Implicit Authentication through Learning User Behavior

Elaine Shi[1], Yuan Niu[2], Markus Jakobsson[3], and Richard Chow[1]

[1] Palo Alto Research Center
emails: eshi@parc.com, rchow@parc.com
[2] University of California, Davis
email: yniu@ucavis.edu
[3] FatSkunk
email: markus@fatskunk.com

Abstract. Users are increasingly dependent on mobile devices. However, current authentication methods like password entry are significantly more frustrating and difficult to perform on these devices, leading users to create and reuse shorter passwords and pins, or no authentication at all. We present implicit authentication - authenticating users based on behavior patterns. We describe our model for performing implicit authentication and assess our techniques using more than two weeks of collected data from over 50 subjects.

Keywords security, usability, implicit authentication, behavior modelling

1 Introduction

As mobile devices quickly gain in usage and popularity [20], more consumers are relying on these devices, particularly smartphones, as their primary source of Internet access [18, 22]. At the same time, continued and rapid increase of online applications and services results in an increase in demand for authentication. Traditional authentication via password input and higher-assurance authentication through use of a second factor (e.g., a SecurID token) both fall short in the context of authentication on mobile devices, where device limitations and consumer attitude demand a more integrated, convenient, yet secure experience [8].

Password-only authentication has been under attack for years from phishing scams and keyloggers. Using a second factor as part of the authentication process provides higher assurance. This practice is already mainstream within enterprises and is slowly entering the consumer market, but still has issues with usability and cost to overcome. Moreover, tokens like SecurID are primarily a desktop computing paradigm. The trend of separate devices like media players, Internet devices, e-readers, and phones consolidated into one device comes from a growing desire to carry fewer devices and is at odds with the idea of carrying a separate device specifically for authentication.

We propose *implicit authentication*, an approach that uses observations of user behavior for authentication. Most people are creatures of habit - a person goes to work in the morning, perhaps with a stop at the coffee shop, but almost always using the same route. Once at work, she might remain in the general vicinity of her office building until lunch time. In the afternoon, perhaps she calls home and picks up her child from school.

In the evening, she goes home. Throughout the day, she checks her various email accounts. Perhaps she also uses online banking and sometimes relies on her smartphone for access away from home. Weekly visits to the grocery store, regular calls to family members, etc. are all rich information that could be gathered and recorded almost entirely using smartphones.

For the reasons above, implicit authentication is particularly suited for mobile devices and portable computers, although it could be implemented for any computer. These devices are capable of collecting a rich set of information, such as location, motion, communication, and usage of applications and would benefit from implicit authentication because of their text input constraints. Furthermore, usage of the device varies from person to person [7] measured through battery and network activity. This information could be used to create an even more detailed profile for each user.

Implicit authentication could be used on medical devices, which sometimes share the input constraints of mobile devices, used to access and manipulate patient records. Account sharing is common on these devices, which could violate HIPAA requirements for patient privacy. Implicit authentication can help protect the privacy of patient records by authenticating users conveniently and in a timely manner under stressful situations. Military personnel, whose daily habits are more routine than the average user, can also benefit from the application of implicit authentication towards their equipment.

For general authentication needs, implicit authentication 1) acts as a second factor and supplements passwords for higher assurance authentication in a cost-effective and user-friendly manner; 2) acts as a primary method of authentication to replace passwords entirely; 3) provides additional assurance for financial transactions such as credit card purchases by acting as a fraud indicator. We note that in the latter scenario, the act of making the transaction may not require any user action on the device.

In this paper, we focus on mobile devices and present and evaluate techniques for the computation of an *authentication score* based on a user's recent activities as observed by the device in her possession. We also describe an architecture for implicit authentication and discuss our findings from tracking habits of over 50 users for at least two weeks each. Our experimental results demonstrate the power of combining multiple features for authentication. We also compile a taxonomy of adversarial models, and show that our system is robust against an informed adversary who tries to game the system by polluting a feature.

Our method for scoring is based on identification of positive events and boosting the score when a ”good“ or habitual event is observed (e.g., buying coffee at the same shop in a similar time every day) and lowering the score upon detection of negative events. Negative events could include those not commonly observed for a user, such as calling an unknown number, or an event commonly associated with abuse or device theft, such as sudden changes in expected location. The passage of *time* is treated as a negative event in that scores gradually degrade. When the score falls below a threshold, the user must explicitly authenticate, perhaps by entering a passcode, before she can continue. Successfully authenticating explicitly will boost the score once more. The threshold may vary for different applications depending on security needs.

1.1 Related Work

Two common approaches to addressing user management of an increasing number of service credentials exist: 1) reducing the number of times the user needs to authenticate and 2) biometrics.

