USR: Enabling Identity Awareness and Usable App Access Control During Hand-free Mobile Interactions

Tao Feng, Zhimin Gao, Dainis Boumber, Tzu-Hua Liu, Nicholas DeSalvo, Xi Zhao and Weidong Shi
Computer Science Department, University of Houston
Email: tfeng3@cs.uh.edu

***Abstract*—We propose a new approach for improving user experience and privacy protection during human-mobile speech interaction by considering factors, such as user identity, application privacy level, usable app access control, and application function class. To integrate these factors into speech recognition on the mobile device, we design and implement a unified speech-speaker recognizer (USR) framework, which will recognize both speech content and speaker identification, and respond accordingly. This USR framework consists of an application interface, a speaker recognition module, a speech recognition module, and an identity management module. A comparison study was conducted contrasting the benefits and limitations of USR framework to the established original speech recognition and app access control framework on mobile device, such as Google Voice, and AppLock. Our results show that, while USR framework intuitively improved mobile privacy by serving only the phone owner for specific customized applications, USR framework was also able to provide better user experience across a number of tasks.**

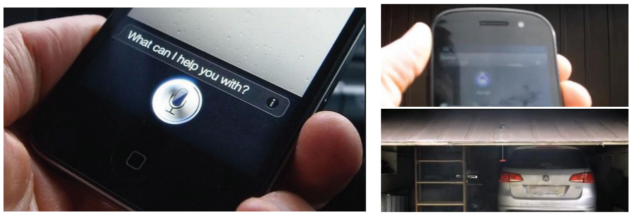
Keywords*—Mobile Device; Privacy, Usability, Identity Management, App Access Control

I. Introduction

The commercialization of Automatic Speech Recognition (ASR) technologies has ushered in a new era of hands-free user-mobile device interactions. Unlike device adaptive interactions (e.g., touch gestures, typing), speech is a more intuitive, inter-personal communication medium [1]. This, coupled with the release of speech recognition services (e.g., Google Voice [2] and third party APIs [3], [4]), explains the surge in popularity of speech based applications.

However, the popularity of speech recognition exposes its users to privacy vulnerabilities. Research has mainly focused on accuracy (i.e., “what the user is speaking”) but not on identity management (i.e., “who is speaking”). This can significantly simplify the attackers task of accessing sensitive information *e.g.*, confidential documents, emails, contact lists) stored on victim’s devices, or even allow the attacker to impersonate the user in highly sensitive operations (*e.g.*, posting status updates on social networks, initiating e-mail, SMS, or voice calls). Furthermore, we have shown that Siri and Google Voice Actions [5], [6] can bypass the login stage

authentication and follow requests of unauthorized users (see Figure 1). They may even enable unauthorized remote control applications. Although there are some app access control applications that may help improve user app security and avoid the aforementioned problems, mostly they are not easy to use and configure. This is due to the additional efforts of entering a password each time to access a locked app, as well as management of the locked app list. In addition to protecting user privacy, speech based identity management can also promote mobile user experiences by allowing users to customize their app access control and function by voice commands.



(a) Bypass the login (b) Remote garage door control

Fig. 1. Potential violations from current speech recognition framework: (a) passcode can be bypassed by Siri and Google Voice Actions, sensitive operations (*e.g.*, making phone calls, sending text messages, posting status) can be performed as if in an unlocked status; (b) sensitive applications such as garage door control do not have an identity authentication and opens to whomever holds the smart phone.

Previous work on speech recognition [2], [7], [8] and speaker recognition [9], [10] has been applied on mobile devices, performing either implicit, or explicit user authentication. However, to the best of our knowledge, no prior work has been jointly performed on a mobile speaker and speech recognition task as well as being implemented as an identity awareness app access and function control framework.

In this paper, we propose and implement a unified speech-speaker recognizer (USR) framework that performs permission and response management based upon a customized identity management policy. The USR framework consists of four main modules: (i) an application interface cooperating with mobile applications, (ii) a speaker recognition module for identity recognition, (iii) a speech recognition module transcribing speech input, and (iv) an identity management module supervising the response to the applications according to the customized identity management policy. USR relies on several factors to perform seamless identity based application management including user identity, application privacy level, and

978-1-63190-024-2 © 2014 ICST
DOI 10.4108/icst.mobicase.2014.257766

TABLE I. EXAMPLE ANDROID APPLICATIONS THAT USE SPEECH RECOGNITION.

Application	**Features**
Slyvi	wake on voice, find places and get directions, update Twitter and Facebook, car mode
Google voice search	search your phone, the web, and nearby locations by speaking, instead of typing. Call your contacts, get directions, and control your phone with voice Actions
Speaktoit	Speaktoit uses natural language technology to answer questions, find information, launch applications, and connect user with various web services. It remembers users' favorite places, services, and preferences.
Utter voice command	Utter voice command runs in the background. It does not have a user-interface and controls the device using voice commands. It supports drive mode and wake on voice commands.
Voice remote control camera	User can remote control the camera inside a SmartPhone. Responding to sounds, the camera works automatically and a user can take a photo hands free.
Drive safely	It reads text messages, SMS and emails aloud and lets you respond by voice.
AVX	It can remote control garage door respond by voice.

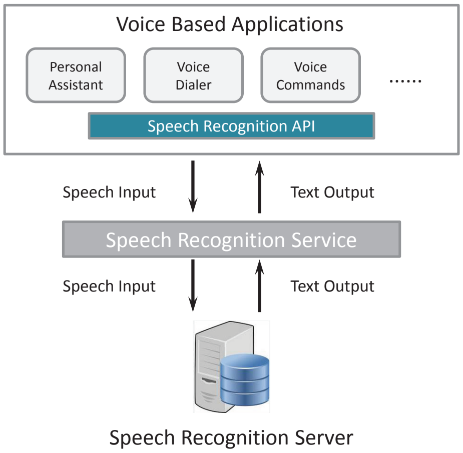


Fig. 2. General mobile speech recognition framework

application function class. In comparison to previous mobile operation permission management applications, an essential feature of the proposed solution is its convenience. The identity feature (*i.e.*, speech) is implicitly captured without disrupting normal user-mobile device interactions. In addition, it offers continuous post-login protection of mobile devices during all the speech interactions, thus protecting sensitive mobile device information and functions. The contributions of the paper are the following:

- Designed and implemented a unified speech-speaker recognizer (USR) framework that provides specific response corresponding to different user identity based on a customized identity management policy.
- Developed an open-source Android library for speaker recognition.
- Conducted a comparison study of USR and the Google speech recognition framework.

