddles of Identity in Modern Times*. 145–281.
35. Stanley Wasserman and Katherine Faust. 1994. *Social network analysis: Methods and applications*. Cambridge University Press.
36. Jacob Otto Wobbrock. 2009. TapSongs. *Proceedings of the 22nd annual ACM symposium on User interface software and technology (UIST '09)*, ACM Press, 93–96. http://doi.org/10.1145/1622176.1622194
37. Shibboleth. *Wikipedia*. Retrieved January 5, 2017 from https://en.wikipedia.org/wiki/Shibboleth