Solutions such as Single Sign-On (SSO) and password managers may reduce the problem of frequent authentication, but they do not identify the user but rather the *device*. Therefore, SSO does not defend well against theft and compromise of devices, nor does it address voluntary account sharing.

According to a study on user perception of authentication on mobile devices, Furnell et al. [8] found that users want a transparent authentication experience that increases security and "authenticates the user continuously/periodically throughout the day in order to maintain confidence in the identity of the user". These users were receptive to biometrics and behavior indicators, but not to security tokens.

Some form of *implicit* authentication exists already in the form of location-based access control [19, 5], or biometrics, notably keystroke dynamics and typing patterns [15, 14, 17]. However, these methods are not easily translatable to mobile devices which possess significantly different keyboards and often provide auto-correction or auto-complete features. More recently, accelerometers in devices have been used to profile and identify users. Chang et al. [4] used accelerometers in television remote controls to identify individuals. Kale et al. [13] and Gafurov et al. [9] used gait recognition to detect whether a device is being used by the owner.

These biometrics and location-based approaches are complementary to our work, as implicit authentication can potentially utilize biometrics as features in computing the authentication score.

The convergence of multiple sources of data is another approach. Combination of multiple biometric factors was used to generate the authentication decision or score [3, 2]. Greenstadt and Beale [10] noted a need for "cognitive security" in personal devices. Specifically, they proposed a multi-modal approach "in which many different low-fidelity streams of biometric information are combined to produce an ongoing positive recognition of a user". Our efforts are a step forward in the direction of realizing the vision laid out in [10] and expand our preliminary work [12].

2 Adversarial Models

Adversaries vary in terms of roles, incentives, and capabilities.

Roles. *Strangers* may steal the legitimate user's device or find it in a public place. *Friends or co-workers* may happen to get hold of the legitimate user's device. Similarly, *family members* may get hold of the legitimate user's device. *Enemies or competitors* may try to reap information from the victim's device, e.g., in the case of political or industrial espionage.

Incentives. A *financially-driven* adversary may wish to use resources associated with the captured device. Such an adversary can benefit in the following ways:

(a) She can make free phone calls or browse the web for free (but at the owner's expense) or simply use the device as a free recreational device.
(b) She may try to gain access to sensitive information such as the user's SSN and bank account, and reap benefits through means of identity theft or stealing money from the user's bank account.
(c) She can derive benefit from resale of the device or hardware components.

An adversary may be driven by *curiosity*. Nosy friends, co-workers or family members may try to read the legitimate user's emails or SMS messages, or examine her browsing or phone call history. In the case of espionage, the adversary may try to obtain sensitive information from the device or through the ability to access the owner's accounts. We refer to such an adversary as the curious adversary.

An adversary may be motivated by the desire to perform *sabotage*, in which the adversary does not reap any financial gain directly, but causes harm to the victim. For example, the adversary can log onto a social network as the legitimate owner, and publish embarrassing remarks.

An adversary may wish to use the device to protect her *anonymity*, for example, if the adversary is involved in drug dealing, terrorist or other illegal activities.

Capabilities. The *uninformed* adversary is unaware of the existence of implicit authentication. After capturing the device, an uninformed adversary is likely to use the device as her own, or use it in a straightforward way to achieve various incentives as described in Section 2.

The *informed* adversary is aware of the existence of implicit authentication, and may try to game the system. For example, the adversary can try to immitate the legitimate user's behavior. It may be easier for the adversary to immitate the legitimate user's behavior is she is a friend or family member of the legitimate user. The same task is harder for a stranger who steals the user's device or picks it up in a public place. The informed adversary's power depends on how many features she can pollute.

A more powerful adversary may infect the device with *malware*. Persistent malware is able to control and observe all events over time and could potentially mimic the victim's behavioral patterns. It is also possible for the malware to log keystrokes to gain access to sensitive account information, such as login credentials for banks. Malware defense is outside the scope of this paper, and should be seen as an orthogonal issue. We refer readers to [11] for a treatment of mobile malware defenses.

3 Data Sources and Architecture

Implicit authentication can potentially employ a wide variety of data sources to make authentication decisions. For instance, modern smartphones provide rich sources of behavioral data, such as: 1) Location and (potentially) co-location. 2) Accelerometer measurements. 3) WiFi, Bluetooth or USB connections. 4) Application usage, such as browsing patterns and software installations. 5) Biometric-style measurements, such as typing patterns and voice data. 6) Contextual data, such as the content of calendar entries. 7) Phone call patterns. Also, auxiliary information about the user might be another source of data for implicit authentication. For instance, a user's calendar held in the cloud could be used to corroborate mobile phone data.

The system architecture determines whether part or all data collected is saved on the device, in the cloud, or held by the service provider or carrier. We explain some possible architectural choices and discuss their pros and cons.