II. RELATED WORK

The USR framework idea draws from multiple threads of solution and research, including Speech Application, Speech Recognition Framework, and Mobile Identity Management.

A. Speech Application

The Android speech recognition API allows developers to integrate speech recognition directly into their applications. Since its release, numerous applications have been developed that leverage speech recognition capabilities provided by the Android platform. These applications include the voice dialer, voice search, voice note, personal assistant (similar to Apple's Siri), voice navigator, voice controlled camera, voice commands, and etc. Table I lists some smartphone applications that use speech recognition. Many speech recognition applications allow a user to interact with a mobile device hands free. They often provide a speech user interface that supports features such as waking up on voice commands, automatically posting messages to Twitter and Facebook using speech-to-text, launching applications based on voice commands, opening calendar, searching based on voice inputs (*e.g.*, search contact list and automatically dial a person's number), to even controlling mobile device hardware such as camera using voices.

Though providing convenience to a mobile user, these speech recognition based applications are potentially vulnerable to malicious exploits. Almost none of these applications we studied offers the capability to differentiate the speakers and enforces appropriate policies on who can interact and control a mobile device using speeches. Things can get even worse when we dive deeper into the speech based API and scrutinize its security. In order to support hands free interactions, activities triggered by speech can be launched while a mobile device is locked in a secure mode. This means voice based actions can take precedence over the secure mode that requires a user to unlock a mobile device. By setting the "FLAG_SHOW_WHEN_LOCKED" flag, a user may bypass the lock screen and interact with the mobile device when the device is still in a secure mode. To give a concrete scenario, an imposter may post to a victim's Twitter or Facebook account using speech when the a mobile device is in locked state.

B. Speech Recognition Framework

Figure 2 shows a general framework for current mobile speech recognition. When user input a speech command to a speech application, the speech application records its voice and calls the system speech recognition API to activate a preset speech recognition service. The recorded *wave* file is then sent to the speech recognition server through the speech recognition service running in the background. After the speech recognition server transcribe the speech, it returns the text of the command to the speech recognition service and the

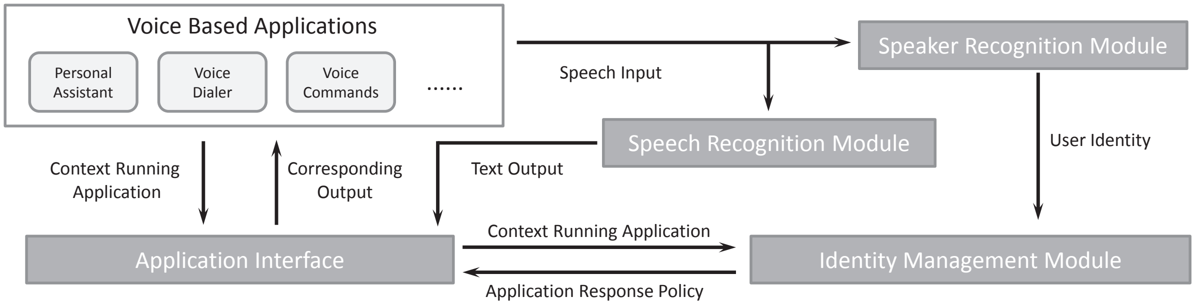

Voice Based Applications
Personal Assistant
Voice Dialer
Voice Commands
......
Speech Input
Speaker Recognition Module
Speech Recognition Module
User Identity
Context Running Application
Corresponding Output
Text Output
Context Running Application
Application Interface
Identity Management Module
Application Response Policy

Fig. 3. Design of USR framework

speech applications, and then the speech application follows the text command.

From the general mobile speech recognition framework, two weakness are obvious during the text command transfer from speech recognition service to system speech recognition API:

- The text command is transferred without any consideration on which application is calling the service and what is the feature of the application (*where usability can be promoted*).
- The text command is sent without any authentication process, which leaves a potential threat to the mobile system since the speech command will always follow the text command (*where privacy can be enhanced*).

C. Mobile Identity Management

Mobile identity management includes two consequent research topics: mobile identity sensing and mobile permission management.

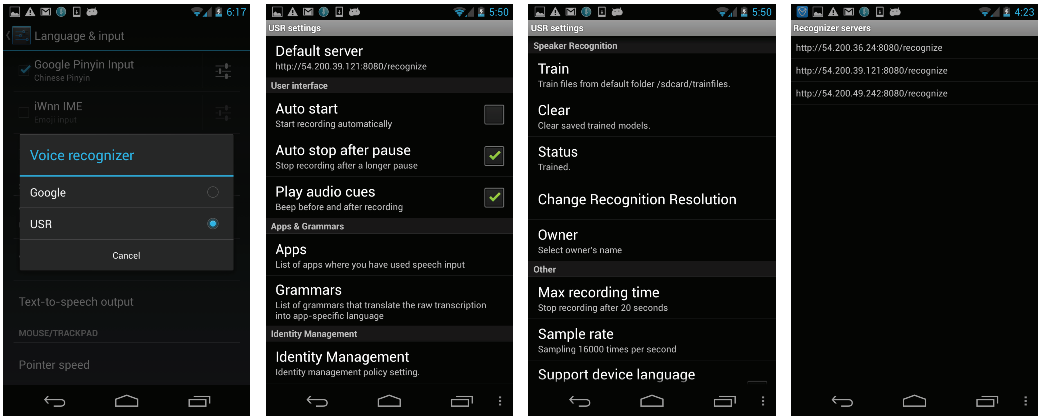
1) Mobile Identity Sensing.: Mobile identity sensing technology can be classified into two categories from the aspect of user's perception: *explicit* and *implicit*. Although explicit identity sensing solutions (*e.g.*, fingerprint scan) may provide stronger protection, it may sacrifice user experience. In a study [11] on users' perceptions of authentication on mobile devices, the results showed that a system that can implicitly and continuously perform user identification in the background without disrupting the normal user-mobile device interaction is a desired solution by the mobile device users. Furthermore, although explicit identity sensing can also solve the identity management problem in the post-login stage, such as AppLock, it sacrifice user experience since it requires specific extra operations both at the authentication stage. On the contrary, because implicit identity sensing happens during normal interactions, it could provide a complement continuous protection in the post-login stage. By the same time, some explicit identity sensing solutions (*e.g.*, password) may not be as strong as it designed to be. In [12], Denning *et al.* prove that text passwords have been known to impose a cognitive burden on the users that results in selection of weak passwords. Hence, implicit identity sensing technology is the preference in our paper.