The mobile device itself can make authentication decisions to decide whether a password is necessary to unlock the device or use a certain application. In this case, data can be stored locally, advantageous for privacy. One can also use local authentication to access a remote service, for instance by using the SIM card to sign and send an authentication decision (or score) to the service provider. While this approach protects the user's privacy, it is not safe against theft and corruption of devices. If the device is captured, an attacker may be able to obtain the data stored in the memory and learn the user's behavioral patterns. As mobile devices are battery- and storage-constrained, the authentication score must be efficient to compute.

Another possibility is that a trusted third party be in charge of making authentication inferences and communicating trust statements to qualified service providers. For example, a carrier, who has access to much of the data needed for implicit authentication and has already established a trust relationship with the consumer, could naturally serve in the trusted third party role. It is also possible for carriers to provide data to third parties entrusted with the analysis of data and the making of authentication decisions.

In all approaches, even with data held locally, there is potential for a privacy breach. Some mitigating measures would be: 1) removing identifying information (such as names or phone numbers) from the data being reported; 2) use a pseudonym approach, e.g., "phone number A, location B, area code D"; 3) use coarse-grained or aggregate data, e.g., reporting a rough geographic location rather than precise coordinates, and reporting aggregate statistics rather than full traces. We will see later that one can adopt measures like these and yet still retain utility for implicit authentication purposes. It would also be interesting to investigate how to apply differential privacy techniques [6].

4 Algorithm

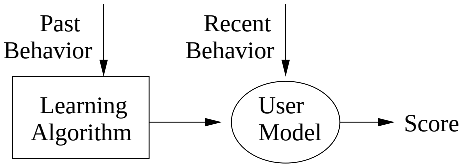


Fig. 1: Architecture.

Figure 1 outlines the framework of the machine learning algorithm. We first learn a *user model* from a user's past behavior which characterizes an individual's behavioral patterns. Given a user model and some recently observed behavior, we can compute the probability that the device is in the hands of the legitimate user. We use this probability as an *authentication score*. The score is used to make an authentication decision:

typically, we can use a threshold to decide whether to accept or reject the user, and the threshold can vary for different applications, depending on whether the application is security sensitive. The score may also be used as a second-factor indicator to augment traditional password-based authentication.

4.1 Modelling User Behavior

The user model should characterize the user's behavioral patterns. For example, how frequently the user typically makes phone calls to numbers in the phone book, where the user typically spends time, etc.

Modelling independent features. We consider an independent feature model by assuming independence between features such as phone calls, location, browser usage, etc.

In general, the user model may also consider combinations of different indicators. For example, given that the user is in her office and has received a call from number A, then with 90% probability, she will send an email to address B in the next 10 minutes.

Dependency between these features is left to future work because we have only $1 \sim 2$ weeks of training data for each user, and this is insufficient to model complex dependencies between features – an attempt to do this may easily result in overfitting.

A user's behavior typically depends on the time of day and day of week. For example, one user might place and receive frequent phone calls in the afternoon, but she may not have much phone activity at night, e.g., between 11pm and 8am. People are generally at work in the same location on weekdays, but their locations vary on weekends. In our experiments, we only model the time-of-day effect, as the scale of data ($1 \sim 2$ weeks for each user) is insufficient for us to model the day-of-week effect.

In our approach we study k independent features. Each feature can be represented by a random variable, $V_1, V_2, \ldots, V_k$. Example of features include:

$$\begin{aligned} V_1 &:= \textit{time elapsed since last good call} \\ V_2 &:= \textit{number of times bad calls occur per day} \\ V_3 &:= \textit{GPS coordinates} \end{aligned}$$

In the above example, a good call is a call made to or received from a known number such as a number from the contact list. By contrast, a bad call is a call made to an unknown number.

A user model is the combination of k probability density functions conditioned on the variable $T =$ (*time of day, day of week*):

$$\text{user model} := \left[p(V_1|T), p(V_2|T), \ldots, p(V_k|T)\right]$$

The learning algorithm in Figure 1 basically estimates the density functions for each feature conditioned on the time-of-day and day-of-week, thereby forming a user model.

4.2 Scoring Recent Behavior

Given a user model and reported recent behavior, the scoring algorithm outputs a score indicating the likelihood that the device is in the hands of the rightful owner.

Scoring independent features. A user's recent behavior may be described by a tuple

$$(t, v_1, v_2, \ldots, v_k)$$

where t denotes the current time, and $v_1, \ldots, v_k$ denote the values of variables $(V_1, \ldots, V_k)$ at time t. Assume features $V_1, \ldots, V_k$ are independent, then we define the score to be probability (or probability density function) of observing recent behavior $v_1, \ldots, v_k$ at time t. As we assume independence between features, the probability can be computed by multiplying the probabilities for each individual feature:

$$\text{score} := p(v_1|t) \cdot p(v_2|t) \cdot \ldots \cdot p(v_k|t)$$

For convenience, we overload the variable t above to denote the time-of-day and day-of-week pair corresponding to the current time t.

4.3 Selection of Features

We extract several features from the data collected. These features fall within three categories: 1) frequency of good events; 2) frequency of bad events; 3) location. Note that although not studied in this paper, one can also incorporate numerous other features into our implicit authentication system, such as the user's typing patterns, calendar, accelerometer patterns, etc. We now explain how we model the above-mentioned categories of features.

Frequency of good events. Good events are events that positively indicate that the device is in the hands of the legitimate user, for example, making a phone call to a family member, or browsing a familiar website.

To model the frequency of good events, we consider the feature G := *time elapsed since last good event*. Within this category, we consider three sub-features,

G_{phone} := *time elapsed since last good phone call*
G_{sms} := *time elapsed since last good SMS sent*
$G_{browser}$:= *time elapsed since last good website visited*

The above features allow us to implement the idea that the authentication score should decay over time if no good events occur. Moreover, the rate of decay depends on the time of day and day of week. If a user typically makes frequent phone calls in the afternoon, then the score should decrease faster in the afternoon during periods of inactivity. By contrast, if the user typically makes no phone calls between 12am and 8am, then the score should decrease more slowly over this period of time.

In the training phase, we learn the distribution of the variables $G_{phone}, G_{sms}, G_{browser}$ conditioned on the time-of-day. During the scoring phase, suppose that at time t, the lapse since the last good event is x minutes. Then the score for this feature at time t can be computed as below:

$$S_G(x, t) := \Pr[G \geq x | T = t] \qquad (1)$$

Basically, the score for this feature is the likelihood that one sees a lapse of x or longer since the last good event around time t of the day. In our implementation, we use the empirical distribution and a piecewise linear function to estimate probabilities.

Frequency of bad events. Bad events are those that negatively indicate that the legitimate owner is using the device. For example, making a call to an unknown number or visiting an unknown URL is a bad event. We consider the feature B := *number of bad events within the past k hours*, for example,

B_{phone} := *number of bad phone calls in the past k hours*
B_{sms} := *number of bad SMS msgs sent in the past k hours*
$B_{browser}$:= *number of bad URLs visited in the past k hours*

In the training phase, we learn the distribution of the variables B_{phone}, B_{sms} and $B_{browser}$. In the scoring phase, suppose the number of bad events within the past k hours is x at time t. We compute a score for this feature as below, indicating the likelihood of seeing at least x bad events within the past k hours.

$$S_B(x,t) := \Pr[B \geq x | T = t]$$

As bad events occur less frequently in the training data than good events, and their correlation with time-of-day is also less significant, we did not model the time-of-day effect for bad events. In the implementation, we let $k = 24$ hours, that is, 1 day.

Location. We model a user's location using GPS coordinates collected. We also collected the user's wifi connections and cellular tower IDs which approximate the user's location. However, these features are sporadic and less fine-grained than GPS coordinates, and represent a subset of information provided by GPS. Therefore, we only use GPS coordinates as part of our user model.

We use a Gaussian Mixture Model (GMM) [16] to model a user's location around a certain time-of-day. We divide the day into 6 hour time epochs, and use the standard Expectation Maximization (EM) algorithm [1] to train a GMM model for a user during each 6-hour epoch. The GMM algorithm is able to output directional clusters, for example, when the user is traveling on a road or highway. Figure 2 shows an example of the results produced by the clustering algorithm. The traces plotted represent one user's GPS trace between 7pm-9pm each day over the entire training period. We obliterated the concrete coordinates to protect privacy of the user.

We use the notation L to denote the random variable corresponding to a user's GPS location. Given the trained GMM model, we can compute the score S_L if the user appears at location x at time t.

$$S_L(x,t) := \mathsf{GMM\_pdf}_t(x)$$

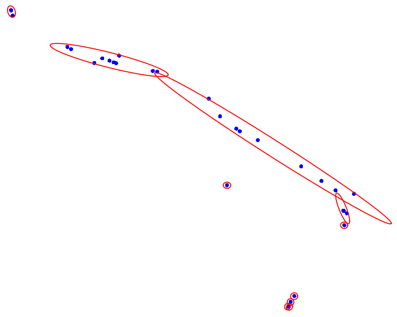
where $\mathsf{GMM\_pdf}_t$ denotes the probability density function of the user's GMM model around time t of the day.