With the increasing popularity of portable devices, several implicit identity sensing approaches have been proposed by leveraging the sensors that can be found in a mobile device, including accelerometer [13], [14], GPS [15], touchscreen [16]–[20], microphone [9], and fingerprint sensor [21], [22]. In our research, we were inspired by SpeakerSense [23], a speaker identification prototype that performs continuous background sensing and speaker identification with minimal power requirements. SpeakerSense manages to acquire training data from phone calls for training speaker models, which is one of the goals we had in mind when designing our software, as well. Our work is related but not complimentary, as the primary goal of our project is construct an implicit speaker recognition based USR framework.

Another application worth mentioning is SoundSense [10], which explores continuously sensing and classifying audio events to recognize general sound types heard by users (e.g., voice or music) and specific activities (e.g., walking, driving cars). These classifications enable a number of different applications including an audio daily diary and music detection service, which were both prototyped by the authors. SoundSense and other continuously sensing applications raise concerns about battery efficiency.

Finally, In all of the aforementioned designs, the goal is not to design new speaker identification or verification algorithms. Instead they are leveraging well-established techniques such as the MFCCs [24] and GMM classifiers [25], which have been proven effective for speaker identification. Our focus, on the other hand, is on extending a technique developed by Eyben *et al.* [26] for emotion recognition to speaker verification, and using them to address the challenges that arise when performing speaker identity sensing on energy constrained mobile phones.

2) Mobile Permission Management.: Multi-user mobile and other devices have been researched as a new topic for recent years. In [27], Karlson *et al.* discussed the privacy and security issues when users of smartphone lend their phone to other physical users. Mobile permission management as a derivative research topic used to handle the privacy and security problems then attracts researchers' attention. Rofouei *et al.* [28] researched on multi-user device-display interaction identity identification by using a group of devices, including a Kinect camera, a multi-touch display and 2 accelerometer-equipped phones (one visible). [29] also present xShare, a protection



6:17
Language & input
☑ Google Pinyin Input
Chinese Pinyin
□ iWnn IME
Emoji input
Voice recognizer
Google ○
USR ●
Cancel
Text-to-speech output
MOUSE/TRACKPAD
Pointer speed

(a) Voice recognizer

5:50
USR settings
Default server
http://54.200.39.121:8080/recognize
User interface
Auto start ☐
Start recording automatically
Auto stop after pause ☑
Stop recording after a longer pause
Play audio cues ☑
Beep before and after recording
Apps & Grammars
Apps
List of apps where you have used speech input
Grammars
List of grammars that translate the raw transcription into app-specific language
Identity Management
Identity Management
Identity management policy setting.

(b) USR framework setting

5:50
USR settings
Speaker Recognition
Train
Train files from default folder /sdcard/trainfiles.
Clear
Clear saved trained models.
Status
Trained.
Change Recognition Resolution
Owner
Select owner's name
Other
Max recording time
Stop recording after 20 seconds
Sample rate
Sampling 16000 times per second
Support device language

(c) USR framework setting continue

4:23
Recognizer servers
http://54.200.36.24:8080/recognize
http://54.200.39.121:8080/recognize
http://54.200.49.242:8080/recognize

(d) Server List

Fig. 4. USR framework setting on the Android device

solution to address privacy and security issues for the shared mobile scenario. However, previous work focusing only on how to utilize permission management to improve privacy and security, and ignores the potential usability promotion can achieved by identity awareness.

III. USR Framework

To address the weakness of the current speech recognition based API that recognizes speech without verifying the speaker, we designed USR framework, a solution that integrates speaker sensing and identity management with speech recognition. A high level diagram of the approach is presented in Figure 3. The solution extends the Android speech recognition API with speaker recognition, identity management support and access control. The new components include, an application interface that detects context running application and responds to the applications, an identity manager module that controls and enforces responding policies to speech commands based on speaker's identity, and a speaker recognition module.

A. Application Interface

The application interface have two core functions. The first function is to detect which application is the owner of the microphone (*e.g.*, personal assistant, voice search, skype). We implement it by utilizing an Android System API, *ActivityChangedListener*, to capture application package name (*e.g.*, "com.skype.raider" for Skype) in a background service. The application interface is then able to send the package name to identity management module to acquire the corresponding application response policy for this application. Another important function of this module is to react to the application based on the application response policy. For example, if the application response policy instructs the command is not allowed to send to the application, the application interface will prevent the text output from sending to the application.

B. Speech Recognition Module

The speech recognition module consists of a voice recognizer service and a speech recognition server. When USR framework installed on the device, users can select our service in Android *Voice Recognizer* (See in Figure 4(a)). Users may configure the settings of the voice recognizer service, speaker recognizer, identity management policy, and and other settings (*sampling rate*) by entering *voice search*, and the details are shown in Figure 4(b) and Figure 4(c). By default the service connects to our speech recognition server, but they can also connect to speech recognition server they desired by entering into the server list (Figure 4(d)).