5 Experiment Design

5.1 Data Collection

As mentioned in Section 3, data used for implicit authentication can come from diverse sources in a real deployment. To study the feasibility of this approach, we developed a data collection application which we posted in the Android Marketplace. We recorded the following types of activities from each user:

Fig. 2 Clusters of a user's GPS trace. The blue dots represent the user's traces in a two-hour epoch over multiple days, and the red ellipses represent the clusters fitted. The major two directional clusters correspond to the user's trajectory on a highway.



SMS: We recorded the time and direction (incoming or outgoing) of the message and the obfuscated phone number.

Phone Calls: We recorded the obfuscated phone number in contact, whether the call was incoming or outgoing, the time the call started, and the duration of the call.

Browser History: We pulled browser history and recorded the obfuscated domain name of each URL and the number of visits to this URL done previously.

Location: We recorded the GPS coordinates if available, only if users enabled it. For this reason, we also collected coarse location calculated and provided by the Android OS. We also recorded the obfuscated SSID if a user is connected to a WiFi network and IDs of cellular base stations as backups.

Ethics. To protect user privacy, we obfuscated phone numbers, SSIDs, and URLs using a keyed hash. The key for each device was randomly generated at install time and stored only on the device. All hashing was performed on the device. As a result of the obfuscation, we can identify instances of the same phone number or URL on each user's log, but we are unable to determine whether two users have overlapping contacts or URLs. We did not collect the user's contact list. In our analysis, we use numbers seen before as an approximation of "good" numbers for both phone and SMS. Similarly, we regard websites visited before as "good" URLs. We also allowed users to specify intervals of time and the data to delete for those intervals.

5.2 Modelling Adversary Behavior

Uninformed Stranger An uninformed stranger would typically use the captured device as her own. To simulate such an adversary, we use a *splicing* approach – we choose a user to be the legitimate user, and another user to be the adversary. We splice a segment of the legitimate user's trace with a segment of the adversary's trace. We use the term *splicing moment* to refer to the point at which the two traces are concatenated. In practice, the splicing moment can be regarded as the time of device capture.

Splicing browser history. As mentioned before, whenever a phone call, SMS or browsing event occurs, our system needs to judge whether the event is good or bad. We deem phone numbers or URLs seen earlier to be good, and otherwise bad. As mentioned earlier, the phone numbers and URLs collected are anonymized using a keyed hash, where the key is randomly selected and different (with high probability) for each user. We can safely assume that a stranger would make calls and send SMS messages to a set of numbers disjoint from the legitimate user. Therefore, we consider all calls and SMS messages after the splicing moment to be bad. However, for browsing history, it may not be realistic to do so, as many users visit a common set of popular websites, such as Google or New York Times.

To address this issue, we use a lower-bound and upper-bound approach. We lower-bound the adversary's advantage by assuming that all websites visited after the splicing moment are bad. We then upper-bound the adversary's advantage by assuming that all websites visited after the splicing moment are good.

Splicing location. The users we recruited in the study come from all over the world. It would not make sense to splice the location trace of a user in San Francisco with that of a user in Chicago. In particular, at the time of device capture, the adversary and the legitimate user are likely to be in the vicinity of each other.

To address this issue, we model an adversary who lives and works in the vicinity of the legitimate user. At the splicing moment, we compute the delta between the location of the the legitimate user and that of the adversary. We then add this delta to the adversary's traces after the splicing moment. This ensures that the adversary and the legitimate user are in the same location at the time of the device capture. We also assume that the adversary starts to move within 1 hour after the capturing the device. If the adversary's trace remains stationary after the splice, we look ahead into the future to find a point when the adversary starts moving, and we shift that part of the trace forward, so that the adversary starts moving 1 hour after the device capture.

One possible objection is that the nature of the location would change after adding a delta. For example, a hypothetical user could be walking in the bay after the splice. This is not a problem for us, as our GMM model only utilizes GPS coordinates and is agnostic to any auxiliary information such as whether a pair of coordinates corresponds to a highway, a residential area or a business. Indeed, harvesting such auxiliary information may help improve the performance of implicit authentication. We leave this as one interesting direction for future work.

Informed Stranger An informed adversary may try to imitate the behavior of the legitimate user. The adversary's power largely depends on how many features she can pollute. Consider an adversary who can pollute the feature $G_{browser}$, as mimicking the legitimate user's browsing history seems to be easier than mimicking other features we selected. For example, the adversary may hesitate to call or SMS the legitimate user's contacts for fear of being exposed. The adversary may also find it difficult to fake the GPS trace to be consistent with the legitimate user's behavior.

Specifically, suppose the adversary can examine the legitimate user's browser history, and generate good browsing events regularly to keep herself alive with the device. Typically, the adversary's goal is to evade detection as long as possible so she can consume the resources associated with the device. So we assume that on top of the keep-alive work, the adversary uses the device following her regular patterns through which she achieves utility.