On the server side, considering the recognition accuracy, we choose Google Speech Recognition Server as the speech recognition server and implement our server as a proxy server to connect it. The reason we do not connect Google Speech Recognition Server directly on the mobile side is that Google Speech Recognition Server requires the voice data to be *FLAC* format, which is not a default encoding method supported by Android system.

C. Speaker Recognition Module

1) Speaker Recognizer Design: Methods involving a set of Mel Frequency Cepstral Coefficients (MFCCs) have been dominant in the field of speaker recognition in the past decades. Human perception of the frequency content of sounds follow a subjectively defined nonlinear scale called the "mel" scale [30] defined as,

$$f_{mel} = 1125ln(1 + \frac{f}{700}) \tag{1}$$

where f is the frequency in Hz. The calculation of MFCCSs can be summarized in Figure 5:

When it comes to speaker recognition, vectors consisting of MFCCs and some features derived from them are used to

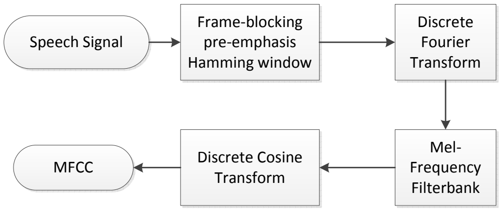

Speech Signal
Frame-blocking pre-emphasis Hamming window
Discrete Fourier Transform
Mel-Frequency Filterbank
Discrete Cosine Transform
MFCC

Fig. 5. MFCC calculation process

build Gaussian Mixture Models (GMM) [31]. More recently, they have also been used in classification schemes that involve Support Vector Machines (SVM) [30] [32]. These methods have been very effective for user verification, often having prediction accuracy of up to 95 percent, with state of the art speaker recognition systems generally having Equal Error Rate (EER) close to 0.

Generally speaking, the following technique is standard: MFCC features are extracted over a chosen frame length with a frame shift of about 1/2 its size to provide for an overlap, then either their means and standard deviations or derivatives and second derivatives are computed [33]. One particular property of the described approach is that given the number of instances *n* and the number of features *m* for almost any reasonable data set, the following property holds: $n >> m$. When SVM classifier is applied to such problems, RBF or polynomial kernel is typically chosen as they transform the feature vectors into higher dimensional space, increasing the probability to find a suitable hyperplane to separate the classes. Unfortunately due to performance and battery use considerations, using anything but a linear kernel on a mobile device is simply not practical.

As a part of this study, we developed an open-source Android library for speaker recognition. Figure 6 depicts a high level diagram of the system. It consists of a Java interface and three internal modules, written in C and C++: i) Feature Extractor ii) Feature Pre-Processor iii) Classifier. For the purpose of feature extraction, openSMILE: The Munich Versatile and Fast Open-Source Audio Feature Extractor [26] has been ported to Android platform. openSMILE is capable of producing output in various formats, which usually makes it directly compatible with most machine-learning libraries, but in some cases, further processing of needed. To address this issue, component ii) has been written. Finally, based on the findings we will discuss further in this section, libSVM [34] has been re-compiled to work on the Android platform, as well. It is worth noting that our software is highly modular and extendable, allowing for extraction of various features or usage of different classifiers or even using several classifiers at once, simply by editing the text configuration file.

In our method, features are computed in 3 steps.

i) A set of Low-level descriptors (LLD) is extracted. The LLDs in question are: Intensity, Loudness, 12 MFCC (Mel Frequency Cepstral Coefficients), Pitch (F0), Probability of voicing, F0 envelope, 8 LSF (Line Spectral Frequencies), Zero-Crossing Rate.

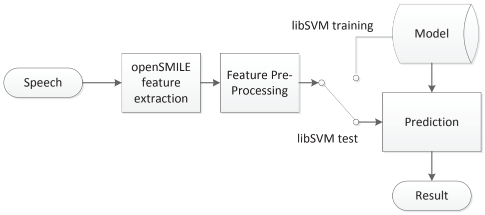


Fig. 6. Design of speaker recognition module

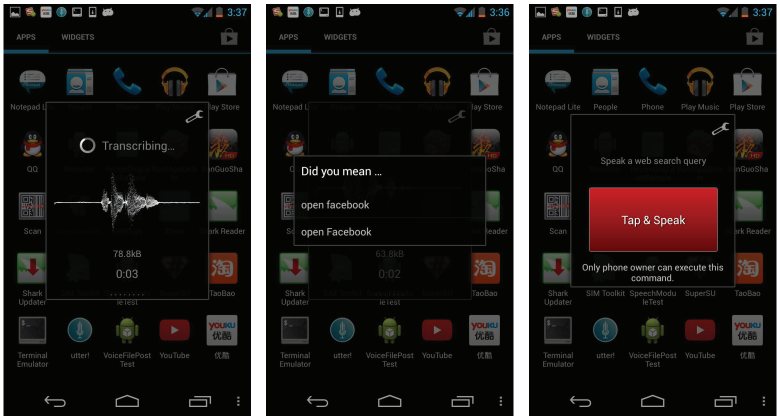
ii) Delta regression coefficients are computed from these LLD's.

iii) The following functionals are applied to both the original LLDs and their delta coefficients: Max./Min. values and their respective relative positions within input, range, arithmetic mean, 2 linear regression coefficients, linear and quadratic error, standard deviation, skewness, kurtosis, quartile 13, 3 inter-quartile ranges.

The first two steps are quite similar to the usual method, in fact, MFCCs are computed in precisely the same manner. The remaining descriptors extracted are common for emotion and speaker trait recognition, but some, such as F0 and LSF have been proved useful in speaker recognition. Step 3 results in 986 acoustic features, but reduces the number of instances to one per PCM file. Thus, we are dealing with a matrix where the inverse of the stated property of classical MFCC features holds, namely, $n << m$.