We model a *mild* and an *aggressive* adversary against the $G_{browser}$ feature. The mild adversary issues a good browser event with every usage of the device. Basically, we splice the legitimate user and the adversary's traces, and add a good browser event for every event in the adversary's trace. The aggressive adversary can maximally pollute the $G_{browser}$ feature. Suppose that the adversary can perform such keep-alive work with sufficient frequency, then she can always obtain full score for the $G_{browser}$ dimension.

Note that while we choose the $G_{browser}$ feature for the adversary to tamper with, this study can in general shed light on the robustness of our system against an adversary capable of polluting one feature, even when she can maximally pollute that feature.

5.3 Detailed Experimental Design

Among the 276 users who downloaded our data collection program, we select roughly 50 users who participated over a period of 12 days or longer, and contributed at least two categories of data among phone, SMS, GPS, and browsing history. We use this set of roughly 50 users to model legitimate users.

We are able to leverage more users' data when modelling the adversary. Since we need not train on the adversary's data, the requirement on the length of the participation is less stringent. Specifically, we select users who have participated over a duration of 3 days or longer, and contributed at least 2 categories of activities.

For each legitimate user selected, we use the first 60% (approximately) for training, and the remaining for testing. We train the user's behavioral model using the training set. We select various thresholds and use the testing set to study how long the legitimate user can continue using the device before a failed authentication. Then we splice each legitimate user's data with each adversary's data, and we study how long the adversary can use the device before a failed authentication (i.e., the adversary is locked out.)

In our experiments, we use the features G_{phone}, $G_{browser}$, G_{sms}, B_{phone}, $B_{browser}$, B_{sms} and L, representing the time elapsed since the previous good phone/browser/SMS events, number of bad phone/browser/SMS events within the past 24 hours, and the GPS location respectively.

6 Results

We evaluate our algorithm using the following metrics. Define the following variables:

X:= *number of times the legitimate user used the device before a failed authentication*
Y:= *number of times the adversary used the device before detection*

If two consecutive events are at least 1 minute apart from each other, we regard them as two different usages. Depending on policy settings, the user may be asked to enter her password when implicit authentication fails to authenticate the user.

We plot X against Y to demonstrate the effectiveness of implicit authentication in distinguishing the legitimate user from the adversary.

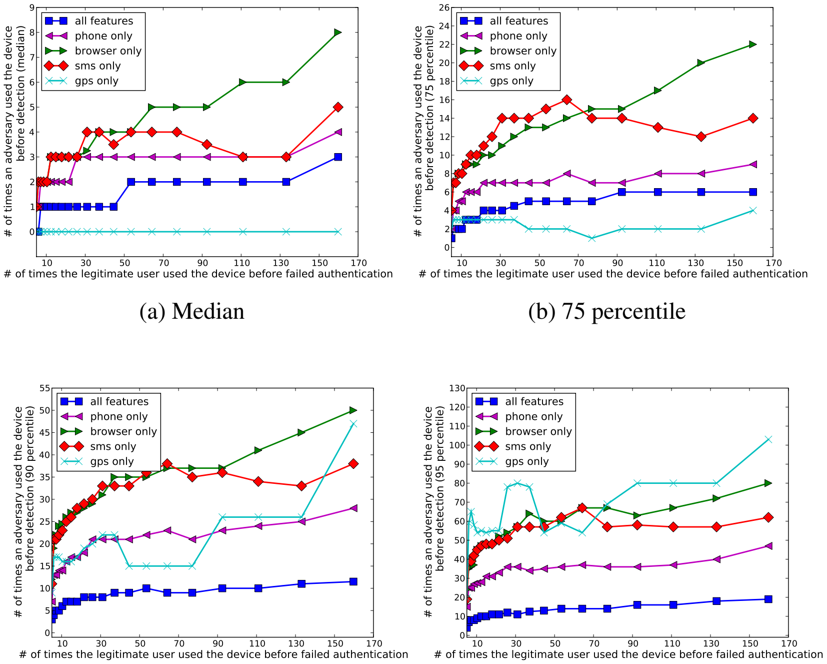
An alternative metric is time till a failed authentication (or detection). However, our system computes scores only when the authentication decision is needed – we assume that an authentication decision is needed whenever the user tries to use the device. Therefore, the metric "time till a failed authentication" depends heavily on when and how often the user uses the device, and this varies significantly for different users. We use the metric "number of usages till a failed authentication" as it is independent of how frequently the user uses the device, and thus a more informative and uniform measure for different users.