To optimize our system for the smart phone environment, we propose a new method of speaker recognition that is based on statistical descriptors of fundamental speech features. This scheme normally used for emotion recognition [26] but not verification, so although the features themselves are not really new, their application is. Since the scheme employs, in its first stage, a modified version of the emobase feature set, proposed by Eyben et al., we named it Speaker Identification Base, or SIDBase. The proposed method, when applied to a verification problem, is almost twice faster and more power efficient than the traditional approach, while maintaining accuracy, true positive rate (TPR) and false accept rate (FAR) similar to that of the state-of-the-art systems.

We take advantage of the increased number of dimensions and employ a linear SVM which benefits from a large number of features without suffering from the overfitting problem of most classifiers. Another reason to choose a Support Vector Machine classifier is that it is a two-class classifier, and speaker recognition is essentially a binary problem, for which it is well-suited.



Transcribing...

78.8kB

0:03

Did you mean ...

open facebook

open Facebook

Speak a web search query

Tap & Speak

Only phone owner can execute this command.

(a) Processing status (b) Identity match (c) Identity mismatch

Fig. 7. USR speaker Recognition Demo

2) Speaker Recognizer in USR framework: The configuration of speaker recognition can be found in Figure 4(c). Owner can perform a set of operations, such as train model, clear trained model, change owner's identity on the mobile device, etc.

We employ a speaker recognition demo application to demonstrate how it corporates with speech recognition module. When a user speaks ”Open Facebook” to the the demo application, the application will record the voice file and show wave of input voice (Figure 7(a)). The voice file is then duplicated and parallel processed by speaker recognition in local and speech recognition in remote. If the user is the owner of the mobile, the application will show the transcribed result as in Figure 7(b). Otherwise, the application will return no result and send the reject notice (*e.g.*, ”Only phone owner can execute this command”, ”Are you Kelvin(*owner name*)?”) to the user both in speech and text. (Figure 7(c)).

D. Identity Management Module

The identity management module can act according to the speaker recognition results and the preset customized identity management policies. The owner of the mobile can configure the identity management policies in the USR framework setting (Figure 4(b)). All the applications have preset identity management policies will show as a list in the setting (Figure 8(a)). The owner can add new application to the management list (Figure 8(b)). For an application in the list, the owner may modify the identity management policy choose whose speech inputs the application should respond to (Figure 8(c)). Currently, in our design, we have three types of policies:

- **Owner**. In this setting, a speech recognition based mobile app only responds to speech commands from a verified speaker;
- **All**. Under this setting, any user can access to the app using speech without authentication; and
- **Tag**. Tagging is a novel feature of our USR framework. Instead of policing speech based access to applications, it returns the recognized speech text with an identity tag in front of it as prefix.

For example, a user may label a hands free voice dialer with Owner. When detecting that an unverified user is accessing the application through speech commands, the speaker identity manager will point out the current user is not the owner of the mobile device and refuse to follow the speech commands. The system will not affect speakers with verified identities as they can continue to interact with the device hands free.

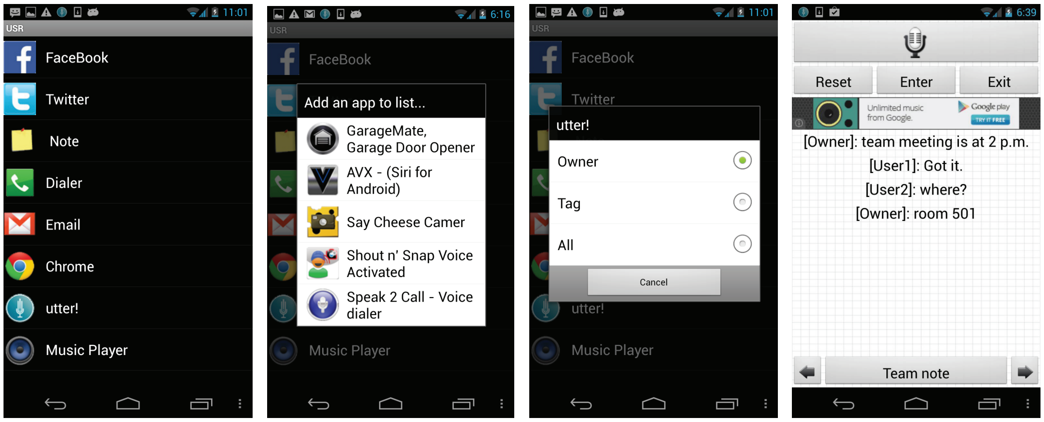
In another scenario, the owner label a notepad with Tag, the application may automatically record meeting minutes with identity if all the people in the meeting have trained model on the device (Figure 8(d)). More importantly, this Tag policy is intended to promote user experience by providing a **identity awareness interface**. This interface is not limited to benefit one or two single applications, but a general identity management interface for all the applications. A lot of similar scenarios (*e.g.*, posting group status on social network, automatically account switching, etc) may also benefit from our interface. It can be inferred that this identity awareness interface has a potential to greatly improve user experience.

IV. Method

We conducted an empirical study in order to explore the benefits and limitations of USR framework. As a baseline for comparison, we used Google Speech Recognizer, a widely used and powerful speech recognizer which dominates the Android market, and AppLock, a representation of current app access control solutions. The focus of the contrast experiment is not the speech recognition accuracy (Since we all use Google Speech Recognition Server), but the privacy protection and user experience promoted by identity awareness as well as the extra instituted cost.

A. Model Training

Before we conduct the user study, we first need to train the owner's model on the smartphone device. The owner user



[Owner]: team meeting is at 2 p.m.
[User1]: Got it.
[User2]: where?
[Owner]: room 501

(a) Application management list (b) Add application to management list (c) Configure application response policy (d) Tag for identity

Fig. 8. USR Identity Management

are provided a Nexus 4 smartphone with one week for their everyday usage. They use the devices naturally and the USR training module will implicitly record their voice commands and send to the USR server. For each owner user, about 400 voice commands(average 20 times for 1 command, 20 commands) is fairly enough for training a user's model.

B. Participant

Sixteen participants (6 female, 10 male), between the ages of 21 and 45, were recruited as mobile owner user. Another sixty subjects (24 female, 36 male), between the ages of 19 and 43 years, are enrolled as the guest users. We pre-trained the model of all mobile owner users with the voice data collected from them as positive samples. For the negative samples, we employed several voice data sets from different sources, i) a LDC speech dataset [35] that was purchased, consisting of 630 speakers of eight major dialects of American English, each reading ten phonetically rich sentences; ii) an open source voice dataset ELSDSR [36], which contains voice messages from 22 speakers (10 female, 12 male), with ages from 24 to 63 included; and iii) a voice dataset collected from our previous data collection (12 volunteers with 6 female and 6 male). By considering speaker recognition performance variations caused by both dialects and accents, we selected subjects from different countries and regions, including different dialects of American native speakers, Europeans, Chinese, Indians, and etc.

All of the participants reported having used smart mobile devices, such as a smartphone or tablet. 75% of the mobile owner users and 68.3% of the guest users reported being familiar with voice operations on mobiles devices.

C. Design

We installed the USR framework on four Google Nexus 4 smart phones with all 16 mobile owner users' model pre-trained on each device. The mobile owner users take turns to use the mobile devices to perform operations following our experiment procedures.

The experiment consisted of two different tasks for different user group and purpose. The first task is a privacy evaluation task. Both the mobile owner user and the guest user are required to enroll in this task. The second task is usability evaluation task, which is only conducted on the mobile owner users.

1) T1-Privacy Evaluation Task: This task is utilized to evaluate the privacy protection enhancements of the USR framework in comparison to the Google Speech Recognizer, which provides no identity authentication. We first gave the participants a quick tutorial on voice operations and voice applications. Then we demonstrated the security vulnerabilities in Google Speech Recognizer by showing them how to bypass the login authentication and use some sensitive operations (*e.g.*, post Facebook status, call or send SMS to someone). Furthermore, we showed them some sensitive applications that can remote control garage doors, automobiles, and etc., all without identity authentication.

After said introduction, we let the mobile owner users set the identity management policy in the USR framework, and ask each guest user to speak at least ten different voice commands to sensitive applications or perform sensitive operations in both quiet and noisy environments. We collected the performance results of the experiment and the execution times of the USR framework with controlled execution time using Google Speech Recognizer. An exit questionnaire and interview were also required for both the mobile owner user and guest user.

2) T2-Usability Evaluation Task: To evaluate the usability promotion of the USR framework in comparison to the Google Speech Recognizer plus AppLock, we first asked the mobile owner users to input voice commands in both quiet and noisy environments to their device to access the apps they desire to use. We then asked the users to repeat the same action by using the Speech Recognizer plus AppLock. After the

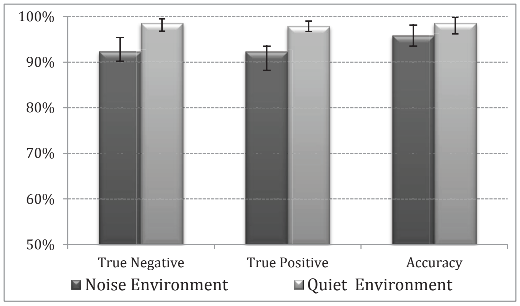


Fig. 9. Speaker recognizer performance result

comparison experiment, we simulated a multi-user scenario that requires four mobile owner users to using a notepad with Tag mode in the USR framework. Both the self-test scenario and multi-user scenario were recorded. As in the previous task, the execution times are also collected for cost evaluations. After the experiments, we introduced the potential use case of multi-user scenarios to the mobile owner users, especially highlighting its extensibility. The mobile owner users also complete an exit questionnaire and interview at the end of this task.

V. Result & Discussion

In this section, we present result and metrics collected in the experiment, as well as the overall study observation and participantspreference and feedback.

A. Performance Evaluation

Figure 9 shows the speaker recognition accuracy performance result of privacy evaluation task and usability evaluation task. Even we have implemented a noisy filter in USR framework, our speaker recognizer performs as we intuitively expected: in the quiet environment, all true positive, true negative, and accuracy rate are better than noisy environment. However, on the other hand, an accuracy of 95.83% also shows its ability in countering noise. Such a high accuracy rate is also a guarantee to the stability of USR framework.

Table II depicts the average extra time cost of USR framework comparing to Google Speech Recognizer. Since the quiet environment is in our laboratory where high speed wifi is provided, and the noisy environment is on the street where only 3G plan is available, the extra delay in noisy environment may come from the network connection status and transmission rate. Another explicit trend we can found is that, as the length of commands increase, the delay also increases. This may result in two reasons: i) larger wave file further expose the low efficiency of our server comparing to Google Speech Recognition server; and ii) speaker recognition for larger wave file will cost more time than small files. Since speaker recognition and speech recognition work parallel in USR framework, the speech result maybe blocked by the speaker recognition process. Although the for some long sentences, USR framework may cost close to 50% extra time to process it, it is still just a several seconds longer waiting time. Compare to the security it brings, this cost is acceptable.

Length of Command	Delay in Noisy	Delay in Quiet
1 to 3 words	25.18%	18.29%
4 to 6 words	35.92%	33.52%
6 and above	45.62%	39.12%

TABLE II. Execution time delay comparing to Google Speech Recognizer

B. Questionnaires

After each task, we asked the participants to answer a post-study questionnaire about the privacy protection enhancement and usability promotion of USR framework for T1 and T2, using a 5-point Likert scale (1: Strongly Disagree and 5: Strongly Agree). Since currently there is no similar framework that protects user privacy or performs identity management during speech interactions, these Likert-scale results are not meant to be directly compared with other previous solutions.

For the first task, Overall, the participants thought USR improved privacy protection during human-mobile speech interactions (47 guest users and 15 mobile owner users Strongly Agreed, and 13 guest users and 1 mobile owner user Agreed) and the extra time delay is acceptable (48 guest users and 14 mobile owner users Strongly Agreed, 9 guest users and 2 mobile owner users Agreed, and 3 guest users Neutral).