6.1 Power of Fusing Multiple Features

Figure 3 shows the power of combining multiple features. The four sub-figures plot the median, 75, 90, and 95 percentiles of the variable Y. Note that the x-axis is sparse, namely, not every integer value for x has a data point. As the x-axis becomes sparser as x grows, we group the points on the x-axis into bins centered at $x = 5 \times 1.2^k$ for $k = \{0, 1, \ldots, 19\}$. Values $x \geq 5 \times 1.2^{19} \simeq 160$ are grouped into the last bin. We then evaluate the median and various percentiles of each bin.

The five different curves represent using each feature alone, and combining all features. Taking the "all features" curves as an example, one can interpret the curves as below: if we set the threshold such that the legitimate user can use the device roughly 100 times before a failed authentication, then with 50% probability, the adversary will be locked out after using the device at most twice. With 75% probability, the adversary will be locked out after 6 or fewer usages of the device. With 90% probability, the adversary will be locked out after 10 or fewer usages of the device. With 95% probability, the adversary will be locked out after 16 or fewer usages of the device. Note that the curves are not monotonic due to the fact the x-axis is sparse.

It is not hard to see that combining different features is more powerful than using each single feature alone. Among all features considered, the GPS feature is the most interesting: while it performs better than all features combined in the median and 75 percentile plots, it has higher variance as demonstrated by the 90 and 95-percentile plots. This is expected for the following reasons. In the experiments, we assume that the adversary make phone calls and sends SMS messages to a set of numbers distinct from the legitimate user. Similarly, the adversary visits a different set of websites than the legitimate user. (In the paragraph below, we will explain how to correct for potential bias introduced by the browser feature.) Therefore, these features exhibit lower variance in how well they distinguish the legitimate user from the adversary. As for the GPS feature, however, the adversary remains in the vicinity of where the device is captured. If the legitimate user usually appears in very few places, e.g., work and home, then the clusters generated typically will tightly fit around these locations, and the adversary is likely to have a lower score. By contrast, if the legitimate user's trace is more disperse, then the clusters generated typically fit more loosely – and if the adversary is in the vicinity, her score will be relatively high.



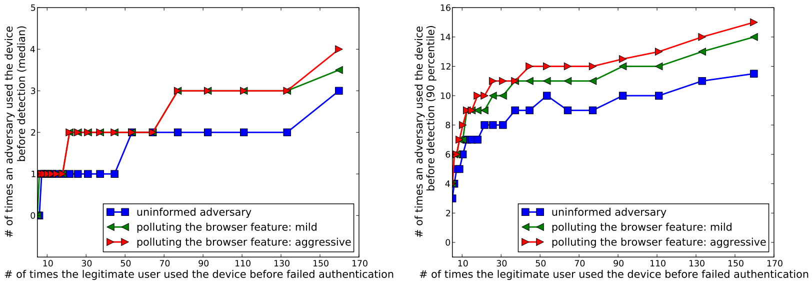
(a) Median

(b) 75 percentile

(c) 90 percentile

(d) 95 percentile

Fig. 3: **Fusion of multiple features.** The combination of multiple features allows us to better distinguish a legitimate user from an adversary. Please refer to the first paragraph of Section 6.1 for details on how the percentiles are computed.



(a) Median.

(b) 90 percentile

Fig. 4: **Informed stranger.** The figure demonstrates that our system is robust against an adversary capable of polluting one feature.

Correcting for bias from the browser feature. As mentioned in Section 2, it is more tricky to splice browser traces than other traces, as there exists a set of popular websites which everyone visits. In Figure 3, we assume that the adversary visits sites distinct from the legitimate user. This may not be realistic in practice, so we correct for the bias using a lower and upper bound approach. Due to space constraints, we defer the results to the full online version [21].

6.2 Stronger Adversarial Models

So far we have assumed that our adversary is an uninformed stranger who uses the device as her own after capturing it. We now consider stronger adversarial models, in particular, an informed stranger, who is aware of the existence of implicit authentication and tries to game it. As explained in Section 5.2, we study an adversary who can pollute the $G_{browser}$ feature by mimicking the legitimate user's browser patterns. One realistic attack is to generate a good event with every usage of the device – we refer to this adversary as the *mild* adversary. We also consider an adversary who aggressively generates good browser events such that she can maximally pollute the $G_{browser}$ feature and always obtain full score on that feature – we refer to this adversary as the *aggressive* adversary. Such an adversary can shed light on how robust our system is against adversaries that can optimally pollute a single feature. Section 2 contains more details on how we model such adversaries.

Figure 4 demonstrates the respective advantages of the mild and the aggressive adversaries. On the y-axis we plot the number of times the adversary can use the device before she is locked out. This number of usages does not include the keep-alive browser events generated by the adversary, as these events are regarded as "work" that the adversary must perform to keep herself alive, and do not create utility for the adversary. For comparison, we also plot the curve for the uninformed adversary. This figure demonstrates that our algorithm is fairly robust to the pollution of a single feature. The aggressive adversary performs only marginally better than the mild adversary, as the $G_{browser}$ score is typically quite high already for the mild adversary.