Most of the users also thought a low rate false reject rate is acceptable considering enhanced privacy, also the Google Speech Recognizer plus Applock is more annoying since it requires explicit password input all the time (11 mobile owner users Strongly Agreed, 3 mobile owner users Agreed, 1 mobile owner user Neutral, and 1 mobile owner user Disagreed). In particular, the participants all found the Tag mode in identity management for multi-user scenarios very interesting (14 Strongly Agreed and 2 Agreed), and have potential to extend some useful applications (12 Strongly Agreed and 4 Agreed).

C. Interviews and Observations

The initial observations from the questionnaires indicated that mobile phone users admit the privacy enhancement and usability promotion by the USR framework. To better understanding the detail experience and feedback of the users, at the end of the study, participants were asked to state their attitude towards USR framework and why. Three of the mobile owner users stated that they had read the news about how to use Siri or Google Voice Action to bypass mobile login procedures, two of them are worried about the privacy vulnerabilities and have even disabled this function. After experiencing our USR framework, they feel their privacy can be properly preserved by it.

> *"I am impressed by the accuracy of USR framework. During the experiment of Task 1, seldom [guest] users can bypass the USR [framework] and control my device. Some [guest] users even try to mimic my accent and voice, however they still failed."-Mobile owner user 2*

> *"Some apps are too sensitive, like a garage control application I saw the other day [some one] can open the garage without any authentication. USR [framework] could at least provide an extra protection to those [sensitive] apps."-Mobile owner user 4*

Although guest users are intend to simulate malicious intruders in Task 1, most of them share similar opinion that USR framework offers sufficient privacy protection after failing to command mobile owners' device for several times. Particularly, one guest user thought the USR framework is really helpful and can be ported to wearable devices, such as Google Glass, or Samsung Watch.

> *"I think this technology is useful because it implicit identify user during normal speech interactions. Since wearable devices have become more and more popular, it [USR framework] may be extend to broader usage scenarios."-Guest user 22*

However, different from the encouraging feedbacks on the privacy protection, one mobile owner user considered the USR framework sacrifice too much usability in his point of view.

> *"Although it [USR framework] protects my privacy on the mobile device, it makes the system hard to operate for me. In a noisy environment, I was faulty recognized as a guest user within ten commands a time, which means I have to repeat one command in ten commands. I think this put extra burden on me."-Mobile owner user 8, the mobile owner user with worst performance rate and the only user have a true positive under noisy environment.*

Although current speaker recognizer performs good and designing a speaker recognizer is part of our study, it is not the most critical part. The core contribution in this work is first proposed a unified speech-speaker recognizer framework, whereas the speaker recognizer performance can be enhanced by further researching on our own speaker recognizer or simply employ and integrate other available advanced solutions. And comparing the other app access control solutions, it does improved usability since it will only require explicit login when there is a false alarm in USR - for AppLock, authentication takes place every time users try to open an app.

Nevertheless, all mobile owner users are interesting in the Tag mode in identity management for multi-user scenarios, and some are extremely attract by this feature and provide us some constructive suggestions.

> *"Well, this idea is amazing. I can imagine several use scenarios now. It [Tag mode] could be used for implicit identity switching, like implicit switch facebook account when the command post facebook status comes from different user on a shared device. Furthermore, unlike current single user status, it may help social network extending some group activity or similar stuffs."-Mobile owner user 2, a mobile owner user with computer science background*

In conclusion, participants appreciated the enhanced protection by USR framework and admit it solves practical problem in real world. For most of them, the sacrificed usability caused by USR (*e.g.*, false reject rate or response delay) is either not notable or acceptable. Specifically, all the users endorse the new Tag mode since its potential to providing fancy services.

VI. Acknowledgement

This material is based upon work supported by the U.S. Department of Homeland Security under Award Number N66001-13-C-3002. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the opinions or policies of the U.S. Government. Mention of trade names or commercial products does not constitute their endorsement by the U.S. Government.

VII. Conclusion & Future Work

We presented the USR framework, a novel framework for integrating speaker recognition based identity management into current human-mobile speech interaction framework. The USR framework employs many factors to perform seamless identity-based application management, including user identity, application privacy level, usable app access control, and application function class. As a part of this study, we also contributed an open-source Android library for speaker recognition. To investigate the effectiveness and efficiency of the technique, a controlled experiment was conducted comparing USR framework with an established speech recognition technique, Google speech recognizer, and in addition, a representation of current app access control solutions, Applock. The comparison study of these two frameworks shows that our unified speech-speaker recognizer framework can significantly improve upon existing practical problems in mobile privacy protection and improve user experience to some extent.

In the future, we would like to investigate potential usage of multi-user scenario utilizing Tag mode in the identity management policy, such as implicit identity switching or group social activity. Another area of future research is to improve the accuracy of the speaker recognition as well as the efficiency of the framework to make it more transparent to mobile users. In the mean time, porting the USR framework to other wearable devices could also be an interesting topic.

References

[1] A. Kumar, A. Tewari, S. Horrigan, M. Kan, F. Metze, and J. Canny, "Rethinking speech recognition on mobile devices," in *Intelligent User Interfaces for Developing Regions 2011*, 2011.

[2] B. Johnson, "Google voice." *Computers in Libraries*, vol. 30, no. 5, pp. 20 – 24, 2010. [Online]. Available: http://ezproxy.lib.uh.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=rzh&AN=2010679765&site=ehost-live

[3] "CMU Sphinx – Speech Recognition Toolkit," http://cmusphinx.sourceforge.net/2010/05/vocalkit-shim-for-speech-recognition-on-iphone/.

[4] "Openears: free speech recognition and speech synthesis for the iphone," http://www.politepix.com/openears/.

[5] "Security flaw! google voice actions usable on lock screen!" http://forum.xda-developers.com/showthread.php?t=806923.

[6] J. Wolford, "Siri bypasses your iphone passcode, by default," http://www.stumbleupon.com/su/2giG5z.

[7] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[8] A. Kumar, P. Reddy, A. Tewari, R. Agrawal, and M. Kam, "Improving literacy in developing countries using speech recognition-supported games on mobile devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 1149–1158. [Online]. Available: http://doi.acm.org/10.1145/2207676.2208564

[9] H. Lu, A. Bernheim Brush, B. Priyantha, A. Karlson, and J. Liu, "Speakersense: Energy efficient unobtrusive speaker identification on mobile phones," in *Pervasive Computing*, ser. Lecture Notes in Computer Science, K. Lyons, J. Hightower, and E. Huang, Eds. Springer Berlin Heidelberg, 2011, vol. 6696, pp. 188–205. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-21726-5_12

[10] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys '09. New York, NY, USA: ACM, 2009, pp. 165–178. [Online]. Available: http://doi.acm.org/10.1145/1555816.1555834

[11] M. Jakobsson, E. Shi, and R. Chow, "Implicit authentication for mobile devices," in *4th USENIX Workshop on Hot Topics in Security*.

[12] T. Denning, K. Bowers, M. van Dijk, and A. Juels, "Exploring implicit memory for painless password recovery," in *CHI '11 Proceedings of the 2011 annual conference on Human factors in computing systems*, New York, NY, 2012, pp. 2615–2618.

[13] J. Mantyjarvi, M. Lindholm, E. Vildjiounaite, S. marja Makela, and H. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[14] T. Feng, X. Zhao, and W. Shi, "Investigating mobile device picking-up motion as a novel biometric modality," in *Biometrics: Theory, Applications and Systems (BTAS), 2013 IEEE Sixth International Conference on*, Sept 2013, pp. 1–6.

[15] P. Marcus, M. Kessel, and C. Linnhoff-Popien, "Securing mobile device-based machine interactions with user location histories," in *Security and Privacy in Mobile Information and Communication Systems*. Springer Berlin Heidelberg, 2012.

[16] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, "Biometric-rich gestures: a novel approach to authentication on multi-touch devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

[17] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong, "Senguard: Passive user identification on smartphones using multiple sensors," in *IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications*, 2011, pp. 141–148.

[18] T. Feng, X. Zhao, B. Carbunar, and W. Shi, "Continuous mobile authentication using virtual key typing biometrics." in *TrustCom/ISPA/IUCC*. IEEE, 2013, pp. 1547–1552. [Online]. Available: http://dblp.uni-trier.de/db/conf/trustcom/trustcom2013.html#FengZCS13

[19] T. Feng, Z. Liu, K.-A. Kwon, W. Shi, B. Carbunar, Y. Jiang, and N. Nguyen, "Continuous mobile authentication using touchscreen gestures," in *Homeland Security (HST), 2012 IEEE Conference on Technologies for*, Nov 2012, pp. 451–456.

[20] T. Feng, J. Yang, Z. Yan, E. M. Tapia, and W. Shi, "Tips: context-aware implicit user identification using touch screen in uncontrolled environments," in *Proceedings of the 15th Workshop on Mobile Computing Systems and Applications*. ACM, 2014, p. 9.

[21] T. Feng, Z. Liu, B. Carbunar, D. Boumber, and W. Shi, "Continuous remote mobile identity management using biometric integrated touch-display," in *Microarchitecture Workshops (MICROW), 2012 45th Annual IEEE/ACM International Symposium on*, Dec 2012, pp. 55–62.

[22] T. Feng, V. Prakash, and W. Shi, "Touch panel with integrated fingerprint sensors based user identity management," in *Technologies for Homeland Security (HST), 2013 IEEE International Conference on*, Nov 2013, pp. 154–160.

[23] H. Lu, A. J. B. Brush, B. Priyantha, A. K. Karlson, and J. Liu, "Speakersense: energy efficient unobtrusive speaker identification on mobile phones," in *Proceedings of the 9th international conference on Pervasive computing*, ser. Pervasive'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 188–205. [Online]. Available: http://dl.acm.org/citation.cfm?id=2021975.2021992

[24] K. Murty and B. Yegnanarayana, "Combining evidence from residual phase and mfcc features for speaker recognition," *Signal Processing Letters, IEEE*, vol. 13, no. 1, pp. 52–55, 2006.

[25] M. Ferras, C.-C. Leung, C. Barras, and J. Gauvain, "Comparison of speaker adaptation methods as feature extraction for svm-based speaker recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 6, pp. 1366–1378, 2010.

[26] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *Proceedings of the international conference on Multimedia*, ser. MM '10. New York, NY, USA: ACM, 2010, pp. 1459–1462. [Online]. Available: http://doi.acm.org/10.1145/1873951.1874246

[27] A. K. Karlson, A. B. Brush, and S. Schechter, "Can i borrow your phone?: understanding concerns when sharing mobile phones," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

[28] M. Rofouei, A. Wilson, A. Brush, and S. Tansley, "Your phone or mine?: fusing body, touch and device sensing for multi-user device-display interaction," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

[29] Y. Liu, A. Rahmati, Y. Huang, H. Jang, L. Zhong, Y. Zhang, and S. Zhang, "xshare: supporting impromptu sharing of mobile phones," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*.

[30] S. huang Chen and Y. ren Luo, "Speaker verification using mfcc and support vector machine," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2009.

[31] D. Reynolds and R. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *Speech and Audio Processing, IEEE Transactions on*, vol. 3, no. 1, pp. 72–83, 1995.

[32] C. H. You, K.-A. Lee, and H. Li, "Gmm-svm kernel with a bhattacharyya-based distance for speaker recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 6, pp. 1300–1312, 2010.

[33] Z. Fang, Z. Guoliang, and S. Zhanjiang, "Comparison of different implementations of mfcc," *J. Comput. Sci. Technol.*, vol. 16, no. 6, pp. 582–589, Nov. 2001. [Online]. Available: http://dx.doi.org/10.1007/BF02943243

[34] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011. [Online]. Available: http://doi.acm.org/10.1145/1961189.1961199

[35] "Linguistic Data Consortium: LDC," http://www.ldc.upenn.edu/.

[36] L. Feng and L. K. Hansen, "A new database for speaker recognition," Richard Petersens Plads, Building 321, DK-2800 Kgs. Lyngby, 2005. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3662