7 Future Work

We dealt primarily with adversarial models of strangers in our experiments. An important future task is to develop models for other types of adversaries, including friends and family members. To get the data necessary, we can recruit the help of family members and friends to study how to game and defend the implicit authentication system.

Another interesting direction is to incorporate more features into implicit authentication. The following are promising candidates: 1) Contextual information, either available from the phone itself or available in the cloud, such as emails, calendars and address books. Mining such contextual information can allow us to predict the legitimate user's whereabouts and activities. For example, if a user living in California has previously booked a flight to New York, then showing up in New York at the expected time should not be treated as an anomaly. If a user's calendar suggests that she has an important meeting with a client, then showing up in a grocery store instead at that time indicates an anomaly. 2) Biometrics. We are particularly interested in biometrics that can be utilized without involving explicit actions from the user. For example, accelerometer or touch-screen biometrics can be collected from the user's normal activities, and are thereby more desirable than asking the user to explicitly swipe her fingerprint for authentication. 3) Other data available from the device, such as application installation and usage patterns, airplane mode, and synchronization patterns with a computer.

Acknowledgment(s)

We gratefully acknowledge Philippe Golle and David Goldberg who made insightful suggestions to the core algorithm. We thank Oliver Brdiczka and Jessica Staddon for helpful discussions.

References

1. em, A Python Package for Learning Gaussian Mixture Models with Expectation Maximization. `http://www.ar.media.kyoto-u.ac.jp/members/david/softwares/em/`.
2. J. Bigun, J. Fierrez-Aguilar, J. Ortega-Garcia, and J. Gonzalez-Rodriguez. Combining biometric evidence for person authentication. In *Advanced Studies in Biometrics*, 2005.
3. R. Brunelli and D. Falavigna. Person identification using multiple cues. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995.
4. K. Chang, J. Hightower, and B. Kveton. Inferring identity using accelerometers in television remote controls. In *International Conference on Pervasive Computing*, 2009.
5. M. L. Damiani and C. Silvestri. Towards movement-aware access control. In *SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS*, 2008.
6. D. Dwork. Differential privacy: A survey of results. In *TAMC*, 2008.
7. H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in smartphone usage. In *MobiSys*, 2010.
8. S. Furnell, N. Clarke, and S. Karatzouni. Beyond the pin: Enhancing user authentication for mobile devices. *Computer Fraud and Security*, 2008.
9. D. Gafurov, K. Helkala, and T. Søndrol. Biometric gait authentication using accelerometer sensor. *JCP*, 1(7):51–59, 2006.
10. R. Greenstadt and J. Beal. Cognitive security for personal devices. In *AISec*, 2008.
11. M. Jakobsson and A. Juels. Server-side detection of malware infection. In *NSPW*, 2009.
12. M. Jakobsson, E. Shi, P. Golle, and R. Chow. Implicit Authentication for Mobile Devices. In *HotSec*, 2009.
13. A. Kale, N. Cuntoor, and V. Krüger. Gait-Based Recognition of Humans Using Continuous HMMs. In *FGR*, 2002.
14. G. Leggett, J. Williams, and M. Usnick. Dynamic identity verification via keystroke characteristics. In *International Journal of Man-Machine Studies*, 1998.
15. F. Monrose and A. Rubin. Authentication via keystroke dynamics. In *CCS*, 1997.
16. A. Moore. Lecture Notes: Gaussian Mixture Models. `http://www.cs.cmu.edu/afs/cs/Web/People/awm/tutorials/gmm.html`.
17. M. Nisenson, I. Yariv, R. El-Yaniv, and R. Meir. Towards behaviometric security systems: Learning to identify a typist. In *PKDD*, 2003.
18. L. Rainie and J. Anderson. The Future of the Internet III. `http://www.pewinternet.org/Reports/2008/The-Future-of-the-Internet-III.aspx`.

19. N. Sastry, U. Shankar, and D. Wagner. Secure verification of location claims. In *WiSe '03: Proceedings of the 2nd ACM workshop on Wireless security*, 2003.
20. S. Schroeder. Smartphones Are Selling Like Crazy. `http://mashable.com/2010/02/05/smartphones-sales/`.
21. E. Shi, Y. Niu, M. Jakobsson, and R. Chow. Implicit Authentication through Learning User Behavior. Full online version available at `http://midgard.cs.ucdavis.edu/~niu/papers/isc2010full.pdf`.
22. L. Whitney. Smartphones to dominate PCs in Gartner forecast. `http://news.cnet.com/8301-1001_3-10434760-92.html`.